Computer algebra independent integration tests

4-Trig-functions/4.3-Tangent/4.3.4.2-a+b-tan-^m-c+d-tan-^n-A+B-tan+C-tan^2-

Nasser M. Abbasi

July 17, 2021 Compiled on July 17, 2021 at 2:23pm

Contents

Chapter 1

Introduction

This report gives the result of running the computer algebra independent integration problems. The listing of the problems are maintained by and can be downloaded from `https://rulebasedintegration.org`

The number of integrals in this report is [171]. This is test number [105].

1.1 Listing of CAS systems tested

The following systems were tested at this time.

1. Mathematica 12.3 (64 bit) on windows 10.
2. Rubi 4.16.1 in Mathematica 12.1 on windows 10.
3. Maple 2021.1 (64 bit) on windows 10.
4. Maxima 5.44 on Linux. (via sagemath 9.3)
5. Fricas 1.3.7 on Linux (via sagemath 9.3)
6. Giac/Xcas 1.7 on Linux. (via sagemath 9.3)
7. Sympy 1.8 under Python 3.8.8 using Anaconda distribution on Ubuntu.
8. Mupad using Matlab 2021a with Symbolic Math Toolbox Version 8.7 under windows 10 (64 bit)

Maxima, Fricas and Giac/Xcas were called from inside SageMath. This was done using SageMath integrate command by changing the name of the algorithm to use the different CAS systems.

Sympy was called directly using Python.

1.2 Results

Important note: A number of problems in this test suite have no antiderivative in closed form. This means the antiderivative of these integrals can not be expressed in terms of elementary, special functions or `Hypergeometric2F1` functions. `RootSum` and `RootOf` are not allowed.

If a CAS returns the above integral unevaluated within the time limit, then the result is counted as passed and assigned an A grade.

However, if CAS times out, then it is assigned an F grade even if the integral is not integrable, as this implies CAS could not determine that the integral is not integrable in the time limit.

If a CAS returns an antiderivative to such an integral, it is assigned an A grade automatically and this special result is listed in the introduction section of each individual test report to make it easy to identify as this can be important result to investigate.

The results given in in the table below reflects the above.

System	solved	Failed
Rubi	% 100.00 (171)	% 0.00 (0)
Mathematica	% 98.83 (169)	% 1.17 (2)
Maple	% 71.35 (122)	% 28.65 (49)
Maxima	% 49.12 (84)	% 50.88 (87)
Fricas	% 49.12 (84)	% 50.88 (87)
Sympy	% 33.92 (58)	% 66.08 (113)
Giac	% 46.20 (79)	% 53.80 (92)
Mupad	% 60.23 (103)	% 39.77 (68)

Table 1.1: Percentage solved for each CAS

The table below gives additional break down of the grading of quality of the antiderivatives generated by each CAS. The grading is given using the letters A,B,C and F with A being the best quality. The grading is accomplished by comparing the antiderivative generated with the optimal antiderivatives included in the test suite. The following table describes the meaning of these grades.

grade	description
A	Integral was solved and antiderivative is optimal in quality and leaf size.
B	Integral was solved and antiderivative is optimal in quality but leaf size is larger than twice the optimal antiderivatives leaf size.
C	Integral was solved and antiderivative is non-optimal in quality. This can be due to one or more of the following reasons 1. antiderivative contains a hypergeometric function and the optimal antiderivative does not. 2. antiderivative contains a special function and the optimal antiderivative does not. 3. antiderivative contains the imaginary unit and the optimal antiderivative does not.
F	Integral was not solved. Either the integral was returned unevaluated within the time limit, or it timed out, or CAS hanged or crashed or an exception was raised.

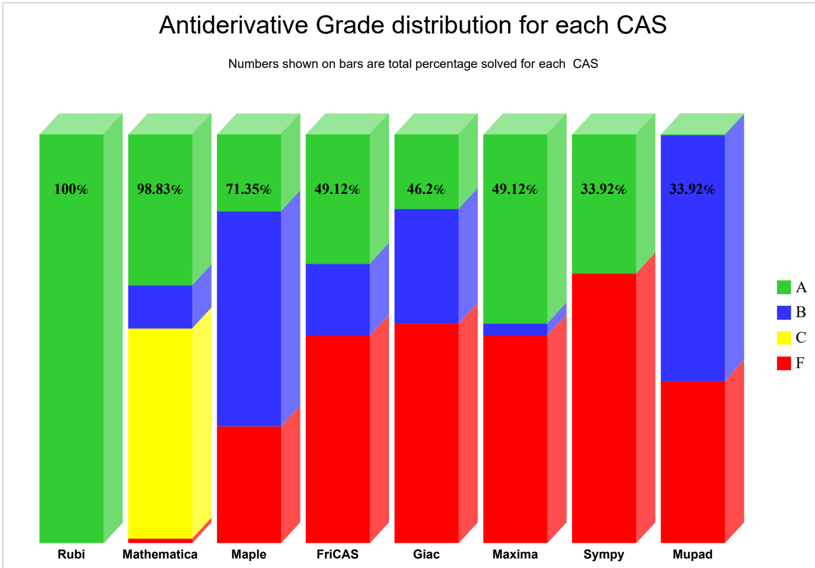
Table 1.2: Description of grading applied to integration result

Grading is implemented for all CAS systems. Based on the above, the following table summarizes the grading for this test suite.

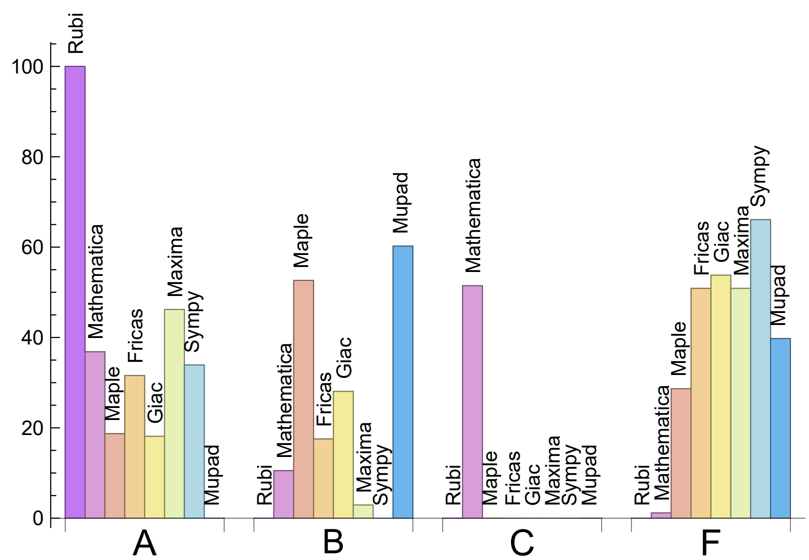
System	% A grade	% B grade	% C grade	% F grade
Rubi	100.00	0.00	0.00	0.00
Mathematica	36.84	10.53	51.46	1.17
Maple	18.71	52.63	0.00	28.65
Maxima	46.20	2.92	0.00	50.88
Fricas	31.58	17.54	0.00	50.88
Sympy	33.92	0.00	0.00	66.08
Giac	18.13	28.07	0.00	53.80
Mupad	0.00	60.23	0.00	39.77

Table 1.3: Antiderivative Grade distribution of each CAS

The following is a Bar chart illustration of the data in the above table.



The figure below compares the CAS systems for each grade level.



The following table shows the distribution of the different types of failure for each CAS. There are 3 types of reasons why it can fail. The first is when CAS returns back the input within the time limit, which means it could not solve it. This the typical normal failure **F** .

The second is due to time out. CAS could not solve the integral within the 3 minutes time limit which is assigned **F(-1)**.

The third is due to an exception generated. Assigned **F(-2)**. This most likely indicates an interface problem between sagemath and the CAS (applicable only to FriCAS, Maxima and Giac) or it could be an indication of an internal error in CAS. This type of error requires more investigations to determine the cause.

System	Number failed	Percentage normal failure	Percentage timeout failure	Percentage exception failure
Rubi	0	0.00 %	0.00 %	0.00 %
Mathematica	2	100.00 %	0.00 %	0.00 %
Maple	49	24.49 %	75.51 %	0.00 %
Maxima	87	25.29 %	52.87 %	21.84 %
Fricas	87	14.94 %	85.06 %	0.00 %
Sympy	113	68.14 %	11.50 %	20.35 %
Giac	92	11.96 %	88.04 %	0.00 %
Mupad	68	38.24 %	61.76 %	0.00 %

Table 1.4: Time and leaf size performance for each CAS

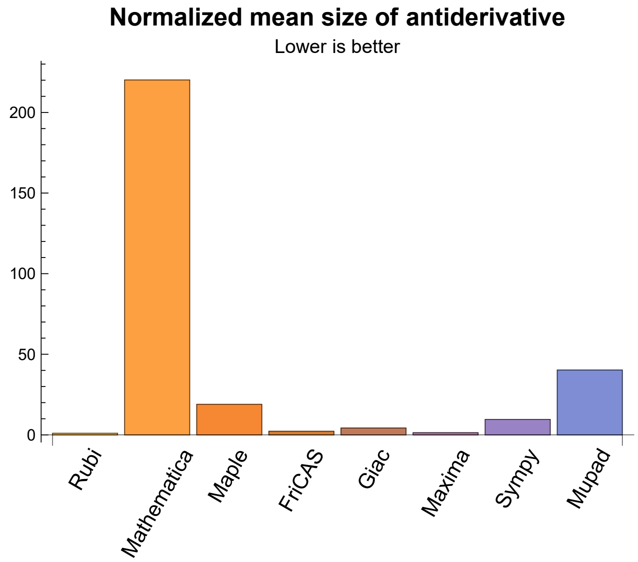
1.3 Performance

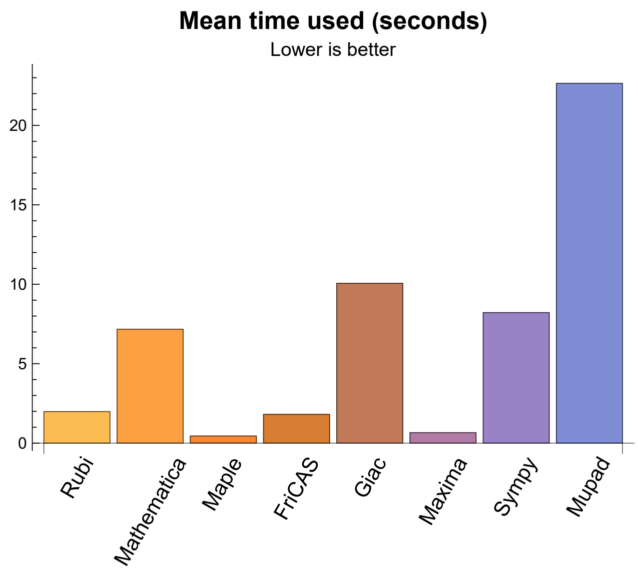
The table below summarizes the performance of each CAS system in terms of CPU time and leaf size of results.

System	Mean time (sec)	Mean size	Normalized mean	Median size	Normalized median
Rubi	1.99	323.47	1.00	287.00	1.00
Mathematica	7.17	110174.08	220.18	322.00	1.39
Maple	0.45	6746.32	18.94	994.00	3.40
Maxima	0.66	375.69	1.35	217.50	1.20
Fricas	1.81	803.68	2.23	269.50	1.58
Sympy	8.22	1779.53	9.58	579.00	2.52
Giac	10.06	968.13	4.25	435.00	2.23
Mupad	22.65	13887.19	40.23	307.00	1.38

Table 1.5: Time and leaf size performance for each CAS

The following are bar charts for the normalized leafsize and time used columns from the above table.





1.4 list of integrals that has no closed form antiderivative

{}

1.5 list of integrals solved by CAS but has no known antiderivative

Rubi {}

Mathematica {}

Maple {}

Maxima {}

Fricas {}

Sympy {}

Giac {}

Mupad {}

1.6 list of integrals solved by CAS but failed verification

The following are integrals solved by CAS but the verification phase failed to verify the anti-derivative produced is correct. This does not mean necessarily that the anti-derivative is wrong, as additional methods of verification might be needed, or more time is needed (3 minutes time limit was used). These integrals are listed here to make it easier to do further investigation to determine why it was not possible to verify the result produced.

Rubi {}

Mathematica {69, 83, 84, 89, 128, 132, 135, 138, 139, 141, 143, 144, 145, 146, 153, 154, 155, 159, 160}

Maple Verification phase not implemented yet.

Maxima Verification phase not implemented yet.

Fricas Verification phase not implemented yet.

Sympy Verification phase not implemented yet.

Giac Verification phase not implemented yet.

Mupad Verification phase not implemented yet.

1.7 Timing

The command `AboluteTiming[]` was used in Mathematica to obtain the elapsed time for each integrate call. In Maple, the command `Usage` was used as in the following example

```
cpu_time := Usage(assign ('result_of _int',int(expr,x)),output='realtime'
```

For all other CAS systems, the elapsed time to complete each integral was found by taking the difference between the time after the call has completed from the time before the call was made. This was done using Python's `time.time()` call.

All elapsed times shown are in seconds. A time limit of 3 minutes was used for each integral. If the integrate command did not complete within this time limit, the integral was aborted and considered to have failed and assigned an F grade. The time used by failed integrals due to time out is not counted in the final statistics.

1.8 Verification

A verification phase was applied on the result of integration for Rubi and Mathematica. Future version of this report will implement verification for the other CAS systems. For the integrals whose result was not run through a verification phase, it is assumed that the antiderivative produced was correct.

Verification phase has 3 minutes time out. An integral whose result was not verified could still be correct. Further investigation is needed on those integrals which failed verifications. Such integrals are marked in the summary table below and also in each integral separate section so they are easy to identify and locate.

1.9 Important notes about some of the results

1.9.1 Important note about Maxima results

Since these integrals are run in a batch mode, using an automated script, and by using sagemath (SageMath uses Maxima), then any integral where Maxima needs an interactive response from the user to answer a question during evaluation of the integral in order to complete the integration, will fail and is counted as failed.

The exception raised is `ValueError`. Therefore Maxima result below is lower than what could result if Maxima was run directly and each question Maxima asks was answered correctly.

The percentage of such failures were not counted for each test file, but for an example, for the Timofeev test file, there were about 14 such integrals out of total 705, or about 2 percent. This pecrentage can be higher or lower depending on the specific input test file.

Such integrals can be indentified by looking at the output of the integration in each section for Maxima. The exception message will indicate of the error is due to the interactive question being asked or not.

Maxima integrate was run using SageMath with the following settings set by default

```
'besselexpand : true'
'display2d : false'
'domain : complex'
'keepfloat : true'
'load(to_poly_solve)'
'load(simplify_sum)'
'load(abs_integrate)' 'load(diag)'
```

SageMath loading of Maxima `abs_integrate` was found to cause some problem. So the following code was added to disable this effect.

```
from sage.interfaces.maxima_lib import maxima_lib
maxima_lib.set('extra_definite_integration_methods', '[]')
maxima_lib.set('extra_integration_methods', '[]')
```

See `https://ask.sagemath.org/question/43088/integrate-results-that-are-different-from-using-maxima/` for reference.

1.9.2 Important note about FriCAS and Giac/XCAS results

There are Few integrals which failed due to SageMath not able to translate the result back to SageMath syntax and not because these CAS system were not able to do the integrations.

These will fail With error `Exception raised: NotImplementedError`

The number of such cases seems to be very small. About 1 or 2 percent of all integrals.

Hopefully the next version of SageMath will have complete translation of FriCAS and XCAS syntax and I will re-run all the tests again when this happens.

1.9.3 Important note about finding leaf size of antiderivative

For Mathematica, Rubi and Maple, the buildin system function `LeafSize` is used to find the leaf size of each antiderivative.

The other CAS systems (SageMath and Sympy) do not have special buildin function for this purpose at this time. Therefore the leaf size for Fricas and Sympy and Giac antiderivatives is determined using the following function, thanks to user `slelievre` at `https://`

ask.sagemath.org/question/57123/could-we-have-a-leaf_count-function-in-base-sagemath/

```
def tree_size(expr):
    r"""
    Return the tree size of this expression.
    """
    if expr not in SR:
        # deal with lists, tuples, vectors
        return 1 + sum(tree_size(a) for a in expr)
    expr = SR(expr)
    x, aa = expr.operator(), expr.operands()
    if x is None:
        return 1
    else:
        return 1 + sum(tree_size(a) for a in aa)
```

For Sympy, which is called directly from Python, the following code is used to obtain the leafsize of its result

```
try:
  # 1.7 is a fudge factor since it is low side from actual leaf count
  leafCount = round(1.7*count_ops(anti))

  except Exception as ee:
         leafCount =1
```

1.9.4 Important note about Mupad results

Matlab's symbolic toolbox does not have a leaf count function to measure the size of the antiderivative, Maple was used to determine the leaf size of Mupad output by post processing.

Currently no grading of the antiderivative for Mupad is implemented. If it can integrate the problem, it was assigned a B grade automatically as a placeholder. In the future, when grading function is implemented for Mupad, the tests will be rerun again.

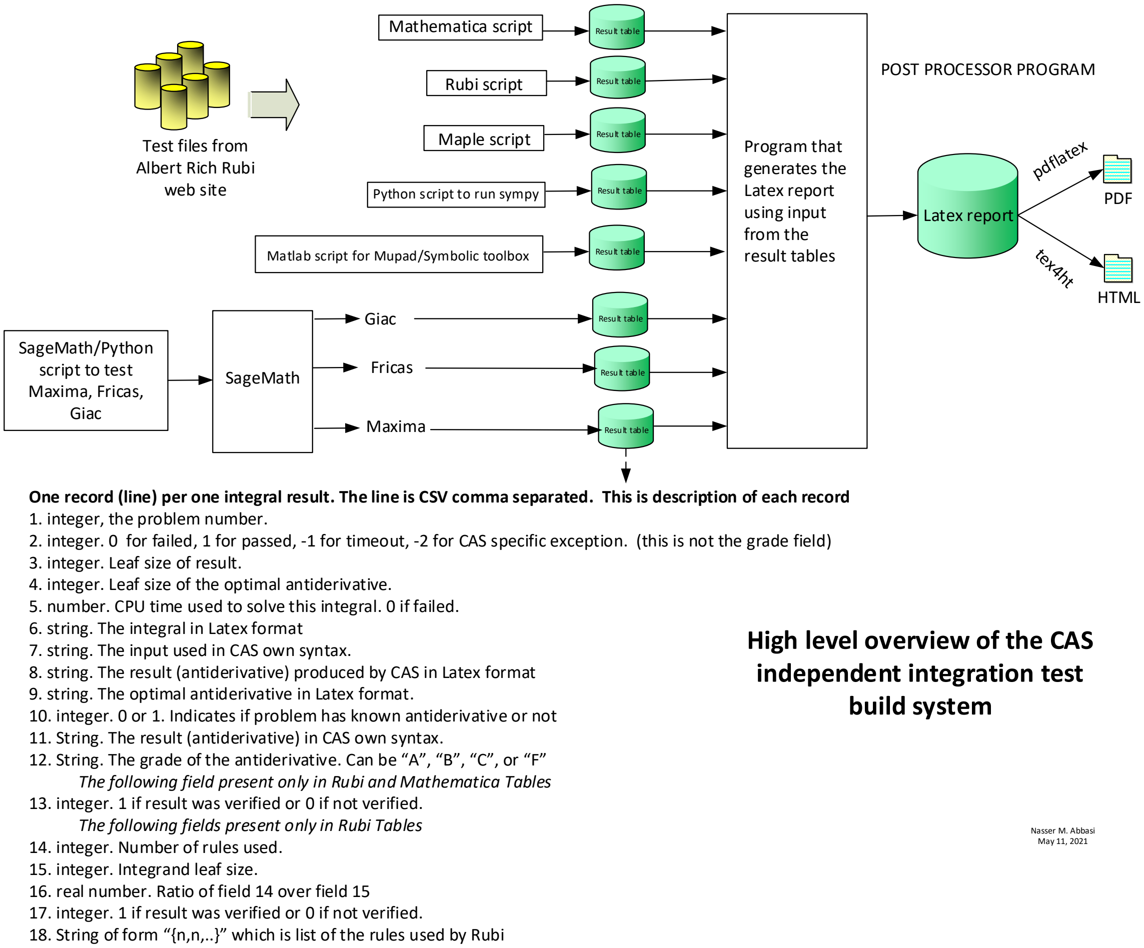
The following is an example of using Matlab's symbolic toolbox (Mupad) to solve an integral

```
integrand = evalin(symengine,'cos(x)*sin(x)')
the_variable = evalin(symengine,'x')
anti = int(integrand,the_variable)
```

Which gives `sin(x)^2/2`

1.10 Design of the test system

The following diagram gives a high level view of the current test build system.



Chapter 2

detailed summary tables of results

2.1 List of integrals sorted by grade for each CAS

2.1.1 Rubi

A grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171 }

B grade: { }

C grade: { }

F grade: { }

2.1.2 Mathematica

A grade: { 1, 2, 3, 4, 28, 45, 46, 47, 48, 53, 69, 74, 75, 76, 82, 84, 88, 91, 92, 93, 94, 98, 99, 100, 101, 104, 105, 106, 107, 111, 112, 113, 114, 115, 120, 128, 129, 130, 131, 133, 134, 135, 136, 137, 141, 142, 147, 148, 149, 150, 151, 152, 156, 157, 158, 161, 162, 163, 166, 167, 168, 169, 170 }

B grade: { 81, 83, 89, 90, 95, 96, 97, 102, 103, 108, 109, 110, 121, 126, 127, 140, 165, 171 }

C grade: { 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 50, 51, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 70, 71, 72, 73, 77, 78, 79, 80, 85, 86, 87, 116, 117, 118, 119, 122, 123, 124, 125, 132, 138, 139, 143, 144, 145, 146, 153, 154, 155, 159, 160 }

F grade: { 49, 164 }

2.1.3 Maple

A grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 31, 32, 33, 38, 53 }

B grade: { 28, 29, 30, 34, 35, 36, 37, 39, 40, 41, 42, 43, 44, 50, 51, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127 }

C grade: { }

F grade: { 45, 46, 47, 48, 49, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171 }

2.1.4 Maxima

A grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 77, 78, 79, 80, 81, 84, 85, 86, 87 }

B grade: { 76, 82, 83, 88, 89 }

C grade: { }

F grade: { 45, 46, 47, 48, 49, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171 }

2.1.5 FriCAS

A grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 34, 35, 50, 51, 52, 53, 54, 57, 58, 59, 60, 61, 64, 65, 66, 67, 70, 71, 72, 73, 74, 79, 80 }

B grade: { 32, 33, 36, 37, 38, 39, 40, 41, 42, 43, 44, 55, 56, 62, 63, 68, 69, 75, 76, 77, 78, 81, 82, 83, 84, 85, 86, 87, 88, 89 }

C grade: { }

F grade: { 45, 46, 47, 48, 49, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171 }

2.1.6 Sympy

A grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 50, 51, 52, 53, 54, 55, 57, 58, 59, 60, 61, 64, 65, 66, 67, 70, 71, 72, 73, 79, 80 }

B grade: { }

C grade: { }

F grade: { 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 56, 62, 63, 68, 69, 74, 75, 76, 77, 78, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171 }

2.1.7 Giac

A grade: { 3, 4, 11, 12, 13, 18, 19, 20, 21, 25, 26, 27, 28, 29, 30, 31, 32, 33, 37, 38, 39, 44, 54, 61, 67, 70, 71, 72, 73, 74, 79 }

B grade: { 1, 2, 5, 6, 7, 8, 9, 10, 14, 15, 16, 17, 22, 23, 24, 34, 35, 36, 40, 41, 42, 43, 51, 52, 53, 55, 56, 59, 60, 62, 63, 66, 68, 69, 75, 76, 77, 78, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89 }

C grade: { }

F grade: { 45, 46, 47, 48, 49, 50, 57, 58, 64, 65, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171 }

2.1.8 Mupad

A grade: { }

B grade: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 92, 93, 94, 95, 100, 101, 106, 110, 111, 112, 113, 114, 115, 117, 118, 119, 123, 124, 125 }

C grade: { }

F grade: { 45, 46, 47, 48, 49, 90, 91, 96, 97, 98, 99, 102, 103, 104, 105, 107, 108, 109, 116, 120, 121, 122, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171 }

2.2 Detailed conclusion table per each integral for all CAS systems

Detailed conclusion table per each integral is given by table below. The elapsed time is in seconds. For failed result it is given as F(-1) if the failure was due to timeout. It is given as F(-2) if the failure was due to an exception being raised, which could indicate a bug in the system. If the failure was due to integral not being evaluated within the time limit, then it is given just an F.

In this table,the column **normalized size** is defined as $\frac{\text{antiderivative leaf size}}{\text{optimal antiderivative leaf size}}$

Problem 1	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	87	87	86	135	86	85	139	1017	84
normalized size	1	1.00	0.99	1.55	0.99	0.98	1.60	11.69	0.97
time (sec)	N/A	0.134	0.598	0.030	0.536	0.598	0.374	3.958	8.825

Problem 2	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	66	66	67	105	66	66	105	616	63
normalized size	1	1.00	1.02	1.59	1.00	1.00	1.59	9.33	0.95
time (sec)	N/A	0.046	0.320	0.024	0.655	0.609	0.244	2.922	8.838

Problem 3	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	42	42	59	66	50	50	82	50	58
normalized size	1	1.00	1.40	1.57	1.19	1.19	1.95	1.19	1.38
time (sec)	N/A	0.060	0.056	0.386	0.598	1.060	0.647	2.267	8.791

Problem 4	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	37	37	44	51	52	59	85	53	69
normalized size	1	1.00	1.19	1.38	1.41	1.59	2.30	1.43	1.86
time (sec)	N/A	0.110	0.072	0.569	0.594	0.461	0.979	3.668	8.956

Problem 5	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	43	43	78	65	68	73	116	119	87
normalized size	1	1.00	1.81	1.51	1.58	1.70	2.70	2.77	2.02
time (sec)	N/A	0.124	0.159	0.436	0.462	1.550	1.664	4.314	8.875

Problem 6	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	66	66	77	96	86	95	150	179	108
normalized size	1	1.00	1.17	1.45	1.30	1.44	2.27	2.71	1.64
time (sec)	N/A	0.159	0.469	0.523	0.663	0.547	2.335	5.637	8.944

Problem 7	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	87	87	101	124	104	121	180	237	127
normalized size	1	1.00	1.16	1.43	1.20	1.39	2.07	2.72	1.46
time (sec)	N/A	0.193	1.029	0.513	0.765	0.540	4.429	7.625	8.888

Problem 8	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	108	108	100	150	122	138	211	299	145
normalized size	1	1.00	0.93	1.39	1.13	1.28	1.95	2.77	1.34
time (sec)	N/A	0.226	1.147	0.532	0.926	0.694	5.868	9.317	8.821

Problem 9	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	148	148	221	249	147	146	250	2228	151
normalized size	1	1.00	1.49	1.68	0.99	0.99	1.69	15.05	1.02
time (sec)	N/A	0.302	6.253	0.025	0.626	0.598	0.600	14.820	8.844

Problem 10	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	112	112	172	199	120	119	194	1509	121
normalized size	1	1.00	1.54	1.78	1.07	1.06	1.73	13.47	1.08
time (sec)	N/A	0.112	1.847	0.028	0.797	0.666	0.435	5.941	8.795

Problem 11	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	87	87	96	140	91	91	151	95	91
normalized size	1	1.00	1.10	1.61	1.05	1.05	1.74	1.09	1.05
time (sec)	N/A	0.135	0.470	0.496	0.801	1.515	1.086	4.426	8.848

Problem 12	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	70	70	91	109	85	92	136	86	90
normalized size	1	1.00	1.30	1.56	1.21	1.31	1.94	1.23	1.29
time (sec)	N/A	0.185	0.284	0.481	0.984	0.766	1.612	7.505	8.853

Problem 13	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	72	72	100	110	93	112	158	118	100
normalized size	1	1.00	1.39	1.53	1.29	1.56	2.19	1.64	1.39
time (sec)	N/A	0.207	0.252	0.515	0.736	0.775	2.298	6.394	8.999

Problem 14	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	88	88	123	141	120	122	212	237	127
normalized size	1	1.00	1.40	1.60	1.36	1.39	2.41	2.69	1.44
time (sec)	N/A	0.263	0.353	0.624	0.703	1.546	4.312	8.117	8.979

Problem 15	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	118	118	152	188	149	157	258	334	156
normalized size	1	1.00	1.29	1.59	1.26	1.33	2.19	2.83	1.32
time (sec)	N/A	0.311	1.175	0.460	0.748	0.620	5.679	11.085	9.078

Problem 16	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	151	151	180	238	175	191	311	435	182
normalized size	1	1.00	1.19	1.58	1.16	1.26	2.06	2.88	1.21
time (sec)	N/A	0.369	2.931	0.562	0.864	1.239	8.801	21.201	8.860

Problem 17	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	165	165	209	314	179	178	313	2870	181
normalized size	1	1.00	1.27	1.90	1.08	1.08	1.90	17.39	1.10
time (sec)	N/A	0.177	1.629	0.030	0.520	0.592	0.667	12.648	8.835

Problem 18	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	140	140	130	234	143	142	248	158	142
normalized size	1	1.00	0.93	1.67	1.02	1.01	1.77	1.13	1.01
time (sec)	N/A	0.208	1.070	0.473	1.019	0.683	1.817	5.958	8.963

Problem 19	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	117	117	113	183	124	133	211	129	118
normalized size	1	1.00	0.97	1.56	1.06	1.14	1.80	1.10	1.01
time (sec)	N/A	0.336	0.471	0.566	0.600	0.647	2.322	8.844	8.963

Problem 20	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	119	119	113	168	125	145	214	152	114
normalized size	1	1.00	0.95	1.41	1.05	1.22	1.80	1.28	0.96
time (sec)	N/A	0.331	0.493	0.452	0.784	0.649	4.424	12.044	8.860

Problem 21	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	127	127	126	186	142	162	260	193	135
normalized size	1	1.00	0.99	1.46	1.12	1.28	2.05	1.52	1.06
time (sec)	N/A	0.355	0.462	0.570	0.603	0.918	5.602	22.948	8.967

Problem 22	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	154	154	164	233	180	181	330	390	169
normalized size	1	1.00	1.06	1.51	1.17	1.18	2.14	2.53	1.10
time (sec)	N/A	0.427	1.275	0.521	0.574	0.625	8.559	27.330	8.999

Problem 23	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	191	191	199	302	215	225	398	528	204
normalized size	1	1.00	1.04	1.58	1.13	1.18	2.08	2.76	1.07
time (sec)	N/A	0.514	0.790	0.551	0.738	0.744	11.005	90.465	8.941

Problem 24	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	233	233	237	376	250	266	469	670	238
normalized size	1	1.00	1.02	1.61	1.07	1.14	2.01	2.88	1.02
time (sec)	N/A	0.558	1.210	0.536	0.764	1.406	27.646	61.538	9.120

Problem 25	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	127	127	138	211	130	190	1309	135	144
normalized size	1	1.00	1.09	1.66	1.02	1.50	10.31	1.06	1.13
time (sec)	N/A	0.468	1.450	0.255	0.715	0.874	2.030	2.111	9.071

Problem 26	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	101	101	118	179	109	149	1020	110	117
normalized size	1	1.00	1.17	1.77	1.08	1.48	10.10	1.09	1.16
time (sec)	N/A	0.243	0.700	0.255	0.549	0.667	1.454	1.809	8.768

Problem 27	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	85	85	98	159	94	110	724	95	100
normalized size	1	1.00	1.15	1.87	1.11	1.29	8.52	1.12	1.18
time (sec)	N/A	0.163	0.198	0.276	0.774	2.313	1.122	1.523	9.065

Problem 28	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	58	58	67	153	88	76	541	94	93
normalized size	1	1.00	1.16	2.64	1.52	1.31	9.33	1.62	1.60
time (sec)	N/A	0.144	0.126	0.732	0.635	0.751	2.954	2.153	9.125

Problem 29	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	80	80	113	174	107	118	966	113	115
normalized size	1	1.00	1.41	2.18	1.34	1.48	12.08	1.41	1.44
time (sec)	N/A	0.201	0.366	0.936	0.465	0.667	5.746	4.111	9.457

Problem 30	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	103	103	138	214	131	177	2064	157	140
normalized size	1	1.00	1.34	2.08	1.27	1.72	20.04	1.52	1.36
time (sec)	N/A	0.342	0.888	0.756	0.967	0.769	12.101	4.810	10.339

Problem 31	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	137	137	163	266	158	234	2621	214	175
normalized size	1	1.00	1.19	1.94	1.15	1.71	19.13	1.56	1.28
time (sec)	N/A	0.682	1.422	0.981	0.617	0.634	33.891	5.475	10.929

Problem 32	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	B	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	208	208	444	364	220	434	4602	290	210
normalized size	1	1.00	2.13	1.75	1.06	2.09	22.12	1.39	1.01
time (sec)	N/A	0.532	4.265	0.252	0.573	0.805	2.978	2.936	9.648

Problem 33	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	B	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	157	157	324	313	197	311	3497	244	165
normalized size	1	1.00	2.06	1.99	1.25	1.98	22.27	1.55	1.05
time (sec)	N/A	0.311	2.178	0.316	0.613	0.823	2.289	2.015	9.107

Problem 34	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	115	115	140	305	185	221	2995	241	163
normalized size	1	1.00	1.22	2.65	1.61	1.92	26.04	2.10	1.42
time (sec)	N/A	0.147	2.190	0.307	0.636	0.594	1.839	1.985	9.011

Problem 35	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	111	111	190	301	177	222	2895	234	153
normalized size	1	1.00	1.71	2.71	1.59	2.00	26.08	2.11	1.38
time (sec)	N/A	0.208	2.224	0.765	0.573	0.804	4.995	2.613	9.094

Problem 36	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	137	137	159	325	208	323	4461	279	180
normalized size	1	1.00	1.16	2.37	1.52	2.36	32.56	2.04	1.31
time (sec)	N/A	0.403	2.411	0.837	1.329	0.920	9.510	5.292	10.693

Problem 37	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	192	192	193	399	262	465	8097	362	230
normalized size	1	1.00	1.01	2.08	1.36	2.42	42.17	1.89	1.20
time (sec)	N/A	0.608	3.567	0.856	0.823	0.910	15.712	6.712	12.148

Problem 38	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	A	A	B	F(-2)	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	331	331	1146	619	389	890	0	505	335
normalized size	1	1.00	3.46	1.87	1.18	2.69	0.00	1.53	1.01
time (sec)	N/A	0.861	6.856	0.283	1.040	0.927	0.000	4.343	10.428

Problem 39	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	250	250	462	566	366	666	0	458	307
normalized size	1	1.00	1.85	2.26	1.46	2.66	0.00	1.83	1.23
time (sec)	N/A	0.581	4.918	0.283	0.649	1.460	0.000	3.009	9.316

Problem 40	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	189	189	288	495	333	478	0	410	280
normalized size	1	1.00	1.52	2.62	1.76	2.53	0.00	2.17	1.48
time (sec)	N/A	0.427	5.453	0.342	0.871	0.592	0.000	2.269	9.181

Problem 41	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	179	179	188	488	330	488	0	410	282
normalized size	1	1.00	1.05	2.73	1.84	2.73	0.00	2.29	1.58
time (sec)	N/A	0.255	4.066	0.287	0.638	0.897	0.000	2.647	9.280

Problem 42	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	175	175	243	483	321	482	0	409	279
normalized size	1	1.00	1.39	2.76	1.83	2.75	0.00	2.34	1.59
time (sec)	N/A	0.316	4.771	0.740	0.476	0.785	0.000	4.717	8.941

Problem 43	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	215	215	223	540	372	683	0	479	315
normalized size	1	1.00	1.04	2.51	1.73	3.18	0.00	2.23	1.47
time (sec)	N/A	0.680	3.151	1.059	0.935	0.726	0.000	8.669	10.976

Problem 44	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	287	287	288	651	454	917	0	560	380
normalized size	1	1.00	1.00	2.27	1.58	3.20	0.00	1.95	1.32
time (sec)	N/A	0.941	6.425	0.930	0.722	0.890	0.000	9.825	13.986

Problem 45	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F	F	F	F	F	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	132	132	110	0	0	0	0	0	-1
normalized size	1	1.00	0.83	0.00	0.00	0.00	0.00	0.00	-0.01
time (sec)	N/A	0.155	0.408	1.112	0.000	1.122	0.000	0.000	0.000

Problem 46	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F	F	F	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	154	154	115	0	0	0	0	0	-1
normalized size	1	1.00	0.75	0.00	0.00	0.00	0.00	0.00	-0.01
time (sec)	N/A	0.138	0.393	180.000	0.000	0.760	0.000	0.000	0.000

Problem 47	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F	F(-1)	F	F	F(-1)	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	170	170	133	0	0	0	0	0	-1
normalized size	1	1.00	0.78	0.00	0.00	0.00	0.00	0.00	-0.01
time (sec)	N/A	0.144	0.533	1.706	0.000	0.552	0.000	0.000	0.000

Problem 48	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F	F(-1)	F	F	F	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	170	170	133	0	0	0	0	0	-1
normalized size	1	1.00	0.78	0.00	0.00	0.00	0.00	0.00	-0.01
time (sec)	N/A	0.137	0.537	1.761	0.000	0.664	0.000	0.000	0.000

Problem 49	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	F	F	F	F	F	F(-1)	F
verified	N/A	Yes	N/A	TBD	TBD	TBD	TBD	TBD	TBD
size	328	328	0	0	0	0	0	0	-1
normalized size	1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.561	28.559	2.042	0.000	0.819	0.000	0.000	0.000

Problem 50	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	353	353	300	994	416	415	1001	0	477
normalized size	1	1.00	0.85	2.82	1.18	1.18	2.84	0.00	1.35
time (sec)	N/A	0.785	6.382	0.032	0.544	1.038	1.648	0.000	8.997

Problem 51	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	248	248	243	631	274	273	617	6502	300
normalized size	1	1.00	0.98	2.54	1.10	1.10	2.49	26.22	1.21
time (sec)	N/A	0.451	3.434	0.028	0.566	0.572	0.981	23.118	8.982

Problem 52	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	161	161	161	334	151	150	326	2918	153
normalized size	1	1.00	1.00	2.07	0.94	0.93	2.02	18.12	0.95
time (sec)	N/A	0.241	1.617	0.025	0.445	0.588	0.500	7.336	8.842

Problem 53	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	A	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	73	73	76	136	74	74	131	918	75
normalized size	1	1.00	1.04	1.86	1.01	1.01	1.79	12.58	1.03
time (sec)	N/A	0.061	0.470	0.025	0.508	1.329	0.278	2.929	8.679

Problem 54	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	156	155	148	506	183	226	2429	186	186
normalized size	1	0.99	0.95	3.24	1.17	1.45	15.57	1.19	1.19
time (sec)	N/A	0.349	1.193	0.244	0.440	1.441	2.370	1.793	10.127

Problem 55	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	265	265	589	948	338	556	9721	531	1875
normalized size	1	1.00	2.22	3.58	1.28	2.10	36.68	2.00	7.08
time (sec)	N/A	0.474	6.864	0.296	0.460	1.342	3.952	2.355	21.136

Problem 56	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	320	320	379	1513	574	987	0	1037	502
normalized size	1	1.00	1.18	4.73	1.79	3.08	0.00	3.24	1.57
time (sec)	N/A	0.703	6.382	0.332	0.643	2.056	0.000	3.169	15.885

Problem 57	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	661	661	573	1807	691	690	1819	0	891
normalized size	1	1.00	0.87	2.73	1.05	1.04	2.75	0.00	1.35
time (sec)	N/A	2.384	6.646	0.034	0.632	1.236	3.184	0.000	9.287

Problem 58	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	443	443	383	1165	463	462	1134	0	561
normalized size	1	1.00	0.86	2.63	1.05	1.04	2.56	0.00	1.27
time (sec)	N/A	1.278	6.498	0.031	0.460	1.155	1.913	0.000	9.119

Problem 59	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	266	264	241	631	260	259	617	6502	300
normalized size	1	0.99	0.91	2.37	0.98	0.97	2.32	24.44	1.13
time (sec)	N/A	0.472	2.887	0.030	0.554	0.599	0.975	33.119	9.006

Problem 60	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	131	131	176	262	135	134	241	2128	141
normalized size	1	1.00	1.34	2.00	1.03	1.02	1.84	16.24	1.08
time (sec)	N/A	0.155	1.183	0.026	0.501	0.557	0.467	5.570	8.808

Problem 61	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	254	252	190	861	290	397	4517	338	325
normalized size	1	0.99	0.75	3.39	1.14	1.56	17.78	1.33	1.28
time (sec)	N/A	0.827	3.030	0.239	0.536	1.650	8.060	3.657	11.275

Problem 62	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-1)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	415	415	2640	1554	496	964	0	912	3958
normalized size	1	1.00	6.36	3.74	1.20	2.32	0.00	2.20	9.54
time (sec)	N/A	1.053	8.020	0.292	0.489	3.000	0.000	3.065	34.031

Problem 63	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	597	597	2499	2465	839	1699	0	1714	807
normalized size	1	1.00	4.19	4.13	1.41	2.85	0.00	2.87	1.35
time (sec)	N/A	1.290	8.161	0.334	0.566	2.651	0.000	2.698	29.277

Problem 64	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	603	603	419	1807	680	679	1819	0	891
normalized size	1	1.00	0.69	3.00	1.13	1.13	3.02	0.00	1.48
time (sec)	N/A	1.533	6.574	0.033	0.480	0.643	3.221	0.000	9.310

Problem 65	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	389	387	297	994	387	386	1001	0	478
normalized size	1	0.99	0.76	2.56	0.99	0.99	2.57	0.00	1.23
time (sec)	N/A	0.705	6.332	0.036	0.449	1.481	1.654	0.000	9.038

Problem 66	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	191	191	212	420	202	201	410	4300	221
normalized size	1	1.00	1.11	2.20	1.06	1.05	2.15	22.51	1.16
time (sec)	N/A	0.244	2.436	0.031	0.580	1.574	0.762	24.807	8.789

Problem 67	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	363	363	255	1304	436	623	7205	573	508
normalized size	1	1.00	0.70	3.59	1.20	1.72	19.85	1.58	1.40
time (sec)	N/A	1.512	4.830	0.253	0.541	3.139	113.330	3.407	13.004

Problem 68	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-1)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	574	574	2467	2250	685	1512	0	1357	701
normalized size	1	1.00	4.30	3.92	1.19	2.63	0.00	2.36	1.22
time (sec)	N/A	2.321	8.577	0.296	0.588	3.086	0.000	7.279	15.699

Problem 69	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	A	B	F(-2)	B	B
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	798	798	1451	3522	1119	2549	0	2505	1172
normalized size	1	1.00	1.82	4.41	1.40	3.19	0.00	3.14	1.47
time (sec)	N/A	2.839	15.892	0.316	0.550	4.314	0.000	7.349	19.238

Problem 70	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	337	337	258	1304	445	627	7205	573	508
normalized size	1	1.00	0.77	3.87	1.32	1.86	21.38	1.70	1.51
time (sec)	N/A	1.588	4.515	0.266	0.585	2.732	115.152	5.522	13.392

Problem 71	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	236	236	190	861	294	390	4517	338	325
normalized size	1	1.00	0.81	3.65	1.25	1.65	19.14	1.43	1.38
time (sec)	N/A	0.804	2.990	0.291	0.550	1.174	8.128	3.056	11.200

Problem 72	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	156	156	148	506	178	212	2429	186	186
normalized size	1	1.00	0.95	3.24	1.14	1.36	15.57	1.19	1.19
time (sec)	N/A	0.342	1.086	0.252	1.490	0.846	2.412	1.929	10.246

Problem 73	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	99	99	117	234	106	118	984	109	109
normalized size	1	1.00	1.18	2.36	1.07	1.19	9.94	1.10	1.10
time (sec)	N/A	0.098	0.197	0.255	0.555	1.177	1.305	2.057	9.898

Problem 74	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	A	A	F(-1)	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	165	164	313	647	243	301	0	272	196
normalized size	1	0.99	1.90	3.92	1.47	1.82	0.00	1.65	1.19
time (sec)	N/A	0.256	1.505	0.517	0.467	0.922	0.000	2.334	21.398

Problem 75	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	281	281	543	1262	520	1345	0	846	393
normalized size	1	1.00	1.93	4.49	1.85	4.79	0.00	3.01	1.40
time (sec)	N/A	0.795	6.957	0.548	0.510	2.537	0.000	6.617	63.656

Problem 76	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	B	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	477	477	898	2298	1096	3643	0	2127	65819
normalized size	1	1.00	1.88	4.82	2.30	7.64	0.00	4.46	137.99
time (sec)	N/A	1.786	9.028	0.523	0.677	7.175	0.000	26.162	24.034

Problem 77	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-1)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	579	579	2463	2250	684	1477	0	1355	701
normalized size	1	1.00	4.25	3.89	1.18	2.55	0.00	2.34	1.21
time (sec)	N/A	2.134	8.548	0.286	0.731	2.368	0.000	3.653	16.677

Problem 78	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-1)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	417	417	2636	1554	493	939	0	912	3958
normalized size	1	1.00	6.32	3.73	1.18	2.25	0.00	2.19	9.49
time (sec)	N/A	1.113	7.943	0.267	0.538	1.278	0.000	3.090	35.258

Problem 79	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	A	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	292	288	606	948	319	505	9721	528	1875
normalized size	1	0.99	2.08	3.25	1.09	1.73	33.29	1.81	6.42
time (sec)	N/A	0.554	6.783	0.270	0.496	0.647	4.099	6.681	22.014

Problem 80	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	A	A	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	140	140	207	438	205	256	4396	299	184
normalized size	1	1.00	1.48	3.13	1.46	1.83	31.40	2.14	1.31
time (sec)	N/A	0.209	2.550	0.296	0.666	0.575	2.132	3.991	11.345

Problem 81	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	293	293	592	1263	513	1275	0	846	430
normalized size	1	1.00	2.02	4.31	1.75	4.35	0.00	2.89	1.47
time (sec)	N/A	0.812	7.504	0.547	0.505	2.708	0.000	5.673	85.865

Problem 82	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	B	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	509	508	984	2012	1185	4174	0	2893	73684
normalized size	1	1.00	1.93	3.95	2.33	8.20	0.00	5.68	144.76
time (sec)	N/A	2.151	8.908	0.481	0.761	6.551	0.000	4.534	31.511

Problem 83	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	B	B	F(-2)	B	B
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	841	841	1758	3364	2519	9594	0	3176	128667
normalized size	1	1.00	2.09	4.00	3.00	11.41	0.00	3.78	152.99
time (sec)	N/A	4.076	8.622	0.623	0.704	17.356	0.000	25.978	58.468

Problem 84	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	A	B	F(-2)	B	B
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	804	804	1445	3522	1110	2490	0	2505	1172
normalized size	1	1.00	1.80	4.38	1.38	3.10	0.00	3.12	1.46
time (sec)	N/A	2.747	15.598	0.294	0.602	2.647	0.000	5.088	20.600

Problem 85	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	597	597	2499	2465	827	1618	0	1709	807
normalized size	1	1.00	4.19	4.13	1.39	2.71	0.00	2.86	1.35
time (sec)	N/A	1.385	8.127	0.320	0.582	1.231	0.000	4.651	30.686

Problem 86	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	352	349	378	1513	543	897	0	1037	502
normalized size	1	0.99	1.07	4.30	1.54	2.55	0.00	2.95	1.43
time (sec)	N/A	0.711	6.344	0.333	0.604	0.537	0.000	2.699	16.535

Problem 87	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	A	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	209	209	261	713	367	566	0	548	327
normalized size	1	1.00	1.25	3.41	1.76	2.71	0.00	2.62	1.56
time (sec)	N/A	0.376	5.279	0.292	0.644	0.717	0.000	1.386	11.877

Problem 88	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	B	B	F(-2)	B	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	487	487	912	2298	1078	3496	0	2125	65817
normalized size	1	1.00	1.87	4.72	2.21	7.18	0.00	4.36	135.15
time (sec)	N/A	1.830	9.244	0.528	0.560	6.767	0.000	17.821	24.606

Problem 89	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	B	B	F(-2)	B	B
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	861	860	1732	3364	2537	9567	0	3176	128666
normalized size	1	1.00	2.01	3.91	2.95	11.11	0.00	3.69	149.44
time (sec)	N/A	4.276	8.740	0.620	0.787	20.867	0.000	90.117	47.926

Problem 90	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	464	464	1232	6661	0	0	0	0	-1
normalized size	1	1.00	2.66	14.36	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.089	6.392	0.618	0.000	0.000	0.000	0.000	0.000

Problem 91	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	325	325	314	4775	0	0	0	0	-1
normalized size	1	1.00	0.97	14.69	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.306	4.818	0.455	0.000	0.000	0.000	0.000	0.000

Problem 92	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	224	224	220	3028	0	0	0	0	22955
normalized size	1	1.00	0.98	13.52	0.00	0.00	0.00	0.00	102.48
time (sec)	N/A	0.628	1.991	0.500	0.000	0.000	0.000	0.000	60.113

Problem 93	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	155	155	150	1472	0	0	0	0	1199
normalized size	1	1.00	0.97	9.50	0.00	0.00	0.00	0.00	7.74
time (sec)	N/A	0.306	0.558	0.436	0.000	0.000	0.000	0.000	17.403

Problem 94	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-2)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	234	234	233	3576	0	0	0	0	62245
normalized size	1	1.00	1.00	15.28	0.00	0.00	0.00	0.00	266.00
time (sec)	N/A	1.087	0.689	0.749	0.000	0.000	0.000	0.000	36.224

Problem 95	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-2)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	317	317	764	5778	0	0	0	0	138318
normalized size	1	1.00	2.41	18.23	0.00	0.00	0.00	0.00	436.33
time (sec)	N/A	1.439	6.386	0.873	0.000	0.000	0.000	0.000	45.420

Problem 96	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	543	543	2819	9797	0	0	0	0	-1
normalized size	1	1.00	5.19	18.04	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	4.037	6.382	0.881	0.000	0.000	0.000	0.000	0.000

Problem 97	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	550	550	1290	11056	0	0	0	0	-1
normalized size	1	1.00	2.35	20.10	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.734	6.414	0.640	0.000	0.000	0.000	0.000	0.000

Problem 98	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	396	396	350	8031	0	0	0	0	-1
normalized size	1	1.00	0.88	20.28	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.727	6.198	0.603	0.000	0.000	0.000	0.000	0.000

Problem 99	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	273	273	260	5149	0	0	0	0	-1
normalized size	1	1.00	0.95	18.86	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	0.879	4.441	0.546	0.000	0.000	0.000	0.000	0.000

Problem 100	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	187	187	202	2517	0	0	0	0	4260
normalized size	1	1.00	1.08	13.46	0.00	0.00	0.00	0.00	22.78
time (sec)	N/A	0.460	1.235	0.407	0.000	0.000	0.000	0.000	44.865

Problem 101	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-2)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	271	271	266	6055	0	0	0	0	106783
normalized size	1	1.00	0.98	22.34	0.00	0.00	0.00	0.00	394.03
time (sec)	N/A	1.814	2.552	0.830	0.000	0.000	0.000	0.000	58.881

Problem 102	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	372	372	1732	9865	0	0	0	0	-1
normalized size	1	1.00	4.66	26.52	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.547	6.313	0.831	0.000	0.000	0.000	0.000	0.000

Problem 103	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	532	532	7678	14441	0	0	0	0	-1
normalized size	1	1.00	14.43	27.14	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	4.087	6.564	0.829	0.000	0.000	0.000	0.000	0.000

Problem 104	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	503	503	564	11478	0	0	0	0	-1
normalized size	1	1.00	1.12	22.82	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.312	6.450	0.595	0.000	0.000	0.000	0.000	0.000

Problem 105	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	353	351	324	7402	0	0	0	0	-1
normalized size	1	0.99	0.92	20.97	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.212	5.359	0.519	0.000	0.000	0.000	0.000	0.000

Problem 106	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	229	229	262	3614	0	0	0	0	5863
normalized size	1	1.00	1.14	15.78	0.00	0.00	0.00	0.00	25.60
time (sec)	N/A	0.629	2.066	0.408	0.000	0.000	0.000	0.000	117.306

Problem 107	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-2)	F(-1)	F(-1)	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	336	336	322	8698	0	0	0	0	-1
normalized size	1	1.00	0.96	25.89	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.812	5.306	0.842	0.000	0.000	0.000	0.000	0.000

Problem 108	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-2)	F(-1)	F(-1)	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	473	473	6112	14119	0	0	0	0	-1
normalized size	1	1.00	12.92	29.85	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	3.896	6.532	0.874	0.000	0.000	0.000	0.000	0.000

Problem 109	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-2)	F(-1)	F(-1)	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	643	643	18214	20663	0	0	0	0	-1
normalized size	1	1.00	28.33	32.14	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	6.065	6.891	0.839	0.000	0.000	0.000	0.000	0.000

Problem 110	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	407	407	1200	25426	0	0	0	0	28858
normalized size	1	1.00	2.95	62.47	0.00	0.00	0.00	0.00	70.90
time (sec)	N/A	1.699	6.432	0.465	0.000	0.000	0.000	0.000	122.079

Problem 111	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	287	287	275	18289	0	0	0	0	21254
normalized size	1	1.00	0.96	63.72	0.00	0.00	0.00	0.00	74.06
time (sec)	N/A	1.001	6.032	0.412	0.000	0.000	0.000	0.000	47.981

Problem 112	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	194	194	192	4138	0	0	0	0	16400
normalized size	1	1.00	0.99	21.33	0.00	0.00	0.00	0.00	84.54
time (sec)	N/A	0.498	1.457	0.396	0.000	0.000	0.000	0.000	23.482

Problem 113	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	133	133	129	5570	0	0	0	0	4326
normalized size	1	1.00	0.97	41.88	0.00	0.00	0.00	0.00	32.53
time (sec)	N/A	0.216	0.216	0.372	0.000	0.000	0.000	0.000	14.206

Problem 114	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-2)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	210	210	194	13474	0	0	0	0	25341
normalized size	1	1.00	0.92	64.16	0.00	0.00	0.00	0.00	120.67
time (sec)	N/A	0.615	0.418	0.522	0.000	0.000	0.000	0.000	69.145

Problem 115	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-2)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	327	327	521	20870	0	0	0	0	225004
normalized size	1	1.00	1.59	63.82	0.00	0.00	0.00	0.00	688.09
time (sec)	N/A	1.379	6.217	0.643	0.000	0.000	0.000	0.000	57.653

Problem 116	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	511	511	920	49725	0	0	0	0	-1
normalized size	1	1.00	1.80	97.31	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.465	6.787	0.578	0.000	0.000	0.000	0.000	0.000

Problem 117	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	343	343	476	36710	0	0	0	0	54886
normalized size	1	1.00	1.39	107.03	0.00	0.00	0.00	0.00	160.02
time (sec)	N/A	1.354	6.510	0.464	0.000	0.000	0.000	0.000	66.251

Problem 118	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	201	201	290	23472	0	0	0	0	40542
normalized size	1	1.00	1.44	116.78	0.00	0.00	0.00	0.00	201.70
time (sec)	N/A	0.554	2.591	0.436	0.000	0.000	0.000	0.000	41.071

Problem 119	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	157	157	218	11427	0	0	0	0	8588
normalized size	1	1.00	1.39	72.78	0.00	0.00	0.00	0.00	54.70
time (sec)	N/A	0.294	1.011	0.360	0.000	0.000	0.000	0.000	19.614

Problem 120	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	262	262	296	26343	0	0	0	0	-1
normalized size	1	1.00	1.13	100.55	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.277	4.950	0.625	0.000	0.000	0.000	0.000	0.000

Problem 121	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	447	446	2078	40619	0	0	0	0	-1
normalized size	1	1.00	4.65	90.87	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.881	6.286	0.719	0.000	0.000	0.000	0.000	0.000

Problem 122	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	585	585	670	85156	0	0	0	0	-1
normalized size	1	1.00	1.15	145.57	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.968	6.906	0.615	0.000	0.000	0.000	0.000	0.000

Problem 123	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	358	358	502	61833	0	0	0	0	88684
normalized size	1	1.00	1.40	172.72	0.00	0.00	0.00	0.00	247.72
time (sec)	N/A	1.551	6.562	0.519	0.000	0.000	0.000	0.000	116.899

Problem 124	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	273	271	300	40201	0	0	0	0	64641
normalized size	1	0.99	1.10	147.26	0.00	0.00	0.00	0.00	236.78
time (sec)	N/A	0.798	3.020	0.611	0.000	0.000	0.000	0.000	88.469

Problem 125	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	B	F(-1)	F(-1)	F	F(-1)	B
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	209	209	223	20647	0	0	0	0	14163
normalized size	1	1.00	1.07	98.79	0.00	0.00	0.00	0.00	67.77
time (sec)	N/A	0.486	0.938	0.395	0.000	0.000	0.000	0.000	37.590

Problem 126	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	365	365	1948	45119	0	0	0	0	-1
normalized size	1	1.00	5.34	123.61	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.466	6.276	0.674	0.000	0.000	0.000	0.000	0.000

Problem 127	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	B	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	679	678	6052	67570	0	0	0	0	-1
normalized size	1	1.00	8.91	99.51	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	5.062	6.427	0.850	0.000	0.000	0.000	0.000	0.000

Problem 128	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F(-1)	F(-1)	F
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	679	679	1202	0	0	0	0	0	-1
normalized size	1	1.00	1.77	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	9.926	9.951	180.000	0.000	0.000	0.000	0.000	0.000

Problem 129	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F(-1)	F	F(-1)	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	505	505	835	0	0	0	0	0	-1
normalized size	1	1.00	1.65	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	7.338	8.876	180.000	0.000	0.000	0.000	0.000	0.000

Problem 130	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	381	383	619	0	0	0	0	0	-1
normalized size	1	1.01	1.62	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	4.973	7.787	180.000	0.000	0.000	0.000	0.000	0.000

Problem 131	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	287	287	441	0	0	0	0	0	-1
normalized size	1	1.00	1.54	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.633	4.244	180.000	0.000	0.000	0.000	0.000	0.000

Problem 132	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	300	300	621058	0	0	0	0	0	-1
normalized size	1	1.00	2070.19	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	3.796	35.731	180.000	0.000	0.000	0.000	0.000	0.000

Problem 133	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	370	370	600	0	0	0	0	0	-1
normalized size	1	1.00	1.62	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.052	7.014	180.000	0.000	0.000	0.000	0.000	0.000

Problem 134	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	597	597	1109	0	0	0	0	0	-1
normalized size	1	1.00	1.86	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	3.589	7.348	180.000	0.000	0.000	0.000	0.000	0.000

Problem 135	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F(-1)	F	F(-1)	F
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	682	682	1304	0	0	0	0	0	-1
normalized size	1	1.00	1.91	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	11.896	9.140	180.000	0.000	0.000	0.000	0.000	0.000

Problem 136	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F(-1)	F	F(-1)	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	508	508	867	0	0	0	0	0	-1
normalized size	1	1.00	1.71	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	7.488	9.002	180.000	0.000	0.000	0.000	0.000	0.000

Problem 137	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	384	384	613	0	0	0	0	0	-1
normalized size	1	1.00	1.60	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	4.311	7.691	180.000	0.000	0.000	0.000	0.000	0.000

Problem 138	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	382	382	1073629	0	0	0	0	0	-1
normalized size	1	1.00	2810.55	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	5.739	39.709	180.000	0.000	0.000	0.000	0.000	0.000

Problem 139	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	402	402	1347065	0	0	0	0	0	-1
normalized size	1	1.00	3350.91	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	7.127	41.050	180.000	0.000	0.000	0.000	0.000	0.000

Problem 140	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	F(-1)	F(-2)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	586	586	3134	0	0	0	0	0	-1
normalized size	1	1.00	5.35	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	3.668	9.062	180.000	0.000	0.000	0.000	0.000	0.000

Problem 141	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F(-1)	F(-1)	F
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	697	697	1261	0	0	0	0	0	-1
normalized size	1	1.00	1.81	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	10.416	9.731	180.000	0.000	0.000	0.000	0.000	0.000

Problem 142	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	505	505	780	0	0	0	0	0	-1
normalized size	1	1.00	1.54	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	6.230	8.968	180.000	0.000	0.000	0.000	0.000	0.000

Problem 143	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	535	535	1654245	0	0	0	0	0	-1
normalized size	1	1.00	3092.05	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	8.314	44.384	180.000	0.000	0.000	0.000	0.000	0.000

Problem 144	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	545	545	2018669	0	0	0	0	0	-1
normalized size	1	1.00	3703.98	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	11.066	47.102	180.000	0.000	0.000	0.000	0.000	0.000

Problem 145	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F(-1)	F(-1)	F(-1)
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	590	590	2345519	0	0	0	0	0	-1
normalized size	1	1.00	3975.46	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	14.020	49.131	180.000	0.000	0.000	0.000	0.000	0.000

Problem 146	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-2)	F(-1)	F(-1)	F(-1)	F(-1)
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	946	946	2719441	0	0	0	0	0	-1
normalized size	1	1.00	2874.67	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	6.464	53.638	180.000	0.000	0.000	0.000	0.000	0.000

Problem 147	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	505	505	785	0	0	0	0	0	-1
normalized size	1	1.00	1.55	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	5.953	8.498	180.000	0.000	0.000	0.000	0.000	0.000

Problem 148	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F(-1)	F	F(-1)	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	383	383	582	0	0	0	0	0	-1
normalized size	1	1.00	1.52	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	4.077	7.077	180.000	0.000	0.000	0.000	0.000	0.000

Problem 149	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	290	290	450	0	0	0	0	0	-1
normalized size	1	1.00	1.55	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.555	6.631	180.000	0.000	0.000	0.000	0.000	0.000

Problem 150	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	239	239	362	0	0	0	0	0	-1
normalized size	1	1.00	1.51	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.457	2.301	180.000	0.000	0.000	0.000	0.000	0.000

Problem 151	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	251	251	264	0	0	0	0	0	-1
normalized size	1	1.00	1.05	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	0.969	2.582	180.000	0.000	0.000	0.000	0.000	0.000

Problem 152	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	375	375	388	0	0	0	0	0	-1
normalized size	1	1.00	1.03	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.765	6.279	180.000	0.000	0.000	0.000	0.000	0.000

Problem 153	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	528	528	1653959	0	0	0	0	0	-1
normalized size	1	1.00	3132.50	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	8.188	44.531	180.000	0.000	0.000	0.000	0.000	0.000

Problem 154	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	380	380	1073499	0	0	0	0	0	-1
normalized size	1	1.00	2825.00	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	5.627	39.856	180.000	0.000	0.000	0.000	0.000	0.000

Problem 155	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	299	299	621084	0	0	0	0	0	-1
normalized size	1	1.00	2077.20	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	3.333	35.587	180.000	0.000	0.000	0.000	0.000	0.000

Problem 156	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	251	251	275	0	0	0	0	0	-1
normalized size	1	1.00	1.10	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.001	3.209	180.000	0.000	0.000	0.000	0.000	0.000

Problem 157	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	383	382	484	0	0	0	0	0	-1
normalized size	1	1.00	1.26	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.878	6.720	180.000	0.000	0.000	0.000	0.000	0.000

Problem 158	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	598	598	902	0	0	0	0	0	-1
normalized size	1	1.00	1.51	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	3.435	6.904	180.000	0.000	0.000	0.000	0.000	0.000

Problem 159	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F(-1)	F(-1)	F
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	549	549	2018643	0	0	0	0	0	-1
normalized size	1	1.00	3676.95	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	10.500	47.170	180.000	0.000	0.000	0.000	0.000	0.000

Problem 160	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	C	F(-1)	F(-1)	F(-1)	F	F(-1)	F
verified	N/A	Yes	NO	TBD	TBD	TBD	TBD	TBD	TBD
size	407	407	1347117	0	0	0	0	0	-1
normalized size	1	1.00	3309.87	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	7.163	41.121	180.000	0.000	0.000	0.000	0.000	0.000

Problem 161	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-2)	F(-1)	F	F(-1)	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	373	373	609	0	0	0	0	0	-1
normalized size	1	1.00	1.63	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.922	7.017	180.000	0.000	0.000	0.000	0.000	0.000

Problem 162	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	379	379	403	0	0	0	0	0	-1
normalized size	1	1.00	1.06	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.814	5.591	180.000	0.000	0.000	0.000	0.000	0.000

Problem 163	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F(-1)	F(-1)	F(-1)	F	F(-1)	F(-1)
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	651	650	903	0	0	0	0	0	-1
normalized size	1	1.00	1.39	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	3.431	6.994	180.000	0.000	0.000	0.000	0.000	0.000

Problem 164	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	F	F	F	F	F(-2)	F	F
verified	N/A	Yes	N/A	TBD	TBD	TBD	TBD	TBD	TBD
size	376	376	0	0	0	0	0	0	-1
normalized size	1	1.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	0.900	25.497	4.344	0.000	1.700	0.000	0.000	0.000

Problem 165	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	F	F(-1)	F	F	F	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	560	551	1390	0	0	0	0	0	-1
normalized size	1	0.98	2.48	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.377	6.408	2.622	0.000	0.913	0.000	0.000	0.000

Problem 166	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F	F(-1)	F	F	F	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	363	360	505	0	0	0	0	0	-1
normalized size	1	0.99	1.39	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.152	6.349	2.191	0.000	0.646	0.000	0.000	0.000

Problem 167	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F	F	F	F	F	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	247	247	202	0	0	0	0	0	-1
normalized size	1	1.00	0.82	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	0.528	2.998	1.735	0.000	1.013	0.000	0.000	0.000

Problem 168	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F	F	F	F	F	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	178	178	135	0	0	0	0	0	-1
normalized size	1	1.00	0.76	0.00	0.00	0.00	0.00	0.00	-0.01
time (sec)	N/A	0.184	0.250	1.366	0.000	1.148	0.000	0.000	0.000

Problem 169	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F	F	F	F	F	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	258	258	204	0	0	0	0	0	-1
normalized size	1	1.00	0.79	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	0.482	1.118	4.918	0.000	0.743	0.000	0.000	0.000

Problem 170	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	A	F	F	F	F(-2)	F	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	403	402	563	0	0	0	0	0	-1
normalized size	1	1.00	1.40	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	1.215	6.196	4.555	0.000	1.553	0.000	0.000	0.000

Problem 171	Optimal	Rubi	Mathematica	Maple	Maxima	Fricas	Sympy	Giac	Mupad
grade	A	A	B	F	F(-1)	F	F	F	F
verified	N/A	Yes	Yes	TBD	TBD	TBD	TBD	TBD	TBD
size	702	702	2238	0	0	0	0	0	-1
normalized size	1	1.00	3.19	0.00	0.00	0.00	0.00	0.00	-0.00
time (sec)	N/A	2.938	6.244	5.038	0.000	1.735	0.000	0.000	0.000

2.3 Detailed conclusion table specific for Rubi results

The following table is specific to Rubi. It gives additional statistics for each integral. the column **steps** is the number of steps used by Rubi to obtain the antiderivative. The **rules** column is the number of unique rules used. The **integrand size** column is the leaf size of the integrand. Finally the ratio $\frac{\text{number of rules}}{\text{integrand size}}$ is given. The larger this ratio is, the harder the integral was to solve. In this test, problem number [38] had the largest ratio of [.2000]

Table 2.1: Rubi specific breakdown of results for each integral

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
1	A	5	5	1.00	36	0.139
2	A	3	3	1.00	30	0.100
3	A	3	3	1.00	36	0.083
4	A	5	4	1.00	38	0.105
5	A	4	4	1.00	38	0.105
6	A	5	5	1.00	38	0.132
7	A	6	5	1.00	38	0.132
8	A	7	5	1.00	38	0.132
9	A	6	6	1.00	38	0.158
10	A	4	4	1.00	32	0.125
11	A	4	4	1.00	38	0.105
12	A	5	4	1.00	40	0.100
13	A	5	4	1.00	40	0.100
14	A	5	5	1.00	40	0.125
15	A	6	6	1.00	40	0.150
16	A	7	6	1.00	40	0.150
17	A	5	4	1.00	32	0.125
18	A	5	4	1.00	38	0.105
19	A	6	5	1.00	40	0.125
20	A	6	5	1.00	40	0.125
21	A	6	5	1.00	40	0.125
22	A	6	6	1.00	40	0.150
23	A	7	7	1.00	40	0.175
24	A	8	7	1.00	40	0.175
25	A	7	7	1.00	40	0.175

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
26	A	6	6	1.00	38	0.158
27	A	6	4	1.00	32	0.125
28	A	3	3	1.00	38	0.079
29	A	4	4	1.00	40	0.100
30	A	5	5	1.00	40	0.125
31	A	6	6	1.00	40	0.150
32	A	7	7	1.00	40	0.175
33	A	6	6	1.00	38	0.158
34	A	3	3	1.00	32	0.094
35	A	4	4	1.00	38	0.105
36	A	5	5	1.00	40	0.125
37	A	6	6	1.00	40	0.150
38	A	8	8	1.00	40	0.200
39	A	7	7	1.00	40	0.175
40	A	5	5	1.00	38	0.132
41	A	4	4	1.00	32	0.125
42	A	5	4	1.00	38	0.105
43	A	6	6	1.00	40	0.150
44	A	7	6	1.00	40	0.150
45	A	7	5	1.00	39	0.128
46	A	7	5	1.00	39	0.128
47	A	7	5	1.00	41	0.122
48	A	7	5	1.00	41	0.122
49	A	13	7	1.00	43	0.163
50	A	6	5	1.00	43	0.116
51	A	5	5	1.00	43	0.116

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	number of rules / integrand leaf size
52	A	4	4	1.00	41	0.098
53	A	3	3	1.00	31	0.097
54	A	5	5	0.99	43	0.116
55	A	5	5	1.00	43	0.116
56	A	4	4	1.00	43	0.093
57	A	7	6	1.00	45	0.133
58	A	6	6	1.00	45	0.133
59	A	5	5	0.99	43	0.116
60	A	4	4	1.00	33	0.121
61	A	6	6	0.99	45	0.133
62	A	6	6	1.00	45	0.133
63	A	6	6	1.00	45	0.133
64	A	7	6	1.00	45	0.133
65	A	6	5	0.99	43	0.116
66	A	5	4	1.00	33	0.121
67	A	7	6	1.00	45	0.133
68	A	7	7	1.00	45	0.156
69	A	7	6	1.00	45	0.133
70	A	7	6	1.00	45	0.133
71	A	6	6	1.00	45	0.133
72	A	5	5	1.00	43	0.116
73	A	4	4	1.00	33	0.121
74	A	3	2	0.99	45	0.044
75	A	4	3	1.00	45	0.067
76	A	5	3	1.00	45	0.067
77	A	7	7	1.00	45	0.156

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
78	A	6	6	1.00	45	0.133
79	A	5	5	0.99	43	0.116
80	A	3	3	1.00	33	0.091
81	A	4	3	1.00	45	0.067
82	A	5	3	1.00	45	0.067
83	A	6	3	1.00	45	0.067
84	A	7	6	1.00	45	0.133
85	A	6	6	1.00	45	0.133
86	A	4	4	0.99	43	0.093
87	A	4	4	1.00	33	0.121
88	A	5	3	1.00	45	0.067
89	A	6	3	1.00	45	0.067
90	A	12	8	1.00	47	0.170
91	A	11	8	1.00	47	0.170
92	A	10	7	1.00	45	0.156
93	A	9	6	1.00	35	0.171
94	A	12	7	1.00	47	0.149
95	A	12	7	1.00	47	0.149
96	A	13	8	1.00	47	0.170
97	A	13	8	1.00	47	0.170
98	A	12	8	1.00	47	0.170
99	A	11	7	1.00	45	0.156
100	A	10	6	1.00	35	0.171
101	A	13	7	1.00	47	0.149
102	A	13	8	1.00	47	0.170
103	A	13	7	1.00	47	0.149

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
104	A	13	8	1.00	47	0.170
105	A	12	7	0.99	45	0.156
106	A	11	6	1.00	35	0.171
107	A	14	7	1.00	47	0.149
108	A	14	8	1.00	47	0.170
109	A	14	8	1.00	47	0.170
110	A	11	7	1.00	47	0.149
111	A	10	7	1.00	47	0.149
112	A	9	6	1.00	45	0.133
113	A	8	5	1.00	35	0.143
114	A	11	6	1.00	47	0.128
115	A	12	7	1.00	47	0.149
116	A	11	8	1.00	47	0.170
117	A	10	7	1.00	47	0.149
118	A	9	6	1.00	45	0.133
119	A	8	5	1.00	35	0.143
120	A	12	7	1.00	47	0.149
121	A	13	7	1.00	47	0.149
122	A	11	7	1.00	47	0.149
123	A	10	7	1.00	47	0.149
124	A	9	6	0.99	45	0.133
125	A	9	6	1.00	35	0.171
126	A	13	7	1.00	47	0.149
127	A	14	7	1.00	47	0.149
128	A	16	8	1.00	49	0.163
129	A	15	8	1.00	49	0.163

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
130	A	14	8	1.01	49	0.163
131	A	13	8	1.00	49	0.163
132	A	13	8	1.00	49	0.163
133	A	9	6	1.00	49	0.122
134	A	10	6	1.00	49	0.122
135	A	16	8	1.00	49	0.163
136	A	15	8	1.00	49	0.163
137	A	14	8	1.00	49	0.163
138	A	14	9	1.00	49	0.184
139	A	14	8	1.00	49	0.163
140	A	10	6	1.00	49	0.122
141	A	16	8	1.00	49	0.163
142	A	15	8	1.00	49	0.163
143	A	15	9	1.00	49	0.184
144	A	15	9	1.00	49	0.184
145	A	15	8	1.00	49	0.163
146	A	11	6	1.00	49	0.122
147	A	15	8	1.00	49	0.163
148	A	14	8	1.00	49	0.163
149	A	13	8	1.00	49	0.163
150	A	12	7	1.00	49	0.143
151	A	8	5	1.00	49	0.102
152	A	9	5	1.00	49	0.102
153	A	15	9	1.00	49	0.184
154	A	14	9	1.00	49	0.184
155	A	13	8	1.00	49	0.163

Continued on next page

Table 2.1 – continued from previous page

#	grade	number of steps used	number of unique rules	normalized antiderivative leaf size	integrand leaf size	$\frac{\text{number of rules}}{\text{integrand leaf size}}$
156	A	8	5	1.00	49	0.102
157	A	9	5	1.00	49	0.102
158	A	10	5	1.00	49	0.102
159	A	15	9	1.00	49	0.184
160	A	14	8	1.00	49	0.163
161	A	9	6	1.00	49	0.122
162	A	9	5	1.00	49	0.102
163	A	10	5	1.00	49	0.102
164	A	9	6	1.00	45	0.133
165	A	9	6	0.98	45	0.133
166	A	8	6	0.99	45	0.133
167	A	7	5	1.00	43	0.116
168	A	6	4	1.00	33	0.121
169	A	8	5	1.00	45	0.111
170	A	9	6	1.00	45	0.133
171	A	10	6	1.00	45	0.133

Chapter 3

Listing of integrals

3.1 $\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+$

Optimal. Leaf size=87

$$\frac{(aC+bB)\tan^2(c+dx)}{2d}+\frac{(aB-bC)\tan(c+dx)}{d}+\frac{(aC+bB)\log(\cos(c+dx))}{d}-x(aB-bC)+\frac{bC\tan^3(c+dx)}{3d}$$

```
[Out] -(B*a-C*b)*x+(B*b+C*a)*ln(cos(d*x+c))/d+(B*a-C*b)*tan(d*x+c)/d+1/2*(B*b+C*a
)*tan(d*x+c)^2/d+1/3*b*C*tan(d*x+c)^3/d
```

Rubi [A] time = 0.13, antiderivative size = 87, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, integrand size = 36, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.139, Rules used = {3632, 3592, 3528, 3525, 3475}

$$\frac{(aC+bB)\tan^2(c+dx)}{2d}+\frac{(aB-bC)\tan(c+dx)}{d}+\frac{(aC+bB)\log(\cos(c+dx))}{d}-x(aB-bC)+\frac{bC\tan^3(c+dx)}{3d}$$

Antiderivative was successfully verified.

```
[In] Int[Tan[c + d*x]*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x
]
```

```
[Out] -((a*B - b*C)*x) + ((b*B + a*C)*Log[Cos[c + d*x]])/d + ((a*B - b*C)*Tan[c +
 d*x])/d + ((b*B + a*C)*Tan[c + d*x]^2)/(2*d) + (b*C*Tan[c + d*x]^3)/(3*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3592

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(
B*d*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*
x])^m*Simp[A*c - B*d + (B*c + A*d)*Tan[e + f*x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, m}, x] && NeQ[b*c - a*d, 0] &&  !LeQ[m, -1]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \tan(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int \tan^2(c+dx)(a+b\tan(c+dx))(B+C$$

$$= \frac{bC\tan^3(c+dx)}{3d} + \int \tan^2(c+dx)(aB-$$

$$= \frac{(bB+aC)\tan^2(c+dx)}{2d} + \frac{bC\tan^3(c+d}{3d}$$

$$= -(aB-bC)x + \frac{(aB-bC)\tan(c+dx)}{d} +$$

$$= -(aB-bC)x + \frac{(bB+aC)\log(\cos(c+d}{d}$$

Mathematica [A] time = 0.60, size = 86, normalized size = 0.99

$$\frac{(6bC-6aB)\tan^{-1}(\tan(c+dx))+3(aC+bB)\tan^2(c+dx)+6(aB-bC)\tan(c+dx)+6(aC+bB)\log(\cos(c+}{6d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Tan[c + d*x]*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x
]^2),x]
```

```
[Out] ((-6*a*B + 6*b*C)*ArcTan[Tan[c + d*x]] + 6*(b*B + a*C)*Log[Cos[c + d*x]] +
6*(a*B - b*C)*Tan[c + d*x] + 3*(b*B + a*C)*Tan[c + d*x]^2 + 2*b*C*Tan[c + d
*x]^3)/(6*d)
```

fricas [A] time = 0.60, size = 85, normalized size = 0.98

$$\frac{2\,Cb\tan\left(dx+c\right)^3-6\,(Ba-Cb)dx+3\,(Ca+Bb)\tan\left(dx+c\right)^2+3\,(Ca+Bb)\log\left(\frac{1}{\tan(dx+c)^2+1}\right)+6\,(Ba-Cb)\tan}{6\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="fricas")
```

```
[Out] 1/6*(2*C*b*tan(d*x + c)^3 - 6*(B*a - C*b)*d*x + 3*(C*a + B*b)*tan(d*x + c)^
2 + 3*(C*a + B*b)*log(1/(tan(d*x + c)^2 + 1)) + 6*(B*a - C*b)*tan(d*x + c))
/d
```

giac [B] time = 3.96, size = 1017, normalized size = 11.69

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="giac")
```

```
[Out] -1/6*(6*B*a*d*x*tan(d*x)^3*tan(c)^3 - 6*C*b*d*x*tan(d*x)^3*tan(c)^3 - 3*C*a
*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + t
an(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 - 3*
B*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2
+ tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 -
 18*B*a*d*x*tan(d*x)^2*tan(c)^2 + 18*C*b*d*x*tan(d*x)^2*tan(c)^2 - 3*C*a*ta
n(d*x)^3*tan(c)^3 - 3*B*b*tan(d*x)^3*tan(c)^3 + 9*C*a*log(4*(tan(d*x)^4*tan
(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)
*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 + 9*B*b*log(4*(tan(d*x)^4*
```

```
tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d
*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 + 6*B*a*tan(d*x)^3*tan(
c)^2 - 6*C*b*tan(d*x)^3*tan(c)^2 + 6*B*a*tan(d*x)^2*tan(c)^3 - 6*C*b*tan(d*
x)^2*tan(c)^3 + 18*B*a*d*x*tan(d*x)*tan(c) - 18*C*b*d*x*tan(d*x)*tan(c) - 3
*C*a*tan(d*x)^3*tan(c) - 3*B*b*tan(d*x)^3*tan(c) + 3*C*a*tan(d*x)^2*tan(c)^
2 + 3*B*b*tan(d*x)^2*tan(c)^2 - 3*C*a*tan(d*x)*tan(c)^3 - 3*B*b*tan(d*x)*ta
n(c)^3 + 2*C*b*tan(d*x)^3 - 9*C*a*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3
*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)
^2 + 1))*tan(d*x)*tan(c) - 9*B*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*
tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^
2 + 1))*tan(d*x)*tan(c) - 12*B*a*tan(d*x)^2*tan(c) + 18*C*b*tan(d*x)^2*tan(
c) - 12*B*a*tan(d*x)*tan(c)^2 + 18*C*b*tan(d*x)*tan(c)^2 + 2*C*b*tan(c)^3 -
 6*B*a*d*x + 6*C*b*d*x + 3*C*a*tan(d*x)^2 + 3*B*b*tan(d*x)^2 - 3*C*a*tan(d*
x)*tan(c) - 3*B*b*tan(d*x)*tan(c) + 3*C*a*tan(c)^2 + 3*B*b*tan(c)^2 + 3*C*a
*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + t
an(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1)) + 3*B*b*log(4*(tan(d*x)^
4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan
(d*x)*tan(c) + 1)/(tan(c)^2 + 1)) + 6*B*a*tan(d*x) - 6*C*b*tan(d*x) + 6*B*a
*tan(c) - 6*C*b*tan(c) + 3*C*a + 3*B*b)/(d*tan(d*x)^3*tan(c)^3 - 3*d*tan(d*
x)^2*tan(c)^2 + 3*d*tan(d*x)*tan(c) - d)
```

maple [A] time = 0.03, size = 135, normalized size = 1.55

$$\frac{bC\left(\tan^3(dx+c)\right)}{3d}+\frac{bB\left(\tan^2(dx+c)\right)}{2d}+\frac{C\left(\tan^2(dx+c)\right)a}{2d}+\frac{aB\tan(dx+c)}{d}-\frac{bC\tan(dx+c)}{d}-\frac{\ln\left(1+\tan^2(dx+\right.}{2d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] 1/3*b*C*tan(d*x+c)^3/d+1/2*b*B*tan(d*x+c)^2/d+1/2/d*C*tan(d*x+c)^2*a+1/d*a*
B*tan(d*x+c)-b*C*tan(d*x+c)/d-1/2/d*ln(1+tan(d*x+c)^2)*B*b-1/2/d*ln(1+tan(d
*x+c)^2)*a*C-1/d*B*arctan(tan(d*x+c))*a+1/d*C*arctan(tan(d*x+c))*b
```

maxima [A] time = 0.54, size = 86, normalized size = 0.99

$$\frac{2\,Cb\tan(dx+c)^3+3\,(Ca+Bb)\tan(dx+c)^2-6\,(Ba-Cb)(dx+c)-3\,(Ca+Bb)\log\left(\tan(dx+c)^2+1\right)+6\,(Ba}{6\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="maxima")
```

```
[Out] 1/6*(2*C*b*tan(d*x + c)^3 + 3*(C*a + B*b)*tan(d*x + c)^2 - 6*(B*a - C*b)*(d
*x + c) - 3*(C*a + B*b)*log(tan(d*x + c)^2 + 1) + 6*(B*a - C*b)*tan(d*x + c
))/d
```

mupad [B] time = 8.83, size = 84, normalized size = 0.97

$$\frac{\tan(c+d\,x)\,\left(B\,a-C\,b\right)-\ln\left(\tan(c+d\,x)^2+1\right)\,\left(\frac{B\,b}{2}+\frac{C\,a}{2}\right)+\tan(c+d\,x)^2\,\left(\frac{B\,b}{2}+\frac{C\,a}{2}\right)-d\,x\,\left(B\,a-C\,b\right)+\frac{C\,b\,\mathrm{ta}}{}}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x)),x
)
```

```
[Out] (tan(c + d*x)*(B*a - C*b) - log(tan(c + d*x)^2 + 1)*((B*b)/2 + (C*a)/2) + t
an(c + d*x)^2*((B*b)/2 + (C*a)/2) - d*x*(B*a - C*b) + (C*b*tan(c + d*x)^3)/
3)/d
```

sympy [A] time = 0.37, size = 139, normalized size = 1.60

$$\begin{cases} -Bax+\frac{Ba\tan(c+dx)}{d}-\frac{Bb\log\left(\tan^2(c+dx)+1\right)}{2d}+\frac{Bb\tan^2(c+dx)}{2d}-\frac{Ca\log\left(\tan^2(c+dx)+1\right)}{2d}+\frac{Ca\tan^2(c+dx)}{2d}+Cbx+\frac{Cb\tan^3(c+dx)}{3d} \\ x\left(a+b\tan(c)\right)\left(B\tan(c)+C\tan^2(c)\right)\tan(c) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((-B*a*x + B*a*tan(c + d*x)/d - B*b*log(tan(c + d*x)**2 + 1)/(2*d)
 + B*b*tan(c + d*x)**2/(2*d) - C*a*log(tan(c + d*x)**2 + 1)/(2*d) + C*a*tan
(c + d*x)**2/(2*d) + C*b*x + C*b*tan(c + d*x)**3/(3*d) - C*b*tan(c + d*x)/d
, Ne(d, 0)), (x*(a + b*tan(c))*(B*tan(c) + C*tan(c)**2)*tan(c), True))
```

3.2 $\int (a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$

Optimal. Leaf size=66

$$-\frac{(aB-bC)\log(\cos(c+dx))}{d}-x(aC+bB)+\frac{C(a+b\tan(c+dx))^2}{2bd}+\frac{bB\tan(c+dx)}{d}$$

```
[Out] -(B*b+C*a)*x-(B*a-C*b)*ln(cos(d*x+c))/d+b*B*tan(d*x+c)/d+1/2*C*(a+b*tan(d*x
+c))^2/b/d
```

Rubi [A] time = 0.05, antiderivative size = 66, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, integrand size = 30, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.100, Rules used = {3630, 3525, 3475}

$$-\frac{(aB-bC)\log(\cos(c+dx))}{d}-x(aC+bB)+\frac{C(a+b\tan(c+dx))^2}{2bd}+\frac{bB\tan(c+dx)}{d}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] -((b*B + a*C)*x) - ((a*B - b*C)*Log[Cos[c + d*x]])/d + (b*B*Tan[c + d*x])/d
 + (C*(a + b*Tan[c + d*x])^2)/(2*b*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int (a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \frac{C(a+b\tan(c+dx))^2}{2bd} + \int (a+b\tan(c+dx))(-C$$
$$= -(bB+aC)x + \frac{bB\tan(c+dx)}{d} + \frac{C(a+b\tan(c+a}{2bd}$$
$$= -(bB+aC)x - \frac{(aB-bC)\log(\cos(c+dx))}{d} + \frac{bB\,ta}{}$$

Mathematica [A] time = 0.32, size = 67, normalized size = 1.02

$$\frac{-2(aC+bB)\tan^{-1}(\tan(c+dx)) + 2(aC+bB)\tan(c+dx) + 2(bC-aB)\log(\cos(c+dx)) + bC\tan^2(c+dx)}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] (-2*(b*B + a*C)*ArcTan[Tan[c + d*x]] + 2*(-(a*B) + b*C)*Log[Cos[c + d*x]] +
 2*(b*B + a*C)*Tan[c + d*x] + b*C*Tan[c + d*x]^2)/(2*d)
```

fricas [A] time = 0.61, size = 66, normalized size = 1.00

$$\frac{Cb\tan(dx+c)^2 - 2(Ca+Bb)dx - (Ba-Cb)\log\left(\frac{1}{\tan(dx+c)^2+1}\right) + 2(Ca+Bb)\tan(dx+c)}{2d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="fric
as")
```

```
[Out] 1/2*(C*b*tan(d*x + c)^2 - 2*(C*a + B*b)*d*x - (B*a - C*b)*log(1/(tan(d*x +
c)^2 + 1)) + 2*(C*a + B*b)*tan(d*x + c))/d
```

giac [B] time = 2.92, size = 616, normalized size = 9.33

$$-\frac{2Cadx\tan(dx)^2\tan(c)^2 + 2Bbdx\tan(dx)^2\tan(c)^2 + Ba\log\left(\frac{4\left(\tan(dx)^4\tan(c)^2 - 2\tan(dx)^3\tan(c) + \tan(dx)^2\tan(c)^2 + \tan(dx\right.}{\tan(c)^2+1}\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="giac
")
```

```
[Out] -1/2*(2*C*a*d*x*tan(d*x)^2*tan(c)^2 + 2*B*b*d*x*tan(d*x)^2*tan(c)^2 + B*a*l
og(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan
(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 - C*b*
log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + ta
n(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 - 4*C
*a*d*x*tan(d*x)*tan(c) - 4*B*b*d*x*tan(d*x)*tan(c) - C*b*tan(d*x)^2*tan(c)^
2 - 2*B*a*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan
(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)*tan(c)
 + 2*C*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(
c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)*tan(c)
+ 2*C*a*tan(d*x)^2*tan(c) + 2*B*b*tan(d*x)^2*tan(c) + 2*C*a*tan(d*x)*tan(c)
^2 + 2*B*b*tan(d*x)*tan(c)^2 + 2*C*a*d*x + 2*B*b*d*x - C*b*tan(d*x)^2 - C*b
*tan(c)^2 + B*a*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)
^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1)) - C*b*log
(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d
*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1)) - 2*C*a*tan(d*x) - 2*B*b*tan
(d*x) - 2*C*a*tan(c) - 2*B*b*tan(c) - C*b)/(d*tan(d*x)^2*tan(c)^2 - 2*d*tan
(d*x)*tan(c) + d)
```

maple [A] time = 0.02, size = 105, normalized size = 1.59

$$\frac{Cb\left(\tan^{2}\left(dx+c\right)\right)}{2d}+\frac{bB\tan\left(dx+c\right)}{d}+\frac{C\tan\left(dx+c\right)a}{d}+\frac{a\ln\left(1+\tan^{2}\left(dx+c\right)\right)B}{2d}-\frac{\ln\left(1+\tan^{2}\left(dx+c\right)\right)Cb}{2d}-\frac{B\,\mathrm{ar}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] 1/2/d*C*b*tan(d*x+c)^2+b*B*tan(d*x+c)/d+1/d*C*tan(d*x+c)*a+1/2/d*ln(1+tan(d
*x+c)^2)*a*B-1/2/d*ln(1+tan(d*x+c)^2)*C*b-1/d*B*arctan(tan(d*x+c))*b-1/d*C*
arctan(tan(d*x+c))*a
```

maxima [A] time = 0.65, size = 66, normalized size = 1.00

$$\frac{Cb\tan\left(dx+c\right)^{2}-2\left(Ca+Bb\right)\left(dx+c\right)+\left(Ba-Cb\right)\log\left(\tan\left(dx+c\right)^{2}+1\right)+2\left(Ca+Bb\right)\tan\left(dx+c\right)}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="maxi
ma")
```

```
[Out] 1/2*(C*b*tan(d*x + c)^2 - 2*(C*a + B*b)*(d*x + c) + (B*a - C*b)*log(tan(d*x
 + c)^2 + 1) + 2*(C*a + B*b)*tan(d*x + c))/d
```

mupad [B] time = 8.84, size = 63, normalized size = 0.95

$$\frac{\tan(c+d\,x)\,(B\,b+C\,a)+\ln\left(\tan(c+d\,x)^2+1\right)\left(\frac{B\,a}{2}-\frac{C\,b}{2}\right)-d\,x\,(B\,b+C\,a)+\frac{C\,b\tan(c+d\,x)^2}{2}}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x)),x)
```

```
[Out] (tan(c + d*x)*(B*b + C*a) + log(tan(c + d*x)^2 + 1)*((B*a)/2 - (C*b)/2) - d
*x*(B*b + C*a) + (C*b*tan(c + d*x)^2)/2)/d
```

sympy [A] time = 0.24, size = 105, normalized size = 1.59

$$\begin{cases} \frac{Ba\log\left(\tan^2(c+dx)+1\right)}{2d}-Bbx+\frac{Bb\tan(c+dx)}{d}-Cax+\frac{Ca\tan(c+dx)}{d}-\frac{Cb\log\left(\tan^2(c+dx)+1\right)}{2d}+\frac{Cb\tan^2(c+dx)}{2d} & \text{for } d\neq 0 \\ x\,(a+b\tan(c))\left(B\tan(c)+C\tan^2(c)\right) & \text{otherwise} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a*log(tan(c + d*x)**2 + 1)/(2*d) - B*b*x + B*b*tan(c + d*x)/d
- C*a*x + C*a*tan(c + d*x)/d - C*b*log(tan(c + d*x)**2 + 1)/(2*d) + C*b*tan
(c + d*x)**2/(2*d), Ne(d, 0)), (x*(a + b*tan(c))*(B*tan(c) + C*tan(c)**2),
True))
```

3.3 $\int \cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+\right.$

Optimal. Leaf size=42

$$-\frac{(aC+bB)\log(\cos(c+dx))}{d}+x(aB-bC)+\frac{bC\tan(c+dx)}{d}$$

```
[Out] (B*a-C*b)*x-(B*b+C*a)*ln(cos(d*x+c))/d+b*C*tan(d*x+c)/d
```

Rubi [A] time = 0.06, antiderivative size = 42, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, integrand size = 36, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.083, Rules used = {3632, 3525, 3475}

$$-\frac{(aC+bB)\log(\cos(c+dx))}{d}+x(aB-bC)+\frac{bC\tan(c+dx)}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x
]
```

```
[Out] (a*B - b*C)*x - ((b*B + a*C)*Log[Cos[c + d*x]])/d + (b*C*Tan[c + d*x])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int (a+b\tan(c+dx))(B+C\tan(c+dx$$
$$= (aB-bC)x + \frac{bC\tan(c+dx)}{d} + (bB+a$$
$$= (aB-bC)x - \frac{(bB+aC)\log(\cos(c+dx}{d}$$

Mathematica [A] time = 0.06, size = 59, normalized size = 1.40

$$aBx - \frac{aC\log(\cos(c+dx))}{d} - \frac{bB\log(\cos(c+dx))}{d} - \frac{bC\tan^{-1}(\tan(c+dx))}{d} + \frac{bC\tan(c+dx)}{d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x
]^2),x]
```

```
[Out] a*B*x - (b*C*ArcTan[Tan[c + d*x]])/d - (b*B*Log[Cos[c + d*x]])/d - (a*C*Log
[Cos[c + d*x]])/d + (b*C*Tan[c + d*x])/d
```

fricas [A] time = 1.06, size = 50, normalized size = 1.19

$$\frac{2\,(Ba-Cb)dx + 2\,Cb\tan\left(dx+c\right) - (Ca+Bb)\log\left(\frac{1}{\tan\left(dx+c\right)^2+1}\right)}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="fricas")
```

```
[Out] 1/2*(2*(B*a - C*b)*d*x + 2*C*b*tan(d*x + c) - (C*a + B*b)*log(1/(tan(d*x +
c)^2 + 1)))/d
```

giac [A] time = 2.27, size = 50, normalized size = 1.19

$$\frac{2\,Cb\tan\left(dx+c\right) + 2\,(Ba-Cb)(dx+c) + (Ca+Bb)\log\left(\tan\left(dx+c\right)^2+1\right)}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="giac")
```

```
[Out] 1/2*(2*C*b*tan(d*x + c) + 2*(B*a - C*b)*(d*x + c) + (C*a + B*b)*log(tan(d*x
 + c)^2 + 1))/d
```

maple [A] time = 0.39, size = 66, normalized size = 1.57

$$aBx - bCx - \frac{bB\ln(\cos(dx+c))}{d} + \frac{Bac}{d} + \frac{bC\tan(dx+c)}{d} - \frac{aC\ln(\cos(dx+c))}{d} - \frac{Cbc}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] a*B*x-b*C*x-b*B*ln(cos(d*x+c))/d+1/d*B*a*c+b*C*tan(d*x+c)/d-1/d*a*C*ln(cos(
d*x+c))-1/d*C*b*c
```

maxima [A] time = 0.60, size = 50, normalized size = 1.19

$$\frac{2\,Cb\tan(dx+c) + 2\,(Ba - Cb)(dx+c) + (Ca + Bb)\log\left(\tan(dx+c)^2 + 1\right)}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algo
rithm="maxima")
```

```
[Out] 1/2*(2*C*b*tan(d*x + c) + 2*(B*a - C*b)*(d*x + c) + (C*a + B*b)*log(tan(d*x
 + c)^2 + 1))/d
```

mupad [B] time = 8.79, size = 58, normalized size = 1.38

$$B\,a\,x - C\,b\,x + \frac{C\,b\tan(c+d\,x)}{d} + \frac{B\,b\,\ln\left(\tan(c+d\,x)^2+1\right)}{2\,d} + \frac{C\,a\,\ln\left(\tan(c+d\,x)^2+1\right)}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x)),x
)
```

```
[Out] B*a*x - C*b*x + (C*b*tan(c + d*x))/d + (B*b*log(tan(c + d*x)^2 + 1))/(2*d)
+ (C*a*log(tan(c + d*x)^2 + 1))/(2*d)
```

sympy [A] time = 0.65, size = 82, normalized size = 1.95

$$\begin{cases} Bax + \frac{Bb\log\left(\tan^2(c+dx)+1\right)}{2d} + \frac{Ca\log\left(\tan^2(c+dx)+1\right)}{2d} - Cbx + \frac{Cb\tan(c+dx)}{d} & \text{for } d \neq 0 \\ x\,(a + b\tan(c))\left(B\tan(c) + C\tan^2(c)\right)\cot(c) & \text{otherwise} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a*x + B*b*log(tan(c + d*x)**2 + 1)/(2*d) + C*a*log(tan(c + d*x
)**2 + 1)/(2*d) - C*b*x + C*b*tan(c + d*x)/d, Ne(d, 0)), (x*(a + b*tan(c))*
(B*tan(c) + C*tan(c)**2)*cot(c), True))
```

3.4 $\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+\right.$

Optimal. Leaf size=37

$$x(aC+bB)+\frac{aB\log(\sin(c+dx))}{d}-\frac{bC\log(\cos(c+dx))}{d}$$

```
[Out] (B*b+C*a)*x-b*C*ln(cos(d*x+c))/d+a*B*ln(sin(d*x+c))/d
```

Rubi [A] time = 0.11, antiderivative size = 37, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3632, 3589, 3475, 3531}

$$x(aC+bB)+\frac{aB\log(\sin(c+dx))}{d}-\frac{bC\log(\cos(c+dx))}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^2*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] (b*B + a*C)*x - (b*C*Log[Cos[c + d*x]])/d + (a*B*Log[Sin[c + d*x]])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3589

```
Int[(((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_
.)*(x_)]))/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Dist[(B*d)
/b, Int[Tan[e + f*x], x], x] + Dist[1/b, Int[Simp[A*b*c + (A*b*d + B*(b*c -
 a*d))*Tan[e + f*x], x]/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d,
e, f, A, B}, x] && NeQ[b*c - a*d, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot^2(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \int \cot(c+dx)(a+b\tan(c+dx))(B+$$
$$= (bC)\int \tan(c+dx)\,dx + \int \cot(c+dx$$
$$= (bB+aC)x - \frac{bC\log(\cos(c+dx))}{d} + ($$
$$= (bB+aC)x - \frac{bC\log(\cos(c+dx))}{d} + $$

Mathematica [A] time = 0.07, size = 44, normalized size = 1.19

$$\frac{aB(\log(\tan(c+dx))+\log(\cos(c+dx)))}{d}+aCx+bBx-\frac{bC\log(\cos(c+dx))}{d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^2*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] b*B*x + a*C*x - (b*C*Log[Cos[c + d*x]])/d + (a*B*(Log[Cos[c + d*x]] + Log[T
an[c + d*x]]))/d
```

fricas [A] time = 0.46, size = 59, normalized size = 1.59

$$\frac{2\,(Ca+Bb)dx+Ba\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)-Cb\log\left(\frac{1}{\tan(dx+c)^2+1}\right)}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] 1/2*(2*(C*a + B*b)*d*x + B*a*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1)) - C*b
*log(1/(tan(d*x + c)^2 + 1)))/d
```

giac [A] time = 3.67, size = 53, normalized size = 1.43

$$\frac{2\,Ba\log\left(|\tan\left(dx+c\right)|\right)+2\,(Ca+Bb)(dx+c)-(Ba-Cb)\log\left(\tan\left(dx+c\right)^2+1\right)}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] 1/2*(2*B*a*log(abs(tan(d*x + c))) + 2*(C*a + B*b)*(d*x + c) - (B*a - C*b)*l
og(tan(d*x + c)^2 + 1))/d
```

maple [A] time = 0.57, size = 51, normalized size = 1.38

$$Bxb+aCx+\frac{aB\ln\left(\sin\left(dx+c\right)\right)}{d}+\frac{Bbc}{d}-\frac{bC\ln\left(\cos\left(dx+c\right)\right)}{d}+\frac{Cac}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^2*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] B*x*b+a*C*x+a*B*ln(sin(d*x+c))/d+1/d*B*b*c-b*C*ln(cos(d*x+c))/d+1/d*C*a*c
```

maxima [A] time = 0.59, size = 52, normalized size = 1.41

$$\frac{2\,Ba\log\left(\tan\left(dx+c\right)\right)+2\,(Ca+Bb)(dx+c)-(Ba-Cb)\log\left(\tan\left(dx+c\right)^2+1\right)}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/2*(2*B*a*log(tan(d*x + c)) + 2*(C*a + B*b)*(d*x + c) - (B*a - C*b)*log(ta
n(d*x + c)^2 + 1))/d
```

mupad [B] time = 8.96, size = 69, normalized size = 1.86

$$\frac{B\,a\,\ln\left(\tan\left(c+d\,x\right)\right)}{d}-\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(a+b\,1\mathrm{i}\right)}{2\,d}+\frac{\ln\left(\tan\left(c+d\,x\right)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,\left(b+a\,1\mathrm{i}\right)\,1\mathrm{i}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)*1i)/(2*d) - (log(tan(c + d*x)
 - 1i)*(B + C*1i)*(a + b*1i))/(2*d) + (B*a*log(tan(c + d*x)))/d
```

sympy [**A**] time = 0.98, size = 85, normalized size = 2.30

$$\begin{cases} -\frac{Ba\log\left(\tan^2(c+dx)+1\right)}{2d} + \frac{Ba\log(\tan(c+dx))}{d} + Bbx + Cax + \frac{Cb\log\left(\tan^2(c+dx)+1\right)}{2d} & \text{for } d \neq 0 \\ x\,(a + b\tan(c))\left(B\tan(c) + C\tan^2(c)\right)\cot^2(c) & \text{otherwise} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**2*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((-B*a*log(tan(c + d*x)**2 + 1)/(2*d) + B*a*log(tan(c + d*x))/d +
B*b*x + C*a*x + C*b*log(tan(c + d*x)**2 + 1)/(2*d), Ne(d, 0)), (x*(a + b*ta
n(c))*(B*tan(c) + C*tan(c)**2)*cot(c)**2, True))
```

3.5 $\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+\right.$

Optimal. Leaf size=43

$$\frac{(aC+bB)\log(\sin(c+dx))}{d}-(x(aB-bC))-\frac{aB\cot(c+dx)}{d}$$

```
[Out] -(B*a-C*b)*x-a*B*cot(d*x+c)/d+(B*b+C*a)*ln(sin(d*x+c))/d
```

Rubi [A] time = 0.12, antiderivative size = 43, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3632, 3591, 3531, 3475}

$$\frac{(aC+bB)\log(\sin(c+dx))}{d}+x(-(aB-bC))-\frac{aB\cot(c+dx)}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^3*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] -((a*B - b*C)*x) - (a*B*Cot[c + d*x])/d + ((b*B + a*C)*Log[Sin[c + d*x]])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3591

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((
b*c - a*d)*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^
2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[a*A*c +
 b*B*c + A*b*d - a*B*d - (A*b*c - a*B*c - a*A*d - b*B*d)*Tan[e + f*x], x],
x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c - a*d, 0] && LtQ[m,
 -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot^3(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \int \cot^2(c+dx)(a+b\tan(c+dx))(B+$$
$$= -\frac{aB\cot(c+dx)}{d} + \int \cot(c+dx)(bB-$$
$$= -(aB-bC)x - \frac{aB\cot(c+dx)}{d} + (bB+$$
$$= -(aB-bC)x - \frac{aB\cot(c+dx)}{d} + \frac{(bB+}{}$$

Mathematica [C] time = 0.16, size = 78, normalized size = 1.81

$$-\frac{aB\cot(c+dx)\,{}_2F_1\left(-\frac{1}{2},1;\frac{1}{2};-\tan^2(c+dx)\right)}{d}+\frac{aC(\log(\tan(c+dx))+\log(\cos(c+dx)))}{d}+\frac{bB(\log(\tan(c+dx))}{d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^3*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] b*C*x - (a*B*Cot[c + d*x]*Hypergeometric2F1[-1/2, 1, 1/2, -Tan[c + d*x]^2])
/d + (b*B*(Log[Cos[c + d*x]] + Log[Tan[c + d*x]]))/d + (a*C*(Log[Cos[c + d*
x]] + Log[Tan[c + d*x]]))/d
```

fricas [A] time = 1.55, size = 73, normalized size = 1.70

$$-\frac{2\,(Ba-Cb)dx\tan\left(dx+c\right)-(Ca+Bb)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)+2\,Ba}{2\,d\tan\left(dx+c\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] -1/2*(2*(B*a - C*b)*d*x*tan(d*x + c) - (C*a + B*b)*log(tan(d*x + c)^2/(tan(
d*x + c)^2 + 1))*tan(d*x + c) + 2*B*a)/(d*tan(d*x + c))
```

giac [B] time = 4.31, size = 119, normalized size = 2.77

$$\frac{Ba\tan\left(\frac{1}{2}dx+\frac{1}{2}c\right)-2\,(Ba-Cb)(dx+c)-2\,(Ca+Bb)\log\left(\tan\left(\frac{1}{2}dx+\frac{1}{2}c\right)^{2}+1\right)+2\,(Ca+Bb)\log\left(\left|\tan\left(\frac{1}{2}dx\right.\right.\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] 1/2*(B*a*tan(1/2*d*x + 1/2*c) - 2*(B*a - C*b)*(d*x + c) - 2*(C*a + B*b)*log
(tan(1/2*d*x + 1/2*c)^2 + 1) + 2*(C*a + B*b)*log(abs(tan(1/2*d*x + 1/2*c)))
 - (2*C*a*tan(1/2*d*x + 1/2*c) + 2*B*b*tan(1/2*d*x + 1/2*c) + B*a)/tan(1/2*
d*x + 1/2*c))/d
```

maple [A] time = 0.44, size = 65, normalized size = 1.51

$$-aBx+bCx-\frac{aB\cot(dx+c)}{d}+\frac{Bb\ln(\sin(dx+c))}{d}-\frac{Bac}{d}+\frac{aC\ln(\sin(dx+c))}{d}+\frac{Cbc}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^3*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -a*B*x+b*C*x-a*B*cot(d*x+c)/d+1/d*B*b*ln(sin(d*x+c))-1/d*B*a*c+1/d*a*C*ln(s
in(d*x+c))+1/d*C*b*c
```

maxima [A] time = 0.46, size = 68, normalized size = 1.58

$$-\frac{2\,(Ba-Cb)(dx+c)+(Ca+Bb)\log\left(\tan(dx+c)^{2}+1\right)-2\,(Ca+Bb)\log(\tan(dx+c))+\frac{2\,Ba}{\tan(dx+c)}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(B*a - C*b)*(d*x + c) + (C*a + B*b)*log(tan(d*x + c)^2 + 1) - 2*(C*
a + B*b)*log(tan(d*x + c)) + 2*B*a/tan(d*x + c))/d
```

mupad [B] time = 8.87, size = 87, normalized size = 2.02

$$\frac{\ln(\tan(c+d\,x))\,(B\,b+C\,a)}{d}-\frac{\ln(\tan(c+d\,x)+1\mathrm{i})\,(B-C\,1\mathrm{i})\,(b+a\,1\mathrm{i})}{2\,d}-\frac{B\,a\cot(c+d\,x)}{d}+\frac{\ln(\tan(c+d\,x)-\mathrm{i})\,(}{2}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x))*(B*b + C*a))/d + (log(tan(c + d*x) - 1i)*(B + C*1i)*(a +
 b*1i)*1i)/(2*d) - (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b))/(2*d) - (
B*a*cot(c + d*x))/d
```

sympy [A] time = 1.66, size = 116, normalized size = 2.70

$$\begin{cases} \text{NaN} & \text{for } c = \\ x\left(a+b\tan\left(c\right)\right)\left(B\tan\left(c\right)+C\tan^{2}\left(c\right)\right)\cot^{3}\left(c\right) & \text{for } d = \\ \text{NaN} & \text{for } c = \\ -Bax-\frac{Ba}{d\tan\left(c+dx\right)}-\frac{Bb\log\left(\tan^{2}\left(c+dx\right)+1\right)}{2d}+\frac{Bb\log\left(\tan\left(c+dx\right)\right)}{d}-\frac{Ca\log\left(\tan^{2}\left(c+dx\right)+1\right)}{2d}+\frac{Ca\log\left(\tan\left(c+dx\right)\right)}{d}+Cbx & \text{otherw} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**3*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))*(B*tan(c) + C*tan(c
)**2)*cot(c)**3, Eq(d, 0)), (nan, Eq(c, -d*x)), (-B*a*x - B*a/(d*tan(c + d*
x)) - B*b*log(tan(c + d*x)**2 + 1)/(2*d) + B*b*log(tan(c + d*x))/d - C*a*lo
g(tan(c + d*x)**2 + 1)/(2*d) + C*a*log(tan(c + d*x))/d + C*b*x, True))
```

3.6 $\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+\right.$

Optimal. Leaf size=66

$$-\frac{(aC+bB)\cot(c+dx)}{d}-\frac{(aB-bC)\log(\sin(c+dx))}{d}-x(aC+bB)-\frac{aB\cot^2(c+dx)}{2d}$$

```
[Out] -(B*b+C*a)*x-(B*b+C*a)*cot(d*x+c)/d-1/2*a*B*cot(d*x+c)^2/d-(B*a-C*b)*ln(sin
(d*x+c))/d
```

Rubi [A] time = 0.16, antiderivative size = 66, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.132, Rules used = {3632, 3591, 3529, 3531, 3475}

$$-\frac{(aC+bB)\cot(c+dx)}{d}-\frac{(aB-bC)\log(\sin(c+dx))}{d}-x(aC+bB)-\frac{aB\cot^2(c+dx)}{2d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^4*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] -((b*B + a*C)*x) - ((b*B + a*C)*Cot[c + d*x])/d - (a*B*Cot[c + d*x]^2)/(2*d
) - ((a*B - b*C)*Log[Sin[c + d*x]])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3591

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((
b*c - a*d)*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^
2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[a*A*c +
 b*B*c + A*b*d - a*B*d - (A*b*c - a*B*c - a*A*d - b*B*d)*Tan[e + f*x], x],
x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c - a*d, 0] && LtQ[m,
 -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot^4(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \int \cot^3(c+dx)(a+b\tan(c+dx))(B+$$

$$= -\frac{aB\cot^2(c+dx)}{2d} + \int \cot^2(c+dx)(bB$$

$$= -\frac{(bB+aC)\cot(c+dx)}{d} - \frac{aB\cot^2(c+}{2d}$$

$$= -(bB+aC)x - \frac{(bB+aC)\cot(c+dx)}{d}$$

$$= -(bB+aC)x - \frac{(bB+aC)\cot(c+dx)}{d}$$

Mathematica [C] time = 0.47, size = 77, normalized size = 1.17

$$-\frac{2(aC+bB)\cot(c+dx)\,{}_2F_1\left(-\frac{1}{2},1;\frac{1}{2};-\tan^2(c+dx)\right)+2(aB-bC)(\log(\tan(c+dx))+\log(\cos(c+dx)))+aB}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^4*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] -1/2*(a*B*Cot[c + d*x]^2 + 2*(b*B + a*C)*Cot[c + d*x]*Hypergeometric2F1[-1/
2, 1, 1/2, -Tan[c + d*x]^2] + 2*(a*B - b*C)*(Log[Cos[c + d*x]] + Log[Tan[c
+ d*x]]))/d
```

fricas [A] time = 0.55, size = 95, normalized size = 1.44

$$-\frac{(Ba-Cb)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan(dx+c)^2+(2(Ca+Bb)dx+Ba)\tan(dx+c)^2+Ba+2(Ca+Bb)\tan(dx+c)}{2d\tan(dx+c)^2}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] -1/2*((B*a - C*b)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*tan(d*x + c)^2 +
 (2*(C*a + B*b)*d*x + B*a)*tan(d*x + c)^2 + B*a + 2*(C*a + B*b)*tan(d*x + c
))/(d*tan(d*x + c)^2)
```

giac [B] time = 5.64, size = 179, normalized size = 2.71

$$-\frac{Ba\tan\left(\frac{1}{2}dx+\frac{1}{2}c\right)^2-4Ca\tan\left(\frac{1}{2}dx+\frac{1}{2}c\right)-4Bb\tan\left(\frac{1}{2}dx+\frac{1}{2}c\right)+8(Ca+Bb)(dx+c)-8(Ba-Cb)\log\left(\mathrm{ta}\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] -1/8*(B*a*tan(1/2*d*x + 1/2*c)^2 - 4*C*a*tan(1/2*d*x + 1/2*c) - 4*B*b*tan(1
/2*d*x + 1/2*c) + 8*(C*a + B*b)*(d*x + c) - 8*(B*a - C*b)*log(tan(1/2*d*x +
 1/2*c)^2 + 1) + 8*(B*a - C*b)*log(abs(tan(1/2*d*x + 1/2*c))) - (12*B*a*tan
(1/2*d*x + 1/2*c)^2 - 12*C*b*tan(1/2*d*x + 1/2*c)^2 - 4*C*a*tan(1/2*d*x + 1
/2*c) - 4*B*b*tan(1/2*d*x + 1/2*c) - B*a)/tan(1/2*d*x + 1/2*c)^2)/d
```

maple [A] time = 0.52, size = 96, normalized size = 1.45

$$-\frac{aB\left(\cot^2(dx+c)\right)}{2d}-\frac{aB\ln(\sin(dx+c))}{d}-aCx-\frac{C\cot(dx+c)a}{d}-\frac{Cac}{d}-Bxb-\frac{B\cot(dx+c)b}{d}-\frac{Bbc}{d}+\frac{Cb\ln(\sin(dx}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^4*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -1/2*a*B*cot(d*x+c)^2/d-a*B*ln(sin(d*x+c))/d-a*C*x-1/d*C*cot(d*x+c)*a-1/d*C
*a*c-B*x*b-1/d*B*cot(d*x+c)*b-1/d*B*b*c+1/d*C*b*ln(sin(d*x+c))
```

maxima [A] time = 0.66, size = 86, normalized size = 1.30

$$-\frac{2\,(Ca+Bb)(dx+c)-(Ba-Cb)\log\left(\tan\left(dx+c\right)^2+1\right)+2\,(Ba-Cb)\log\left(\tan\left(dx+c\right)\right)+\frac{Ba+2\,(Ca+Bb)\tan(dx+c)}{\tan(dx+c)^2}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(C*a + B*b)*(d*x + c) - (B*a - C*b)*log(tan(d*x + c)^2 + 1) + 2*(B*
a - C*b)*log(tan(d*x + c)) + (B*a + 2*(C*a + B*b)*tan(d*x + c))/tan(d*x + c
)^2)/d
```

mupad [B] time = 8.94, size = 108, normalized size = 1.64

$$-\frac{\ln\left(\tan\left(c+d\,x\right)\right)\,\left(B\,a-C\,b\right)}{d}-\frac{\cot\left(c+d\,x\right)^2\,\left(\frac{B\,a}{2}+\tan\left(c+d\,x\right)\,\left(B\,b+C\,a\right)\right)}{d}+\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^4*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B + C*1i)*(a + b*1i))/(2*d) - (cot(c + d*x)^2*((B*
a)/2 + tan(c + d*x)*(B*b + C*a)))/d - (log(tan(c + d*x))*(B*a - C*b))/d - (
log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)*1i)/(2*d)
```

sympy [A] time = 2.33, size = 150, normalized size = 2.27

$$\begin{cases} \mathrm{NaN} \\ x\left(a+b\tan\left(c\right)\right)\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\cot^4\left(c\right) \\ \frac{Ba\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}-\frac{Ba\log\left(\tan\left(c+dx\right)\right)}{d}-\frac{Ba}{2d\tan^2\left(c+dx\right)}-Bbx-\frac{Bb}{d\tan\left(c+dx\right)}-Cax-\frac{Ca}{d\tan\left(c+dx\right)}-\frac{Cb\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+ \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**4*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((nan, (Eq(c, 0) | Eq(c, -d*x)) & (Eq(d, 0) | Eq(c, -d*x))), (x*(a
 + b*tan(c))*(B*tan(c) + C*tan(c)**2)*cot(c)**4, Eq(d, 0)), (B*a*log(tan(c
+ d*x)**2 + 1)/(2*d) - B*a*log(tan(c + d*x))/d - B*a/(2*d*tan(c + d*x)**2)
- B*b*x - B*b/(d*tan(c + d*x)) - C*a*x - C*a/(d*tan(c + d*x)) - C*b*log(tan
(c + d*x)**2 + 1)/(2*d) + C*b*log(tan(c + d*x))/d, True))
```

3.7 $\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+\right.$

Optimal. Leaf size=87

$$-\frac{(aC+bB)\cot^2(c+dx)}{2d}+\frac{(aB-bC)\cot(c+dx)}{d}-\frac{(aC+bB)\log(\sin(c+dx))}{d}+x(aB-bC)-\frac{aB\cot^3(c+dx)}{3d}$$

```
[Out] (B*a-C*b)*x+(B*a-C*b)*cot(d*x+c)/d-1/2*(B*b+C*a)*cot(d*x+c)^2/d-1/3*a*B*cot
(d*x+c)^3/d-(B*b+C*a)*ln(sin(d*x+c))/d
```

Rubi [A] time = 0.19, antiderivative size = 87, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.132, Rules used = {3632, 3591, 3529, 3531, 3475}

$$-\frac{(aC+bB)\cot^2(c+dx)}{2d}+\frac{(aB-bC)\cot(c+dx)}{d}-\frac{(aC+bB)\log(\sin(c+dx))}{d}+x(aB-bC)-\frac{aB\cot^3(c+dx)}{3d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^5*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] (a*B - b*C)*x + ((a*B - b*C)*Cot[c + d*x])/d - ((b*B + a*C)*Cot[c + d*x]^2)
/(2*d) - (a*B*Cot[c + d*x]^3)/(3*d) - ((b*B + a*C)*Log[Sin[c + d*x]])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3591

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((
b*c - a*d)*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^
2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[a*A*c +
 b*B*c + A*b*d - a*B*d - (A*b*c - a*B*c - a*A*d - b*B*d)*Tan[e + f*x], x],
x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c - a*d, 0] && LtQ[m,
 -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot^5(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \int \cot^4(c+dx)(a+b\tan(c+dx))(B+$$

$$= -\frac{aB\cot^3(c+dx)}{3d} + \int \cot^3(c+dx)(bB$$

$$= -\frac{(bB+aC)\cot^2(c+dx)}{2d} - \frac{aB\cot^3(c+}{3d}$$

$$= \frac{(aB-bC)\cot(c+dx)}{d} - \frac{(bB+aC)\cot}{2d}$$

$$= (aB-bC)x + \frac{(aB-bC)\cot(c+dx)}{d} -$$

$$= (aB-bC)x + \frac{(aB-bC)\cot(c+dx)}{d} -$$

Mathematica [C] time = 1.03, size = 101, normalized size = 1.16

$$-\frac{3(aC+bB)\left(\cot^2(c+dx)+2(\log(\tan(c+dx))+\log(\cos(c+dx)))\right)+2aB\cot^3(c+dx)\,{}_2F_1\left(-\frac{3}{2},1;-\frac{1}{2};-\tan^2(\right.}{6d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^5*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] -1/6*(2*a*B*Cot[c + d*x]^3*Hypergeometric2F1[-3/2, 1, -1/2, -Tan[c + d*x]^2
] + 6*b*C*Cot[c + d*x]*Hypergeometric2F1[-1/2, 1, 1/2, -Tan[c + d*x]^2] + 3
*(b*B + a*C)*(Cot[c + d*x]^2 + 2*(Log[Cos[c + d*x]] + Log[Tan[c + d*x]])))/
d
```

fricas [A] time = 0.54, size = 121, normalized size = 1.39

$$-\frac{3\,(Ca+Bb)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^3-3\,(2\,(Ba-Cb)dx-Ca-Bb)\tan\left(dx+c\right)^3-6\,(Ba-Cb)\tan\left(dx+\right.}{6\,d\tan\left(dx+c\right)^3}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] -1/6*(3*(C*a + B*b)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*tan(d*x + c)^3
 - 3*(2*(B*a - C*b)*d*x - C*a - B*b)*tan(d*x + c)^3 - 6*(B*a - C*b)*tan(d*x
 + c)^2 + 2*B*a + 3*(C*a + B*b)*tan(d*x + c))/(d*tan(d*x + c)^3)
```

giac [B] time = 7.62, size = 237, normalized size = 2.72

$$\frac{Ba\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-3\,Ca\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2-3\,Bb\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2-15\,Ba\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)+12\,Cb\tan\left(\frac{1}{2}\,dx\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] 1/24*(B*a*tan(1/2*d*x + 1/2*c)^3 - 3*C*a*tan(1/2*d*x + 1/2*c)^2 - 3*B*b*tan
(1/2*d*x + 1/2*c)^2 - 15*B*a*tan(1/2*d*x + 1/2*c) + 12*C*b*tan(1/2*d*x + 1/
2*c) + 24*(B*a - C*b)*(d*x + c) + 24*(C*a + B*b)*log(tan(1/2*d*x + 1/2*c)^2
 + 1) - 24*(C*a + B*b)*log(abs(tan(1/2*d*x + 1/2*c))) + (44*C*a*tan(1/2*d*x
 + 1/2*c)^3 + 44*B*b*tan(1/2*d*x + 1/2*c)^3 + 15*B*a*tan(1/2*d*x + 1/2*c)^2
 - 12*C*b*tan(1/2*d*x + 1/2*c)^2 - 3*C*a*tan(1/2*d*x + 1/2*c) - 3*B*b*tan(1
/2*d*x + 1/2*c) - B*a)/tan(1/2*d*x + 1/2*c)^3)/d
```

maple [A] time = 0.51, size = 124, normalized size = 1.43

$$-\frac{aB\left(\cot^3\left(dx+c\right)\right)}{3d}+\frac{aB\cot\left(dx+c\right)}{d}+aBx+\frac{Bac}{d}-\frac{aC\left(\cot^2\left(dx+c\right)\right)}{2d}-\frac{aC\ln\left(\sin\left(dx+c\right)\right)}{d}-\frac{Bb\left(\cot^2\left(dx+c\right)\right)}{2d}-\frac{Bl}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^5*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -1/3*a*B*cot(d*x+c)^3/d+a*B*cot(d*x+c)/d+a*B*x+1/d*B*a*c-1/2/d*a*C*cot(d*x+
c)^2-1/d*a*C*ln(sin(d*x+c))-1/2/d*B*b*cot(d*x+c)^2-1/d*B*b*ln(sin(d*x+c))-b
*C*x-1/d*C*cot(d*x+c)*b-1/d*C*b*c
```

maxima [A] time = 0.77, size = 104, normalized size = 1.20

$$\frac{6\,(Ba-Cb)(dx+c)+3\,(Ca+Bb)\log\left(\tan\left(dx+c\right)^2+1\right)-6\,(Ca+Bb)\log\left(\tan\left(dx+c\right)\right)+\frac{6\,(Ba-Cb)\tan(dx+c)^2-2\,B}{\tan(dx}}{6\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/6*(6*(B*a - C*b)*(d*x + c) + 3*(C*a + B*b)*log(tan(d*x + c)^2 + 1) - 6*(C
*a + B*b)*log(tan(d*x + c)) + (6*(B*a - C*b)*tan(d*x + c)^2 - 2*B*a - 3*(C*
a + B*b)*tan(d*x + c))/tan(d*x + c)^3)/d
```

mupad [B] time = 8.89, size = 127, normalized size = 1.46

$$-\frac{\cot\left(c+d\,x\right)^3\,\left(\left(C\,b-B\,a\right)\,\tan\left(c+d\,x\right)^2+\left(\frac{B\,b}{2}+\frac{C\,a}{2}\right)\,\tan\left(c+d\,x\right)+\frac{B\,a}{3}\right)}{d}-\frac{\ln\left(\tan\left(c+d\,x\right)\right)\,\left(B\,b+C\,a\right)}{d}-\frac{\ln\left(\mathrm{t}\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^5*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b))/(2*d) - (log(tan(c + d*x))*(
B*b + C*a))/d - (log(tan(c + d*x) - 1i)*(B + C*1i)*(a + b*1i)*1i)/(2*d) - (
cot(c + d*x)^3*((B*a)/3 + tan(c + d*x)*((B*b)/2 + (C*a)/2) - tan(c + d*x)^2
*(B*a - C*b)))/d
```

sympy [A] time = 4.43, size = 180, normalized size = 2.07

$$\begin{cases} \mathrm{NaN} \\ x\left(a+b\tan\left(c\right)\right)\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\cot^5\left(c\right) \\ Bax+\frac{Ba}{d\tan\left(c+dx\right)}-\frac{Ba}{3d\tan^3\left(c+dx\right)}+\frac{Bb\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}-\frac{Bb\log\left(\tan\left(c+dx\right)\right)}{d}-\frac{Bb}{2d\tan^2\left(c+dx\right)}+\frac{Ca\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}-\frac{Ca\log}{} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**5*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((nan, (Eq(c, 0) | Eq(c, -d*x)) & (Eq(d, 0) | Eq(c, -d*x))), (x*(a
 + b*tan(c))*(B*tan(c) + C*tan(c)**2)*cot(c)**5, Eq(d, 0)), (B*a*x + B*a/(d
*tan(c + d*x)) - B*a/(3*d*tan(c + d*x)**3) + B*b*log(tan(c + d*x)**2 + 1)/(
2*d) - B*b*log(tan(c + d*x))/d - B*b/(2*d*tan(c + d*x)**2) + C*a*log(tan(c
+ d*x)**2 + 1)/(2*d) - C*a*log(tan(c + d*x))/d - C*a/(2*d*tan(c + d*x)**2)
- C*b*x - C*b/(d*tan(c + d*x)), True))
```

3.8 $\int \cot^6(c+dx)(a+b\tan(c+dx)) \left(B\tan(c+dx)+C\tan^2(c\right.$

Optimal. Leaf size=108

$$-\frac{(aC+bB)\cot^3(c+dx)}{3d}+\frac{(aB-bC)\cot^2(c+dx)}{2d}+\frac{(aC+bB)\cot(c+dx)}{d}+\frac{(aB-bC)\log(\sin(c+dx))}{d}+x(aC+b$$

```
[Out] (B*b+C*a)*x+(B*b+C*a)*cot(d*x+c)/d+1/2*(B*a-C*b)*cot(d*x+c)^2/d-1/3*(B*b+C*
a)*cot(d*x+c)^3/d-1/4*a*B*cot(d*x+c)^4/d+(B*a-C*b)*ln(sin(d*x+c))/d
```

Rubi [A] time = 0.23, antiderivative size = 108, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.132, Rules used = {3632, 3591, 3529, 3531, 3475}

$$-\frac{(aC+bB)\cot^3(c+dx)}{3d}+\frac{(aB-bC)\cot^2(c+dx)}{2d}+\frac{(aC+bB)\cot(c+dx)}{d}+\frac{(aB-bC)\log(\sin(c+dx))}{d}+x(aC+b$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^6*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] (b*B + a*C)*x + ((b*B + a*C)*Cot[c + d*x])/d + ((a*B - b*C)*Cot[c + d*x]^2)
/(2*d) - ((b*B + a*C)*Cot[c + d*x]^3)/(3*d) - (a*B*Cot[c + d*x]^4)/(4*d) +
((a*B - b*C)*Log[Sin[c + d*x]])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
```

```
Q[a*c + b*d, 0]
```

Rule 3591

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((
b*c - a*d)*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^
2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[a*A*c +
 b*B*c + A*b*d - a*B*d - (A*b*c - a*B*c - a*A*d - b*B*d)*Tan[e + f*x], x],
x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c - a*d, 0] && LtQ[m,
 -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot^6(c+dx)(a+b\tan(c+dx))\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \int \cot^5(c+dx)(a+b\tan(c+dx))(B+C$$

$$= -\frac{aB\cot^4(c+dx)}{4d} + \int \cot^4(c+dx)(bB$$

$$= -\frac{(bB+aC)\cot^3(c+dx)}{3d} - \frac{aB\cot^4(c+}{4d}$$

$$= \frac{(aB-bC)\cot^2(c+dx)}{2d} - \frac{(bB+aC)\cot}{3d}$$

$$= \frac{(bB+aC)\cot(c+dx)}{d} + \frac{(aB-bC)\cot^2}{2d}$$

$$= (bB+aC)x + \frac{(bB+aC)\cot(c+dx)}{d} + $$

$$= (bB+aC)x + \frac{(bB+aC)\cot(c+dx)}{d} + $$

Mathematica [C] time = 1.15, size = 100, normalized size = 0.93

$$-\frac{4(aC+bB)\cot^3(c+dx)\,{}_2F_1\left(-\frac{3}{2},1;-\frac{1}{2};-\tan^2(c+dx)\right)+3\left((2bC-2aB)\cot^2(c+dx)-4(aB-bC)(\log(\tan(c\right.}{12d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^6*(a + b*Tan[c + d*x])*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] -1/12*(4*(b*B + a*C)*Cot[c + d*x]^3*Hypergeometric2F1[-3/2, 1, -1/2, -Tan[c
 + d*x]^2] + 3*((-2*a*B + 2*b*C)*Cot[c + d*x]^2 + a*B*Cot[c + d*x]^4 - 4*(a
*B - b*C)*(Log[Cos[c + d*x]] + Log[Tan[c + d*x]])))/d
```

fricas [A] time = 0.69, size = 138, normalized size = 1.28

$$\frac{6\,(Ba-Cb)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan(dx+c)^4+3\,(4\,(Ca+Bb)dx+3\,Ba-2\,Cb)\tan(dx+c)^4+12\,(Ca+Bb)\tan(}{12\,d\tan(dx+c)^4}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] 1/12*(6*(B*a - C*b)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*tan(d*x + c)^4
 + 3*(4*(C*a + B*b)*d*x + 3*B*a - 2*C*b)*tan(d*x + c)^4 + 12*(C*a + B*b)*ta
n(d*x + c)^3 + 6*(B*a - C*b)*tan(d*x + c)^2 - 3*B*a - 4*(C*a + B*b)*tan(d*x
 + c))/(d*tan(d*x + c)^4)
```

giac [B] time = 9.32, size = 299, normalized size = 2.77

$$-\frac{3\,Ba\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^4-8\,Ca\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-8\,Bb\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-36\,Ba\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2+24\,Cb\tan}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] -1/192*(3*B*a*tan(1/2*d*x + 1/2*c)^4 - 8*C*a*tan(1/2*d*x + 1/2*c)^3 - 8*B*b
*tan(1/2*d*x + 1/2*c)^3 - 36*B*a*tan(1/2*d*x + 1/2*c)^2 + 24*C*b*tan(1/2*d*
x + 1/2*c)^2 + 120*C*a*tan(1/2*d*x + 1/2*c) + 120*B*b*tan(1/2*d*x + 1/2*c)
- 192*(C*a + B*b)*(d*x + c) + 192*(B*a - C*b)*log(tan(1/2*d*x + 1/2*c)^2 +
```

```
1) - 192*(B*a - C*b)*log(abs(tan(1/2*d*x + 1/2*c))) + (400*B*a*tan(1/2*d*x
+ 1/2*c)^4 - 400*C*b*tan(1/2*d*x + 1/2*c)^4 - 120*C*a*tan(1/2*d*x + 1/2*c)^
3 - 120*B*b*tan(1/2*d*x + 1/2*c)^3 - 36*B*a*tan(1/2*d*x + 1/2*c)^2 + 24*C*b
*tan(1/2*d*x + 1/2*c)^2 + 8*C*a*tan(1/2*d*x + 1/2*c) + 8*B*b*tan(1/2*d*x +
1/2*c) + 3*B*a)/tan(1/2*d*x + 1/2*c)^4)/d
```

maple [A] time = 0.53, size = 150, normalized size = 1.39

$$-\frac{aB\left(\cot^4(dx+c)\right)}{4d}+\frac{aB\left(\cot^2(dx+c)\right)}{2d}+\frac{aB\ln(\sin(dx+c))}{d}-\frac{aC\left(\cot^3(dx+c)\right)}{3d}+\frac{C\cot(dx+c)\,a}{d}+aCx+\frac{Cac}{d}-$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^6*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -1/4*a*B*cot(d*x+c)^4/d+1/2*a*B*cot(d*x+c)^2/d+a*B*ln(sin(d*x+c))/d-1/3/d*a
*C*cot(d*x+c)^3+1/d*C*cot(d*x+c)*a+a*C*x+1/d*C*a*c-1/3/d*B*b*cot(d*x+c)^3+1
/d*B*cot(d*x+c)*b+B*x*b+1/d*B*b*c-1/2/d*C*b*cot(d*x+c)^2-1/d*C*b*ln(sin(d*x
+c))
```

maxima [A] time = 0.93, size = 122, normalized size = 1.13

$$\frac{12\,(Ca+Bb)(dx+c)-6\,(Ba-Cb)\log\left(\tan(dx+c)^2+1\right)+12\,(Ba-Cb)\log\left(\tan(dx+c)\right)+\frac{12\,(Ca+Bb)\tan(dx+c)^3+}{}}{12\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/12*(12*(C*a + B*b)*(d*x + c) - 6*(B*a - C*b)*log(tan(d*x + c)^2 + 1) + 12
*(B*a - C*b)*log(tan(d*x + c)) + (12*(C*a + B*b)*tan(d*x + c)^3 + 6*(B*a -
C*b)*tan(d*x + c)^2 - 3*B*a - 4*(C*a + B*b)*tan(d*x + c))/tan(d*x + c)^4)/d
```

mupad [B] time = 8.82, size = 145, normalized size = 1.34

$$\frac{\ln\left(\tan\left(c+d\,x\right)\right)\,\left(B\,a-C\,b\right)}{d}-\frac{\cot\left(c+d\,x\right)^4\,\left(\left(-B\,b-C\,a\right)\,\tan\left(c+d\,x\right)^3+\left(\frac{C\,b}{2}-\frac{B\,a}{2}\right)\,\tan\left(c+d\,x\right)^2+\left(\frac{B\,b}{3}+\frac{C\,a}{3}\right)}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^6*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x))*(B*a - C*b))/d - (cot(c + d*x)^4*((B*a)/4 + tan(c + d*x)
*((B*b)/3 + (C*a)/3) - tan(c + d*x)^3*(B*b + C*a) - tan(c + d*x)^2*((B*a)/2
```

```
 - (C*b)/2)))/d - (log(tan(c + d*x) - 1i)*(B + C*1i)*(a + b*1i))/(2*d) + (l
og(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)*1i)/(2*d)
```

sympy [A] time = 5.87, size = 211, normalized size = 1.95

$$\begin{cases} \text{NaN} \\ x(a + b\tan(c))\left(B\tan(c) + C\tan^2(c)\right)\cot^6(c) \\ -\frac{Ba\log\left(\tan^2(c+dx)+1\right)}{2d} + \frac{Ba\log(\tan(c+dx))}{d} + \frac{Ba}{2d\tan^2(c+dx)} - \frac{Ba}{4d\tan^4(c+dx)} + Bbx + \frac{Bb}{d\tan(c+dx)} - \frac{Bb}{3d\tan^3(c+dx)} + Cax + \frac{}{d\tan} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**6*(a+b*tan(d*x+c))*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((nan, (Eq(c, 0) | Eq(c, -d*x)) & (Eq(d, 0) | Eq(c, -d*x))), (x*(a
 + b*tan(c))*(B*tan(c) + C*tan(c)**2)*cot(c)**6, Eq(d, 0)), (-B*a*log(tan(c
 + d*x)**2 + 1)/(2*d) + B*a*log(tan(c + d*x))/d + B*a/(2*d*tan(c + d*x)**2)
 - B*a/(4*d*tan(c + d*x)**4) + B*b*x + B*b/(d*tan(c + d*x)) - B*b/(3*d*tan(
c + d*x)**3) + C*a*x + C*a/(d*tan(c + d*x)) - C*a/(3*d*tan(c + d*x)**3) + C
*b*log(tan(c + d*x)**2 + 1)/(2*d) - C*b*log(tan(c + d*x))/d - C*b/(2*d*tan(
c + d*x)**2), True))
```

3.9 $\int \tan(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c\right.$

Optimal. Leaf size=148

$$\frac{\left(a^2C+2abB-b^2C\right)\log(\cos(c+dx))}{d}-x\left(a^2B-2abC-b^2B\right)+\frac{(4bB-aC)(a+b\tan(c+dx))^3}{12b^2d}-\frac{b(aC+bB)\tan(c}{d}$$

```
[Out] -(B*a^2-B*b^2-2*C*a*b)*x+(2*B*a*b+C*a^2-C*b^2)*ln(cos(d*x+c))/d-b*(B*b+C*a)
*tan(d*x+c)/d-1/2*C*(a+b*tan(d*x+c))^2/d+1/12*(4*B*b-C*a)*(a+b*tan(d*x+c))^
3/b^2/d+1/4*C*tan(d*x+c)*(a+b*tan(d*x+c))^3/b/d
```

Rubi [A] time = 0.30, antiderivative size = 148, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.158, Rules used = {3632, 3607, 3630, 3528, 3525, 3475}

$$\frac{\left(a^2C+2abB-b^2C\right)\log(\cos(c+dx))}{d}-x\left(a^2B-2abC-b^2B\right)+\frac{(4bB-aC)(a+b\tan(c+dx))^3}{12b^2d}-\frac{b(aC+bB)\tan(c}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Tan[c + d*x]*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] -((a^2*B - b^2*B - 2*a*b*C)*x) + ((2*a*b*B + a^2*C - b^2*C)*Log[Cos[c + d*x
]])/d - (b*(b*B + a*C)*Tan[c + d*x])/d - (C*(a + b*Tan[c + d*x])^2)/(2*d) +
 ((4*b*B - a*C)*(a + b*Tan[c + d*x])^3)/(12*b^2*d) + (C*Tan[c + d*x]*(a + b
*Tan[c + d*x])^3)/(4*b*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
```

```
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3607

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*B*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(m
+ n)), x] + Dist[1/(d*(m + n)), Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan
[e + f*x])^n*Simp[a^2*A*d*(m + n) - b*B*(b*c*(m - 1) + a*d*(n + 1)) + d*(m
+ n)*(2*a*A*b + B*(a^2 - b^2))*Tan[e + f*x] - (b*B*(b*c - a*d)*(m - 1) - b*
(A*b + a*B)*d*(m + n))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e,
f, A, B, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2,
 0] && GtQ[m, 1] && (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(IGtQ[n, 1] &
& ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \tan(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int \tan^2(c+dx)(a+b\tan(c+dx))^2(B+$$

$$= \frac{C\tan(c+dx)(a+b\tan(c+dx))^3}{4bd} + \frac{\int(}{}$$

$$= \frac{(4bB-aC)(a+b\tan(c+dx))^3}{12b^2d} + \frac{C\tan}{}$$

$$= -\frac{C(a+b\tan(c+dx))^2}{2d} + \frac{(4bB-aC)(a}{}$$

$$= -\left(a^2B-b^2B-2abC\right)x - \frac{b(bB+aC)\tan}{d}$$

$$= -\left(a^2B-b^2B-2abC\right)x + \frac{\left(2abB+a^2C\right.}{}$$

Mathematica [C] time = 6.25, size = 221, normalized size = 1.49

$$\frac{C\tan(c+dx)(a+b\tan(c+dx))^3}{4bd} + \frac{\frac{(4bB-aC)(a+b\tan(c+dx))^3}{3bd} + \frac{2\left((bB-aC)\left(-i(a-ib)^2\log(\tan(c+dx)+i)+i(a+ib)^2\log(-\tan(c+dx)+i)-2b\right.\right.}{}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[Tan[c + d*x]*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] (C*Tan[c + d*x]*(a + b*Tan[c + d*x])^3)/(4*b*d) + (((4*b*B - a*C)*(a + b*Ta
n[c + d*x])^3)/(3*b*d) + (2*((b*B - a*C)*(I*(a + I*b)^2*Log[I - Tan[c + d*x
]] - I*(a - I*b)^2*Log[I + Tan[c + d*x]] - 2*b^2*Tan[c + d*x]) - C*((I*a -
b)^3*Log[I - Tan[c + d*x]] - (I*a + b)^3*Log[I + Tan[c + d*x]] + 6*a*b^2*Ta
n[c + d*x] + b^3*Tan[c + d*x]^2)))/d)/(4*b)
```

fricas [A] time = 0.60, size = 146, normalized size = 0.99

$$\frac{3\,Cb^2\tan(dx+c)^4 + 4\left(2\,Cab+Bb^2\right)\tan(dx+c)^3 - 12\left(Ba^2-2\,Cab-Bb^2\right)dx + 6\left(Ca^2+2\,Bab-Cb^2\right)\tan(dx}{12\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] 1/12*(3*C*b^2*tan(d*x + c)^4 + 4*(2*C*a*b + B*b^2)*tan(d*x + c)^3 - 12*(B*a
^2 - 2*C*a*b - B*b^2)*d*x + 6*(C*a^2 + 2*B*a*b - C*b^2)*tan(d*x + c)^2 + 6*
```

```
(C*a^2 + 2*B*a*b - C*b^2)*log(1/(tan(d*x + c)^2 + 1)) + 12*(B*a^2 - 2*C*a*b
 - B*b^2)*tan(d*x + c))/d
```

giac [B] time = 14.82, size = 2228, normalized size = 15.05

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] -1/12*(12*B*a^2*d*x*tan(d*x)^4*tan(c)^4 - 24*C*a*b*d*x*tan(d*x)^4*tan(c)^4
- 12*B*b^2*d*x*tan(d*x)^4*tan(c)^4 - 6*C*a^2*log(4*(tan(d*x)^4*tan(c)^2 - 2
*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) +
 1)/(tan(c)^2 + 1))*tan(d*x)^4*tan(c)^4 - 12*B*a*b*log(4*(tan(d*x)^4*tan(c)
^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*ta
n(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^4*tan(c)^4 + 6*C*b^2*log(4*(tan(d*x)^4*t
an(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*
x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^4*tan(c)^4 - 48*B*a^2*d*x*tan(d*x)^
3*tan(c)^3 + 96*C*a*b*d*x*tan(d*x)^3*tan(c)^3 + 48*B*b^2*d*x*tan(d*x)^3*tan
(c)^3 - 6*C*a^2*tan(d*x)^4*tan(c)^4 - 12*B*a*b*tan(d*x)^4*tan(c)^4 + 9*C*b^
2*tan(d*x)^4*tan(c)^4 + 24*C*a^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*
tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^
2 + 1))*tan(d*x)^3*tan(c)^3 + 48*B*a*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d
*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(t
an(c)^2 + 1))*tan(d*x)^3*tan(c)^3 - 24*C*b^2*log(4*(tan(d*x)^4*tan(c)^2 - 2
*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) +
 1)/(tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 + 12*B*a^2*tan(d*x)^4*tan(c)^3 - 24
*C*a*b*tan(d*x)^4*tan(c)^3 - 12*B*b^2*tan(d*x)^4*tan(c)^3 + 12*B*a^2*tan(d*
x)^3*tan(c)^4 - 24*C*a*b*tan(d*x)^3*tan(c)^4 - 12*B*b^2*tan(d*x)^3*tan(c)^4
 + 72*B*a^2*d*x*tan(d*x)^2*tan(c)^2 - 144*C*a*b*d*x*tan(d*x)^2*tan(c)^2 - 7
2*B*b^2*d*x*tan(d*x)^2*tan(c)^2 - 6*C*a^2*tan(d*x)^4*tan(c)^2 - 12*B*a*b*ta
n(d*x)^4*tan(c)^2 + 6*C*b^2*tan(d*x)^4*tan(c)^2 + 12*C*a^2*tan(d*x)^3*tan(c
)^3 + 24*B*a*b*tan(d*x)^3*tan(c)^3 - 24*C*b^2*tan(d*x)^3*tan(c)^3 - 6*C*a^2
*tan(d*x)^2*tan(c)^4 - 12*B*a*b*tan(d*x)^2*tan(c)^4 + 6*C*b^2*tan(d*x)^2*ta
n(c)^4 + 8*C*a*b*tan(d*x)^4*tan(c) + 4*B*b^2*tan(d*x)^4*tan(c) - 36*C*a^2*l
og(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan
(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 - 72*B
*a*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2
 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2
+ 36*C*b^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*ta
n(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan
(c)^2 - 36*B*a^2*tan(d*x)^3*tan(c)^2 + 96*C*a*b*tan(d*x)^3*tan(c)^2 + 48*B*
b^2*tan(d*x)^3*tan(c)^2 - 36*B*a^2*tan(d*x)^2*tan(c)^3 + 96*C*a*b*tan(d*x)^
2*tan(c)^3 + 48*B*b^2*tan(d*x)^2*tan(c)^3 + 8*C*a*b*tan(d*x)*tan(c)^4 + 4*B
```

```
*b^2*tan(d*x)*tan(c)^4 - 3*C*b^2*tan(d*x)^4 - 48*B*a^2*d*x*tan(d*x)*tan(c)
+ 96*C*a*b*d*x*tan(d*x)*tan(c) + 48*B*b^2*d*x*tan(d*x)*tan(c) + 12*C*a^2*ta
n(d*x)^3*tan(c) + 24*B*a*b*tan(d*x)^3*tan(c) - 24*C*b^2*tan(d*x)^3*tan(c) -
 12*C*a^2*tan(d*x)^2*tan(c)^2 - 24*B*a*b*tan(d*x)^2*tan(c)^2 + 12*C*b^2*tan
(d*x)^2*tan(c)^2 + 12*C*a^2*tan(d*x)*tan(c)^3 + 24*B*a*b*tan(d*x)*tan(c)^3
- 24*C*b^2*tan(d*x)*tan(c)^3 - 3*C*b^2*tan(c)^4 - 8*C*a*b*tan(d*x)^3 - 4*B*
b^2*tan(d*x)^3 + 24*C*a^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c)
+ tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))
*tan(d*x)*tan(c) + 48*B*a*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c
) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1
))*tan(d*x)*tan(c) - 24*C*b^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan
(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 +
 1))*tan(d*x)*tan(c) + 36*B*a^2*tan(d*x)^2*tan(c) - 96*C*a*b*tan(d*x)^2*tan
(c) - 48*B*b^2*tan(d*x)^2*tan(c) + 36*B*a^2*tan(d*x)*tan(c)^2 - 96*C*a*b*ta
n(d*x)*tan(c)^2 - 48*B*b^2*tan(d*x)*tan(c)^2 - 8*C*a*b*tan(c)^3 - 4*B*b^2*t
an(c)^3 + 12*B*a^2*d*x - 24*C*a*b*d*x - 12*B*b^2*d*x - 6*C*a^2*tan(d*x)^2 -
 12*B*a*b*tan(d*x)^2 + 6*C*b^2*tan(d*x)^2 + 12*C*a^2*tan(d*x)*tan(c) + 24*B
*a*b*tan(d*x)*tan(c) - 24*C*b^2*tan(d*x)*tan(c) - 6*C*a^2*tan(c)^2 - 12*B*a
*b*tan(c)^2 + 6*C*b^2*tan(c)^2 - 6*C*a^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan
(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/
(tan(c)^2 + 1)) - 12*B*a*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c)
 + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1)
) + 6*C*b^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*t
an(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1)) - 12*B*a^2*ta
n(d*x) + 24*C*a*b*tan(d*x) + 12*B*b^2*tan(d*x) - 12*B*a^2*tan(c) + 24*C*a*b
*tan(c) + 12*B*b^2*tan(c) - 6*C*a^2 - 12*B*a*b + 9*C*b^2)/(d*tan(d*x)^4*tan
(c)^4 - 4*d*tan(d*x)^3*tan(c)^3 + 6*d*tan(d*x)^2*tan(c)^2 - 4*d*tan(d*x)*ta
n(c) + d)
```

maple [A] time = 0.02, size = 249, normalized size = 1.68

$$\frac{b^2C\left(\tan^4(dx+c)\right)}{4d}+\frac{B\left(\tan^3(dx+c)\right)b^2}{3d}+\frac{2C\left(\tan^3(dx+c)\right)ab}{3d}+\frac{B\left(\tan^2(dx+c)\right)ab}{d}+\frac{C\left(\tan^2(dx+c)\right)a^2}{2d}-\frac{b^2C}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] 1/4/d*b^2*C*tan(d*x+c)^4+1/3/d*B*tan(d*x+c)^3*b^2+2/3/d*C*tan(d*x+c)^3*a*b+
1/d*B*tan(d*x+c)^2*a*b+1/2/d*C*tan(d*x+c)^2*a^2-1/2/d*b^2*C*tan(d*x+c)^2+a^
2*B*tan(d*x+c)/d-b^2*B*tan(d*x+c)/d-2/d*C*a*b*tan(d*x+c)-1/d*ln(1+tan(d*x+c
)^2)*B*a*b-1/2/d*ln(1+tan(d*x+c)^2)*a^2*C+1/2/d*ln(1+tan(d*x+c)^2)*b^2*C-1/
d*B*arctan(tan(d*x+c))*a^2+1/d*B*arctan(tan(d*x+c))*b^2+2/d*C*arctan(tan(d*
x+c))*a*b
```

maxima [A] time = 0.63, size = 147, normalized size = 0.99

$$\frac{3\,Cb^2\tan(dx+c)^4+4\left(2\,Cab+Bb^2\right)\tan(dx+c)^3+6\left(Ca^2+2\,Bab-Cb^2\right)\tan(dx+c)^2-12\left(Ba^2-2\,Cab-B\right.}{12\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/12*(3*C*b^2*tan(d*x + c)^4 + 4*(2*C*a*b + B*b^2)*tan(d*x + c)^3 + 6*(C*a^
2 + 2*B*a*b - C*b^2)*tan(d*x + c)^2 - 12*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c
) - 6*(C*a^2 + 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1) + 12*(B*a^2 - 2*C*a
*b - B*b^2)*tan(d*x + c))/d
```

mupad [B] time = 8.84, size = 151, normalized size = 1.02

$$x\left(-B\,a^2+2\,C\,a\,b+B\,b^2\right)+\frac{\tan(c+d\,x)^3\left(\frac{B\,b^2}{3}+\frac{2\,C\,a\,b}{3}\right)}{d}-\frac{\tan(c+d\,x)\left(-B\,a^2+2\,C\,a\,b+B\,b^2\right)}{d}-\frac{\ln\left(\tan(c+d\,x\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^2
,x)
```

```
[Out] x*(B*b^2 - B*a^2 + 2*C*a*b) + (tan(c + d*x)^3*((B*b^2)/3 + (2*C*a*b)/3))/d
- (tan(c + d*x)*(B*b^2 - B*a^2 + 2*C*a*b))/d - (log(tan(c + d*x)^2 + 1)*((C
*a^2)/2 - (C*b^2)/2 + B*a*b))/d + (tan(c + d*x)^2*((C*a^2)/2 - (C*b^2)/2 +
B*a*b))/d + (C*b^2*tan(c + d*x)^4)/(4*d)
```

sympy [A] time = 0.60, size = 250, normalized size = 1.69

$$\begin{cases} -Ba^2x+\frac{Ba^2\tan(c+dx)}{d}-\frac{Bab\log\left(\tan^2(c+dx)+1\right)}{d}+\frac{Bab\tan^2(c+dx)}{d}+Bb^2x+\frac{Bb^2\tan^3(c+dx)}{3d}-\frac{Bb^2\tan(c+dx)}{d}-\frac{Ca^2\log\left(\tan^2(c+\right.}{2d} \\ x\left(a+b\tan(c)\right)^2\left(B\tan(c)+C\tan^2(c)\right)\tan(c) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((-B*a**2*x + B*a**2*tan(c + d*x)/d - B*a*b*log(tan(c + d*x)**2 +
1)/d + B*a*b*tan(c + d*x)**2/d + B*b**2*x + B*b**2*tan(c + d*x)**3/(3*d) -
B*b**2*tan(c + d*x)/d - C*a**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*a**2*tan(
c + d*x)**2/(2*d) + 2*C*a*b*x + 2*C*a*b*tan(c + d*x)**3/(3*d) - 2*C*a*b*tan
(c + d*x)/d + C*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*b**2*tan(c + d*x)**
4/(4*d) - C*b**2*tan(c + d*x)**2/(2*d), Ne(d, 0)), (x*(a + b*tan(c))**2*(B*
tan(c) + C*tan(c)**2)*tan(c), True))
```

3.10 $\int (a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$

Optimal. Leaf size=112

$$-\frac{\left(a^2B-2abC-b^2B\right)\log(\cos(c+dx))}{d}-x\left(a^2C+2abB-b^2C\right)+\frac{b(aB-bC)\tan(c+dx)}{d}+\frac{B(a+b\tan(c+dx))^2}{2d}+$$

```
[Out] -(2*B*a*b+C*a^2-C*b^2)*x-(B*a^2-B*b^2-2*C*a*b)*ln(cos(d*x+c))/d+b*(B*a-C*b)
*tan(d*x+c)/d+1/2*B*(a+b*tan(d*x+c))^2/d+1/3*C*(a+b*tan(d*x+c))^3/b/d
```

Rubi [A] time = 0.11, antiderivative size = 112, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 32, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3630, 3528, 3525, 3475}

$$-\frac{\left(a^2B-2abC-b^2B\right)\log(\cos(c+dx))}{d}-x\left(a^2C+2abB-b^2C\right)+\frac{b(aB-bC)\tan(c+dx)}{d}+\frac{B(a+b\tan(c+dx))^2}{2d}+$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] -((2*a*b*B + a^2*C - b^2*C)*x) - ((a^2*B - b^2*B - 2*a*b*C)*Log[Cos[c + d*x
]])/d + (b*(a*B - b*C)*Tan[c + d*x])/d + (B*(a + b*Tan[c + d*x])^2)/(2*d) +
 (C*(a + b*Tan[c + d*x])^3)/(3*b*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int (a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx=\frac{C(a+b\tan(c+dx))^3}{3bd}+\int (a+b\tan(c+dx))^2(-$$
$$=\frac{B(a+b\tan(c+dx))^2}{2d}+\frac{C(a+b\tan(c+dx))^3}{3bd}+$$
$$=-\left(2abB+a^2C-b^2C\right)x+\frac{b(aB-bC)\tan(c+dx)}{d}$$
$$=-\left(2abB+a^2C-b^2C\right)x-\frac{\left(a^2B-b^2B-2abC\right)\log}{d}$$

Mathematica [C] time = 1.85, size = 172, normalized size = 1.54

$$\frac{3(aB+bC)\left(-2b^2\tan(c+dx)+i\left((a+ib)^2\log(-\tan(c+dx)+i)-(a-ib)^2\log(\tan(c+dx)+i)\right)\right)+3B\left(6ab^2\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] (2*C*(a + b*Tan[c + d*x])^3 + 3*(a*B + b*C)*(I*((a + I*b)^2*Log[I - Tan[c +
 d*x]] - (a - I*b)^2*Log[I + Tan[c + d*x]]) - 2*b^2*Tan[c + d*x]) + 3*B*((I
*a - b)^3*Log[I - Tan[c + d*x]] - (I*a + b)^3*Log[I + Tan[c + d*x]] + 6*a*b
^2*Tan[c + d*x] + b^3*Tan[c + d*x]^2))/(6*b*d)
```

fricas [A] time = 0.67, size = 119, normalized size = 1.06

$$\frac{2\,Cb^2\tan(dx+c)^3-6\left(Ca^2+2\,Bab-Cb^2\right)dx+3\left(2\,Cab+Bb^2\right)\tan(dx+c)^2-3\left(Ba^2-2\,Cab-Bb^2\right)\log\left(\frac{}{\tan}\right.}{6\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="fr
icas")
```

```
[Out] 1/6*(2*C*b^2*tan(d*x + c)^3 - 6*(C*a^2 + 2*B*a*b - C*b^2)*d*x + 3*(2*C*a*b
+ B*b^2)*tan(d*x + c)^2 - 3*(B*a^2 - 2*C*a*b - B*b^2)*log(1/(tan(d*x + c)^2
 + 1)) + 6*(C*a^2 + 2*B*a*b - C*b^2)*tan(d*x + c))/d
```

giac [B] time = 5.94, size = 1509, normalized size = 13.47

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="gi
ac")
```

```
[Out] -1/6*(6*C*a^2*d*x*tan(d*x)^3*tan(c)^3 + 12*B*a*b*d*x*tan(d*x)^3*tan(c)^3 -
6*C*b^2*d*x*tan(d*x)^3*tan(c)^3 + 3*B*a^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*ta
n(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)
/(tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 - 6*C*a*b*log(4*(tan(d*x)^4*tan(c)^2 -
 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c)
 + 1)/(tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 - 3*B*b^2*log(4*(tan(d*x)^4*tan(c
)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*t
an(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 - 18*C*a^2*d*x*tan(d*x)^2*ta
n(c)^2 - 36*B*a*b*d*x*tan(d*x)^2*tan(c)^2 + 18*C*b^2*d*x*tan(d*x)^2*tan(c)^
2 - 6*C*a*b*tan(d*x)^3*tan(c)^3 - 3*B*b^2*tan(d*x)^3*tan(c)^3 - 9*B*a^2*log
(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d
*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 + 18*C*a
*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 +
 tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2 +
9*B*b^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c
)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan(c)
^2 + 6*C*a^2*tan(d*x)^3*tan(c)^2 + 12*B*a*b*tan(d*x)^3*tan(c)^2 - 6*C*b^2*t
an(d*x)^3*tan(c)^2 + 6*C*a^2*tan(d*x)^2*tan(c)^3 + 12*B*a*b*tan(d*x)^2*tan(
c)^3 - 6*C*b^2*tan(d*x)^2*tan(c)^3 + 18*C*a^2*d*x*tan(d*x)*tan(c) + 36*B*a*
b*d*x*tan(d*x)*tan(c) - 18*C*b^2*d*x*tan(d*x)*tan(c) - 6*C*a*b*tan(d*x)^3*t
an(c) - 3*B*b^2*tan(d*x)^3*tan(c) + 6*C*a*b*tan(d*x)^2*tan(c)^2 + 3*B*b^2*t
an(d*x)^2*tan(c)^2 - 6*C*a*b*tan(d*x)*tan(c)^3 - 3*B*b^2*tan(d*x)*tan(c)^3
+ 2*C*b^2*tan(d*x)^3 + 9*B*a^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*ta
n(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2
+ 1))*tan(d*x)*tan(c) - 18*C*a*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*
tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^
2 + 1))*tan(d*x)*tan(c) - 9*B*b^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3
*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)
^2 + 1))*tan(d*x)*tan(c) - 12*C*a^2*tan(d*x)^2*tan(c) - 24*B*a*b*tan(d*x)^2
*tan(c) + 18*C*b^2*tan(d*x)^2*tan(c) - 12*C*a^2*tan(d*x)*tan(c)^2 - 24*B*a*
b*tan(d*x)*tan(c)^2 + 18*C*b^2*tan(d*x)*tan(c)^2 + 2*C*b^2*tan(c)^3 - 6*C*a
^2*d*x - 12*B*a*b*d*x + 6*C*b^2*d*x + 6*C*a*b*tan(d*x)^2 + 3*B*b^2*tan(d*x)
^2 - 6*C*a*b*tan(d*x)*tan(c) - 3*B*b^2*tan(d*x)*tan(c) + 6*C*a*b*tan(c)^2 +
```

```
3*B*b^2*tan(c)^2 - 3*B*a^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c
) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1
)) + 6*C*a*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*
tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1)) + 3*B*b^2*lo
g(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(
d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1)) + 6*C*a^2*tan(d*x) + 12*B*a
*b*tan(d*x) - 6*C*b^2*tan(d*x) + 6*C*a^2*tan(c) + 12*B*a*b*tan(c) - 6*C*b^2
*tan(c) + 6*C*a*b + 3*B*b^2)/(d*tan(d*x)^3*tan(c)^3 - 3*d*tan(d*x)^2*tan(c)
^2 + 3*d*tan(d*x)*tan(c) - d)
```

maple [A] time = 0.03, size = 199, normalized size = 1.78

$$\frac{b^2C\left(\tan^3(dx+c)\right)}{3d}+\frac{B\left(\tan^2(dx+c)\right)b^2}{2d}+\frac{C\left(\tan^2(dx+c)\right)ab}{d}+\frac{2B\tan(dx+c)ab}{d}+\frac{C\tan(dx+c)a^2}{d}-\frac{b^2C\tan}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] 1/3/d*b^2*C*tan(d*x+c)^3+1/2/d*B*tan(d*x+c)^2*b^2+1/d*C*tan(d*x+c)^2*a*b+2/
d*B*tan(d*x+c)*a*b+1/d*C*tan(d*x+c)*a^2-b^2*C*tan(d*x+c)/d+1/2/d*ln(1+tan(d
*x+c)^2)*a^2*B-1/2/d*ln(1+tan(d*x+c)^2)*b^2*B-1/d*ln(1+tan(d*x+c)^2)*C*a*b-
2/d*B*arctan(tan(d*x+c))*a*b-1/d*C*arctan(tan(d*x+c))*a^2+1/d*C*arctan(tan(
d*x+c))*b^2
```

maxima [A] time = 0.80, size = 120, normalized size = 1.07

$$\frac{2\,Cb^2\tan(dx+c)^3+3\left(2\,Cab+Bb^2\right)\tan(dx+c)^2-6\left(Ca^2+2\,Bab-Cb^2\right)(dx+c)+3\left(Ba^2-2\,Cab-Bb^2\right)\mathrm{lo}}{6\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="ma
xima")
```

```
[Out] 1/6*(2*C*b^2*tan(d*x + c)^3 + 3*(2*C*a*b + B*b^2)*tan(d*x + c)^2 - 6*(C*a^2
 + 2*B*a*b - C*b^2)*(d*x + c) + 3*(B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c
)^2 + 1) + 6*(C*a^2 + 2*B*a*b - C*b^2)*tan(d*x + c))/d
```

mupad [B] time = 8.80, size = 121, normalized size = 1.08

$$\frac{\tan(c+d\,x)^2\,\left(\frac{B\,b^2}{2}+C\,a\,b\right)}{d}-x\,\left(C\,a^2+2\,B\,a\,b-C\,b^2\right)+\frac{\tan(c+d\,x)\,\left(C\,a^2+2\,B\,a\,b-C\,b^2\right)}{d}-\frac{\ln\left(\tan(c+d\,x)^2\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^2,x)
```

```
[Out] (tan(c + d*x)^2*((B*b^2)/2 + C*a*b))/d - x*(C*a^2 - C*b^2 + 2*B*a*b) + (tan
(c + d*x)*(C*a^2 - C*b^2 + 2*B*a*b))/d - (log(tan(c + d*x)^2 + 1)*((B*b^2)/
2 - (B*a^2)/2 + C*a*b))/d + (C*b^2*tan(c + d*x)^3)/(3*d)
```

sympy [A] time = 0.43, size = 194, normalized size = 1.73

$$\begin{cases} \frac{Ba^2\log\left(\tan^2(c+dx)+1\right)}{2d} - 2Babx + \frac{2Bab\tan(c+dx)}{d} - \frac{Bb^2\log\left(\tan^2(c+dx)+1\right)}{2d} + \frac{Bb^2\tan^2(c+dx)}{2d} - Ca^2x + \frac{Ca^2\tan(c+dx)}{d} - \frac{Cab\log(}{} \\ x\,(a+b\tan(c))^2\left(B\tan(c)+C\tan^2(c)\right) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a**2*log(tan(c + d*x)**2 + 1)/(2*d) - 2*B*a*b*x + 2*B*a*b*tan(
c + d*x)/d - B*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + B*b**2*tan(c + d*x)**2
/(2*d) - C*a**2*x + C*a**2*tan(c + d*x)/d - C*a*b*log(tan(c + d*x)**2 + 1)/
d + C*a*b*tan(c + d*x)**2/d + C*b**2*x + C*b**2*tan(c + d*x)**3/(3*d) - C*b
**2*tan(c + d*x)/d, Ne(d, 0)), (x*(a + b*tan(c))**2*(B*tan(c) + C*tan(c)**2
), True))
```

3.11 $\int \cot(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2(\right.$

Optimal. Leaf size=87

$$-\frac{\left(a^2C+2abB-b^2C\right)\log(\cos(c+dx))}{d}+x\left(a^2B-2abC-b^2B\right)+\frac{b(aC+bB)\tan(c+dx)}{d}+\frac{C(a+b\tan(c+dx))^2}{2d}$$

```
[Out] (B*a^2-B*b^2-2*C*a*b)*x-(2*B*a*b+C*a^2-C*b^2)*ln(cos(d*x+c))/d+b*(B*b+C*a)*
tan(d*x+c)/d+1/2*C*(a+b*tan(d*x+c))^2/d
```

Rubi [A] time = 0.14, antiderivative size = 87, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3632, 3528, 3525, 3475}

$$-\frac{\left(a^2C+2abB-b^2C\right)\log(\cos(c+dx))}{d}+x\left(a^2B-2abC-b^2B\right)+\frac{b(aC+bB)\tan(c+dx)}{d}+\frac{C(a+b\tan(c+dx))^2}{2d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] (a^2*B - b^2*B - 2*a*b*C)*x - ((2*a*b*B + a^2*C - b^2*C)*Log[Cos[c + d*x]])
/d + (b*(b*B + a*C)*Tan[c + d*x])/d + (C*(a + b*Tan[c + d*x])^2)/(2*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx=\int(a+b\tan(c+dx))^2(B+C\tan(c+dx$$

$$=\frac{C(a+b\tan(c+dx))^2}{2d}+\int(a+b\tan(c$$

$$=\left(a^2B-b^2B-2abC\right)x+\frac{b(bB+aC)\tan}{d}$$

$$=\left(a^2B-b^2B-2abC\right)x-\frac{\left(2abB+a^2C-\right.}{}$$

Mathematica [C] time = 0.47, size = 96, normalized size = 1.10

$$\frac{2b(2aC+bB)\tan(c+dx)+(a-ib)^2(C+iB)\log(\tan(c+dx)+i)+(a+ib)^2(C-iB)\log(-\tan(c+dx)+i)+b^2}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] ((a + I*b)^2*((-I)*B + C)*Log[I - Tan[c + d*x]] + (a - I*b)^2*(I*B + C)*Log
[I + Tan[c + d*x]] + 2*b*(b*B + 2*a*C)*Tan[c + d*x] + b^2*C*Tan[c + d*x]^2)
/(2*d)
```

fricas [A] time = 1.51, size = 91, normalized size = 1.05

$$\frac{Cb^2\tan(dx+c)^2+2\left(Ba^2-2\,Cab-Bb^2\right)dx-\left(Ca^2+2\,Bab-Cb^2\right)\log\left(\frac{1}{\tan(dx+c)^2+1}\right)+2\left(2\,Cab+Bb^2\right)\tan(dx}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] 1/2*(C*b^2*tan(d*x + c)^2 + 2*(B*a^2 - 2*C*a*b - B*b^2)*d*x - (C*a^2 + 2*B*
a*b - C*b^2)*log(1/(tan(d*x + c)^2 + 1)) + 2*(2*C*a*b + B*b^2)*tan(d*x + c)
)/d
```

giac [A] time = 4.43, size = 95, normalized size = 1.09

$$\frac{Cb^2\tan(dx+c)^2 + 4\,Cab\tan(dx+c) + 2\,Bb^2\tan(dx+c) + 2\left(Ba^2 - 2\,Cab - Bb^2\right)(dx+c) + \left(Ca^2 + 2\,Bab - C\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] 1/2*(C*b^2*tan(d*x + c)^2 + 4*C*a*b*tan(d*x + c) + 2*B*b^2*tan(d*x + c) + 2
*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c) + (C*a^2 + 2*B*a*b - C*b^2)*log(tan(d*
x + c)^2 + 1))/d
```

maple [A] time = 0.50, size = 140, normalized size = 1.61

$$a^2Bx + \frac{B\,a^2c}{d} - \frac{a^2C\ln(\cos(dx+c))}{d} - \frac{2Bab\ln(\cos(dx+c))}{d} - 2abCx + \frac{2Cab\tan(dx+c)}{d} - \frac{2Cabc}{d} - Bx\,b^2 + \frac{b^2B\tan}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] a^2*B*x+1/d*B*a^2*c-1/d*a^2*C*ln(cos(d*x+c))-2/d*B*a*b*ln(cos(d*x+c))-2*a*b
*C*x+2/d*C*a*b*tan(d*x+c)-2/d*C*a*b*c-B*x*b^2+b^2*B*tan(d*x+c)/d-1/d*B*b^2*
c+1/2/d*b^2*C*tan(d*x+c)^2+b^2*C*ln(cos(d*x+c))/d
```

maxima [A] time = 0.80, size = 91, normalized size = 1.05

$$\frac{Cb^2\tan(dx+c)^2 + 2\left(Ba^2 - 2\,Cab - Bb^2\right)(dx+c) + \left(Ca^2 + 2\,Bab - Cb^2\right)\log\left(\tan(dx+c)^2 + 1\right) + 2\left(2\,Cab + B\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/2*(C*b^2*tan(d*x + c)^2 + 2*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c) + (C*a^2
+ 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1) + 2*(2*C*a*b + B*b^2)*tan(d*x +
c))/d
```

mupad [B] time = 8.85, size = 91, normalized size = 1.05

$$\frac{\ln\left(\tan(c+d\,x)^2+1\right)\,\left(\frac{C\,a^2}{2} + B\,a\,b - \frac{C\,b^2}{2}\right)}{d} - x\,\left(-B\,a^2 + 2\,C\,a\,b + B\,b^2\right) + \frac{\tan(c+d\,x)\,\left(B\,b^2 + 2\,C\,a\,b\right)}{d} + \frac{C\,b^2\tan}{2}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^2
,x)
```

```
[Out] (log(tan(c + d*x)^2 + 1)*((C*a^2)/2 - (C*b^2)/2 + B*a*b))/d - x*(B*b^2 - B*
a^2 + 2*C*a*b) + (tan(c + d*x)*(B*b^2 + 2*C*a*b))/d + (C*b^2*tan(c + d*x)^2
)/(2*d)
```

sympy [A] time = 1.09, size = 151, normalized size = 1.74

$$\begin{cases} Ba^2x + \frac{Bab\log\left(\tan^2(c+dx)+1\right)}{d} - Bb^2x + \frac{Bb^2\tan(c+dx)}{d} + \frac{Ca^2\log\left(\tan^2(c+dx)+1\right)}{2d} - 2Cabx + \frac{2Cab\tan(c+dx)}{d} - \frac{Cb^2\log\left(\tan^2(c+d\right.}{2d} \\ x\left(a+b\tan(c)\right)^2\left(B\tan(c)+C\tan^2(c)\right)\cot(c) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a**2*x + B*a*b*log(tan(c + d*x)**2 + 1)/d - B*b**2*x + B*b**2*
tan(c + d*x)/d + C*a**2*log(tan(c + d*x)**2 + 1)/(2*d) - 2*C*a*b*x + 2*C*a*
b*tan(c + d*x)/d - C*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*b**2*tan(c + d
*x)**2/(2*d), Ne(d, 0)), (x*(a + b*tan(c))**2*(B*tan(c) + C*tan(c)**2)*cot(
c), True))
```

3.12 $\int \cot^2(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2\right.$

Optimal. Leaf size=70

$$x\left(a^2C+2abB-b^2C\right)+\frac{a^2B\log(\sin(c+dx))}{d}-\frac{b(2aC+bB)\log(\cos(c+dx))}{d}+\frac{b^2C\tan(c+dx)}{d}$$

```
[Out] (2*B*a*b+C*a^2-C*b^2)*x-b*(B*b+2*C*a)*ln(cos(d*x+c))/d+a^2*B*ln(sin(d*x+c))
/d+b^2*C*tan(d*x+c)/d
```

Rubi [A] time = 0.18, antiderivative size = 70, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.100, Rules used = {3632, 3606, 3624, 3475}

$$x\left(a^2C+2abB-b^2C\right)+\frac{a^2B\log(\sin(c+dx))}{d}-\frac{b(2aC+bB)\log(\cos(c+dx))}{d}+\frac{b^2C\tan(c+dx)}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^2*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (2*a*b*B + a^2*C - b^2*C)*x - (b*(b*B + 2*a*C)*Log[Cos[c + d*x]])/d + (a^2*
B*Log[Sin[c + d*x]])/d + (b^2*C*Tan[c + d*x])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[(((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (
f_.)*(x_)]))/((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(b^
2*B*Tan[e + f*x])/(d*f), x] + Dist[1/d, Int[(a^2*A*d - b^2*B*c + (2*a*A*b +
 B*(a^2 - b^2))*d*Tan[e + f*x] + (A*b^2*d - b*B*(b*c - 2*a*d))*Tan[e + f*x]
^2)/(c + d*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && N
eQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rule 3624

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/tan[(e_.) + (f_.)*(x_)], x_Symbol] :> Simp[B*x, x] + (Dist[A, Int[1/Tan[e
 + f*x], x], x] + Dist[C, Int[Tan[e + f*x], x], x]) /; FreeQ[{e, f, A, B, C
}, x] && NeQ[A, C]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int \cot(c+dx)(a+b\tan(c+dx))^2(B+$$
$$= \frac{b^2C\tan(c+dx)}{d} + \int \cot(c+dx)\left(a^2B\right.$$
$$= \left(2abB+a^2C-b^2C\right)x + \frac{b^2C\tan(c+d}{d}$$
$$= \left(2abB+a^2C-b^2C\right)x - \frac{b(bB+2aC)1}{}$$

Mathematica [C] time = 0.28, size = 91, normalized size = 1.30

$$-\frac{-2a^2B\log(\tan(c+dx)) + (a+ib)^2(B+iC)\log(-\tan(c+dx)+i) + (a-ib)^2(B-iC)\log(\tan(c+dx)+i) - 2b^2}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^2*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] -1/2*((a + I*b)^2*(B + I*C)*Log[I - Tan[c + d*x]] - 2*a^2*B*Log[Tan[c + d*x
]] + (a - I*b)^2*(B - I*C)*Log[I + Tan[c + d*x]] - 2*b^2*C*Tan[c + d*x])/d
```

fricas [A] time = 0.77, size = 92, normalized size = 1.31

$$\frac{Ba^2\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right) + 2Cb^2\tan(dx+c) + 2\left(Ca^2+2Bab-Cb^2\right)dx - \left(2Cab+Bb^2\right)\log\left(\frac{1}{\tan(dx+c)^2+1}\right)}{2d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/2*(B*a^2*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1)) + 2*C*b^2*tan(d*x + c)
+ 2*(C*a^2 + 2*B*a*b - C*b^2)*d*x - (2*C*a*b + B*b^2)*log(1/(tan(d*x + c)^2
 + 1)))/d
```

giac [A] time = 7.51, size = 86, normalized size = 1.23

$$\frac{2\,Ba^2\log\left(|\tan\left(dx+c\right)|\right)+2\,Cb^2\tan\left(dx+c\right)+2\left(Ca^2+2\,Bab-Cb^2\right)(dx+c)-\left(Ba^2-2\,Cab-Bb^2\right)\log\left(\tan\left(\right.\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/2*(2*B*a^2*log(abs(tan(d*x + c))) + 2*C*b^2*tan(d*x + c) + 2*(C*a^2 + 2*B
*a*b - C*b^2)*(d*x + c) - (B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)
)/d
```

maple [A] time = 0.48, size = 109, normalized size = 1.56

$$2Bxab+a^2Cx-b^2Cx+\frac{a^2B\ln\left(\sin\left(dx+c\right)\right)}{d}-\frac{b^2B\ln\left(\cos\left(dx+c\right)\right)}{d}+\frac{2Babc}{d}+\frac{b^2C\tan\left(dx+c\right)}{d}-\frac{2Cab\ln\left(\cos\left(dx+\right.\right.}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^2*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] 2*B*x*a*b+a^2*C*x-b^2*C*x+a^2*B*ln(sin(d*x+c))/d-b^2*B*ln(cos(d*x+c))/d+2/d
*B*a*b*c+b^2*C*tan(d*x+c)/d-2/d*C*a*b*ln(cos(d*x+c))+1/d*C*a^2*c-1/d*C*b^2*
c
```

maxima [A] time = 0.98, size = 85, normalized size = 1.21

$$\frac{2\,Ba^2\log\left(\tan\left(dx+c\right)\right)+2\,Cb^2\tan\left(dx+c\right)+2\left(Ca^2+2\,Bab-Cb^2\right)(dx+c)-\left(Ba^2-2\,Cab-Bb^2\right)\log\left(\tan\left(d\right.\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/2*(2*B*a^2*log(tan(d*x + c)) + 2*C*b^2*tan(d*x + c) + 2*(C*a^2 + 2*B*a*b
- C*b^2)*(d*x + c) - (B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1))/d
```

mupad [B] time = 8.85, size = 90, normalized size = 1.29

$$\frac{B\,a^2\,\ln\left(\tan\left(c+d\,x\right)\right)}{d}+\frac{\ln\left(\tan\left(c+d\,x\right)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)\,{\left(b+a\,1\mathrm{i}\right)}^2}{2\,d}+\frac{C\,b^2\,\tan\left(c+d\,x\right)}{d}+\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(B\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (B*a^2*log(tan(c + d*x)))/d + (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)
^2)/(2*d) + (C*b^2*tan(c + d*x))/d + (log(tan(c + d*x) - 1i)*(B + C*1i)*(a*
1i - b)^2)/(2*d)
```

sympy [A] time = 1.61, size = 136, normalized size = 1.94

$$\begin{cases} -\frac{Ba^2\log\left(\tan^2(c+dx)+1\right)}{2d} + \frac{Ba^2\log(\tan(c+dx))}{d} + 2Babx + \frac{Bb^2\log\left(\tan^2(c+dx)+1\right)}{2d} + Ca^2x + \frac{Cab\log\left(\tan^2(c+dx)+1\right)}{d} - Cb^2x + \frac{Cb \\ x(a+b\tan(c))^2\left(B\tan(c)+C\tan^2(c)\right)\cot^2(c) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**2*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((-B*a**2*log(tan(c + d*x)**2 + 1)/(2*d) + B*a**2*log(tan(c + d*x)
)/d + 2*B*a*b*x + B*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*a**2*x + C*a*b*
log(tan(c + d*x)**2 + 1)/d - C*b**2*x + C*b**2*tan(c + d*x)/d, Ne(d, 0)), (
x*(a + b*tan(c))**2*(B*tan(c) + C*tan(c)**2)*cot(c)**2, True))
```

3.13 $\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx$

Optimal. Leaf size=72

$$-x\left(a^2B-2abC-b^2B\right)-\frac{a^2B\cot(c+dx)}{d}+\frac{a(aC+2bB)\log(\sin(c+dx))}{d}-\frac{b^2C\log(\cos(c+dx))}{d}$$

```
[Out] -(B*a^2-B*b^2-2*C*a*b)*x-a^2*B*cot(d*x+c)/d-b^2*C*ln(cos(d*x+c))/d+a*(2*B*b
+C*a)*ln(sin(d*x+c))/d
```

Rubi [A] time = 0.21, antiderivative size = 72, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.100, Rules used = {3632, 3604, 3624, 3475}

$$-x\left(a^2B-2abC-b^2B\right)-\frac{a^2B\cot(c+dx)}{d}+\frac{a(aC+2bB)\log(\sin(c+dx))}{d}-\frac{b^2C\log(\cos(c+dx))}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^3*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] -((a^2*B - b^2*B - 2*a*b*C)*x) - (a^2*B*Cot[c + d*x])/d - (b^2*C*Log[Cos[c
+ d*x]])/d + (a*(2*b*B + a*C)*Log[Sin[c + d*x]])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3604

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> -Simp
[((B*c - A*d)*(b*c - a*d)^2*(c + d*Tan[e + f*x])^(n + 1))/(f*d^2*(n + 1)*(c
^2 + d^2)), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*S
imp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c +
 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)*T
an[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c
- a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3624

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/tan[(e_.) + (f_.)*(x_)], x_Symbol] :> Simp[B*x, x] + (Dist[A, Int[1/Tan[e
 + f*x], x], x] + Dist[C, Int[Tan[e + f*x], x], x]) /; FreeQ[{e, f, A, B, C
```

```
}, x] && NeQ[A, C]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx=\int \cot^2(c+dx)(a+b\tan(c+dx))^2(B$$

$$=-\frac{a^2B\cot(c+dx)}{d}+\int \cot(c+dx)\Big(a(2$$

$$=-\left(a^2B-b^2B-2abC\right)x-\frac{a^2B\cot(c+}{d}$$

$$=-\left(a^2B-b^2B-2abC\right)x-\frac{a^2B\cot(c+}{d}$$

Mathematica [C] time = 0.25, size = 100, normalized size = 1.39

$$\frac{-2a^2B\cot(c+dx)+2a(aC+2bB)\log(\tan(c+dx))+i(a+ib)^2(B+iC)\log(-\tan(c+dx)+i)-(a-ib)^2(C+iB}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^3*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-2*a^2*B*Cot[c + d*x] + I*(a + I*b)^2*(B + I*C)*Log[I - Tan[c + d*x]] + 2*
a*(2*b*B + a*C)*Log[Tan[c + d*x]] - (a - I*b)^2*(I*B + C)*Log[I + Tan[c + d
*x]])/(2*d)
```

fricas [A] time = 0.77, size = 112, normalized size = 1.56

$$-\frac{Cb^2\log\left(\frac{1}{\tan(dx+c)^2+1}\right)\tan(dx+c)+2\left(Ba^2-2Cab-Bb^2\right)dx\tan(dx+c)+2Ba^2-\left(Ca^2+2Bab\right)\log\left(\frac{\tan(dx+}{\tan(dx+c)}\right.}{2d\tan(dx+c)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] -1/2*(C*b^2*log(1/(tan(d*x + c)^2 + 1))*tan(d*x + c) + 2*(B*a^2 - 2*C*a*b -
 B*b^2)*d*x*tan(d*x + c) + 2*B*a^2 - (C*a^2 + 2*B*a*b)*log(tan(d*x + c)^2/(
tan(d*x + c)^2 + 1))*tan(d*x + c))/(d*tan(d*x + c))
```

giac [A] time = 6.39, size = 118, normalized size = 1.64

$$-\frac{2\left(Ba^2-2\,Cab-Bb^2\right)(dx+c)+\left(Ca^2+2\,Bab-Cb^2\right)\log\left(\tan\left(dx+c\right)^2+1\right)-2\left(Ca^2+2\,Bab\right)\log\left(\left|\tan\left(dx+\right.\right.\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] -1/2*(2*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c) + (C*a^2 + 2*B*a*b - C*b^2)*log
(tan(d*x + c)^2 + 1) - 2*(C*a^2 + 2*B*a*b)*log(abs(tan(d*x + c))) + 2*(C*a^
2*tan(d*x + c) + 2*B*a*b*tan(d*x + c) + B*a^2)/tan(d*x + c))/d
```

maple [A] time = 0.52, size = 110, normalized size = 1.53

$$-a^2Bx+Bx\,b^2+2abCx-\frac{a^2B\cot\left(dx+c\right)}{d}+\frac{2Bab\ln\left(\sin\left(dx+c\right)\right)}{d}-\frac{B\,a^2c}{d}+\frac{B\,b^2c}{d}+\frac{a^2C\ln\left(\sin\left(dx+c\right)\right)}{d}-\frac{b^2C\ln\left(\mathrm{co}\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^3*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -a^2*B*x+B*x*b^2+2*a*b*C*x-a^2*B*cot(d*x+c)/d+2/d*B*a*b*ln(sin(d*x+c))-1/d*
B*a^2*c+1/d*B*b^2*c+1/d*a^2*C*ln(sin(d*x+c))-b^2*C*ln(cos(d*x+c))/d+2/d*C*a
*b*c
```

maxima [A] time = 0.74, size = 93, normalized size = 1.29

$$-\frac{2\left(Ba^2-2\,Cab-Bb^2\right)(dx+c)+\left(Ca^2+2\,Bab-Cb^2\right)\log\left(\tan\left(dx+c\right)^2+1\right)-2\left(Ca^2+2\,Bab\right)\log\left(\tan\left(dx+\right.\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] -1/2*(2*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c) + (C*a^2 + 2*B*a*b - C*b^2)*log
(tan(d*x + c)^2 + 1) - 2*(C*a^2 + 2*B*a*b)*log(tan(d*x + c)) + 2*B*a^2/tan(
d*x + c))/d
```

mupad [B] time = 9.00, size = 100, normalized size = 1.39

$$\frac{\ln(\tan(c+dx))\left(C a^2+2 B b a\right)}{d}-\frac{\ln(\tan(c+dx)-\mathrm{i})\ (-C+B\,1\mathrm{i})\ (-b+a\,1\mathrm{i})^2}{2d}+\frac{\ln(\tan(c+dx)+1\mathrm{i})\ (C+B\,1\mathrm{i}}{2d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(tan(c + d*x))*(C*a^2 + 2*B*a*b))/d - (log(tan(c + d*x) - 1i)*(B*1i - C
)*(a*1i - b)^2)/(2*d) + (log(tan(c + d*x) + 1i)*(B*1i + C)*(a*1i + b)^2)/(2
*d) - (B*a^2*cot(c + d*x))/d
```

sympy [A] time = 2.30, size = 158, normalized size = 2.19

$$\begin{cases} \mathrm{NaN} \\ x\,(a+b\tan(c))^2\left(B\tan(c)+C\tan^2(c)\right)\cot^3(c) \\ \mathrm{NaN} \\ -Ba^2x-\frac{Ba^2}{d\tan(c+dx)}-\frac{Bab\log\left(\tan^2(c+dx)+1\right)}{d}+\frac{2Bab\log(\tan(c+dx))}{d}+Bb^2x-\frac{Ca^2\log\left(\tan^2(c+dx)+1\right)}{2d}+\frac{Ca^2\log(\tan(c+dx))}{d}+2C \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**3*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**2*(B*tan(c) + C*ta
n(c)**2)*cot(c)**3, Eq(d, 0)), (nan, Eq(c, -d*x)), (-B*a**2*x - B*a**2/(d*t
an(c + d*x)) - B*a*b*log(tan(c + d*x)**2 + 1)/d + 2*B*a*b*log(tan(c + d*x))
/d + B*b**2*x - C*a**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*a**2*log(tan(c +
d*x))/d + 2*C*a*b*x + C*b**2*log(tan(c + d*x)**2 + 1)/(2*d), True))
```

3.14 $\int \cot^4(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx)+C\tan^2\right.$

Optimal. Leaf size=88

$$-\frac{\left(a^2B-2abC-b^2B\right)\log(\sin(c+dx))}{d}-\frac{a^2B\cot^2(c+dx)}{2d}+x\left(b^2C-a(aC+2bB)\right)-\frac{a(aC+2bB)\cot(c+dx)}{d}$$

```
[Out] (b^2*C-a*(2*B*b+C*a))*x-a*(2*B*b+C*a)*cot(d*x+c)/d-1/2*a^2*B*cot(d*x+c)^2/d
-(B*a^2-B*b^2-2*C*a*b)*ln(sin(d*x+c))/d
```

Rubi [A] time = 0.26, antiderivative size = 88, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3632, 3604, 3628, 3531, 3475}

$$-\frac{\left(a^2B-2abC-b^2B\right)\log(\sin(c+dx))}{d}-\frac{a^2B\cot^2(c+dx)}{2d}+x\left(b^2C-a(aC+2bB)\right)-\frac{a(aC+2bB)\cot(c+dx)}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^4*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (b^2*C - a*(2*b*B + a*C))*x - (a*(2*b*B + a*C)*Cot[c + d*x])/d - (a^2*B*Cot
[c + d*x]^2)/(2*d) - ((a^2*B - b^2*B - 2*a*b*C)*Log[Sin[c + d*x]])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3604

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> -Simp
[((B*c - A*d)*(b*c - a*d)^2*(c + d*Tan[e + f*x])^(n + 1))/(f*d^2*(n + 1)*(c
^2 + d^2)), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*S
imp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c +
```

```
 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)*T
an[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c
- a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int \cot^3(c+dx)(a+b\tan(c+dx))^2(B$$

$$= -\frac{a^2B\cot^2(c+dx)}{2d} + \int \cot^2(c+dx)\,($$

$$= -\frac{a(2bB+aC)\cot(c+dx)}{d} - \frac{a^2B\cot^2(}{2d}$$

$$= \left(b^2C-a(2bB+aC)\right)x - \frac{a(2bB+aC)}{d}$$

$$= \left(b^2C-a(2bB+aC)\right)x - \frac{a(2bB+aC)}{d}$$

Mathematica [C] time = 0.35, size = 123, normalized size = 1.40

$$\frac{-2\left(a^2B-2abC-b^2B\right)\log(\tan(c+dx))-a^2B\cot^2(c+dx)-2a(aC+2bB)\cot(c+dx)+(a-ib)^2(B-iC)\log(\text{ta}}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^4*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-2*a*(2*b*B + a*C)*Cot[c + d*x] - a^2*B*Cot[c + d*x]^2 + (a + I*b)^2*(B +
I*C)*Log[I - Tan[c + d*x]] - 2*(a^2*B - b^2*B - 2*a*b*C)*Log[Tan[c + d*x]]
+ (a - I*b)^2*(B - I*C)*Log[I + Tan[c + d*x]])/(2*d)
```

fricas [A] time = 1.55, size = 122, normalized size = 1.39

$$-\frac{\left(Ba^2 - 2\,Cab - Bb^2\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^2 + Ba^2 + \left(Ba^2 + 2\left(Ca^2 + 2\,Bab - Cb^2\right)dx\right)\tan\left(dx+c\right)^2 +}{2\,d\tan\left(dx+c\right)^2}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] -1/2*((B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*ta
n(d*x + c)^2 + B*a^2 + (B*a^2 + 2*(C*a^2 + 2*B*a*b - C*b^2)*d*x)*tan(d*x +
c)^2 + 2*(C*a^2 + 2*B*a*b)*tan(d*x + c))/(d*tan(d*x + c)^2)
```

giac [B] time = 8.12, size = 237, normalized size = 2.69

$$-\frac{Ba^2\tan\left(\frac{1}{2}\,dx + \frac{1}{2}\,c\right)^2 - 4\,Ca^2\tan\left(\frac{1}{2}\,dx + \frac{1}{2}\,c\right) - 8\,Bab\tan\left(\frac{1}{2}\,dx + \frac{1}{2}\,c\right) + 8\left(Ca^2 + 2\,Bab - Cb^2\right)\left(dx+c\right) - 8\,(B}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] -1/8*(B*a^2*tan(1/2*d*x + 1/2*c)^2 - 4*C*a^2*tan(1/2*d*x + 1/2*c) - 8*B*a*b
*tan(1/2*d*x + 1/2*c) + 8*(C*a^2 + 2*B*a*b - C*b^2)*(d*x + c) - 8*(B*a^2 -
2*C*a*b - B*b^2)*log(tan(1/2*d*x + 1/2*c)^2 + 1) + 8*(B*a^2 - 2*C*a*b - B*b
^2)*log(abs(tan(1/2*d*x + 1/2*c))) - (12*B*a^2*tan(1/2*d*x + 1/2*c)^2 - 24*
C*a*b*tan(1/2*d*x + 1/2*c)^2 - 12*B*b^2*tan(1/2*d*x + 1/2*c)^2 - 4*C*a^2*ta
n(1/2*d*x + 1/2*c) - 8*B*a*b*tan(1/2*d*x + 1/2*c) - B*a^2)/tan(1/2*d*x + 1/
2*c)^2)/d
```

maple [A] time = 0.62, size = 141, normalized size = 1.60

$$-\frac{a^2B\left(\cot^2\left(dx+c\right)\right)}{2d} - \frac{a^2B\ln\left(\sin\left(dx+c\right)\right)}{d} - a^2Cx - \frac{C\cot\left(dx+c\right)a^2}{d} - \frac{C\,a^2c}{d} - 2Bxab - \frac{2B\cot\left(dx+c\right)ab}{d} - \frac{2Babc}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^4*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -1/2*a^2*B*cot(d*x+c)^2/d-a^2*B*ln(sin(d*x+c))/d-a^2*C*x-1/d*C*cot(d*x+c)*a
^2-1/d*C*a^2*c-2*B*x*a*b-2/d*B*cot(d*x+c)*a*b-2/d*B*a*b*c+2/d*C*a*b*ln(sin(
d*x+c))+1/d*b^2*B*ln(sin(d*x+c))+b^2*C*x+1/d*C*b^2*c
```

maxima [A] time = 0.70, size = 120, normalized size = 1.36

$$-\frac{2\left(Ca^2+2\,Bab-Cb^2\right)(dx+c)-\left(Ba^2-2\,Cab-Bb^2\right)\log\left(\tan\left(dx+c\right)^2+1\right)+2\left(Ba^2-2\,Cab-Bb^2\right)\log\left(\tan\left(\right.\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] -1/2*(2*(C*a^2 + 2*B*a*b - C*b^2)*(d*x + c) - (B*a^2 - 2*C*a*b - B*b^2)*log
(tan(d*x + c)^2 + 1) + 2*(B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)) + (B*a
^2 + 2*(C*a^2 + 2*B*a*b)*tan(d*x + c))/tan(d*x + c)^2)/d
```

mupad [B] time = 8.98, size = 127, normalized size = 1.44

$$\frac{\ln\left(\tan\left(c+d\,x\right)\right)\,\left(-B\,a^2+2\,C\,a\,b+B\,b^2\right)}{d}-\frac{\cot\left(c+d\,x\right)^2\,\left(\frac{B\,a^2}{2}+\tan\left(c+d\,x\right)\,\left(C\,a^2+2\,B\,b\,a\right)\right)}{d}-\frac{\ln\left(\tan\left(c+d\,x\right)\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^4*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(tan(c + d*x))*(B*b^2 - B*a^2 + 2*C*a*b))/d - (cot(c + d*x)^2*((B*a^2)/
2 + tan(c + d*x)*(C*a^2 + 2*B*a*b)))/d - (log(tan(c + d*x) + 1i)*(B - C*1i)
*(a*1i + b)^2)/(2*d) - (log(tan(c + d*x) - 1i)*(B + C*1i)*(a*1i - b)^2)/(2*
d)
```

sympy [A] time = 4.31, size = 212, normalized size = 2.41

$$\begin{cases} \text{NaN} \\ x\left(a+b\tan\left(c\right)\right)^2\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\cot^4\left(c\right) \\ \frac{Ba^2\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}-\frac{Ba^2\log\left(\tan\left(c+dx\right)\right)}{d}-\frac{Ba^2}{2d\tan^2\left(c+dx\right)}-2Babx-\frac{2Bab}{d\tan\left(c+dx\right)}-\frac{Bb^2\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+\frac{Bb^2\log\left(\tan\left(c+dx\right)\right)}{d} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**4*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, (Eq(c, 0) | Eq(c, -d*x)) & (Eq(d, 0) | Eq(c, -d*x))), (x*(a
 + b*tan(c))**2*(B*tan(c) + C*tan(c)**2)*cot(c)**4, Eq(d, 0)), (B*a**2*log(
tan(c + d*x)**2 + 1)/(2*d) - B*a**2*log(tan(c + d*x))/d - B*a**2/(2*d*tan(c
 + d*x)**2) - 2*B*a*b*x - 2*B*a*b/(d*tan(c + d*x)) - B*b**2*log(tan(c + d*x
)**2 + 1)/(2*d) + B*b**2*log(tan(c + d*x))/d - C*a**2*x - C*a**2/(d*tan(c +
 d*x)) - C*a*b*log(tan(c + d*x)**2 + 1)/d + 2*C*a*b*log(tan(c + d*x))/d + C
*b**2*x, True))
```

3.15 $\int \cot^5(c+dx)(a+b\tan(c+dx))^2 \left(B\tan(c+dx) + C\tan^2(\right.$

Optimal. Leaf size=118

$$\frac{\left(a^2B-2abC-b^2B\right)\cot(c+dx)}{d}+x\left(a^2B-2abC-b^2B\right)-\frac{a^2B\cot^3(c+dx)}{3d}+\frac{\left(b^2C-a(aC+2bB)\right)\log(\sin(c+dx)}{d}$$

```
[Out] (B*a^2-B*b^2-2*C*a*b)*x+(B*a^2-B*b^2-2*C*a*b)*cot(d*x+c)/d-1/2*a*(2*B*b+C*a
)*cot(d*x+c)^2/d-1/3*a^2*B*cot(d*x+c)^3/d+(b^2*C-a*(2*B*b+C*a))*ln(sin(d*x+
c))/d
```

Rubi [A] time = 0.31, antiderivative size = 118, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3632, 3604, 3628, 3529, 3531, 3475}

$$\frac{\left(a^2B-2abC-b^2B\right)\cot(c+dx)}{d}+x\left(a^2B-2abC-b^2B\right)-\frac{a^2B\cot^3(c+dx)}{3d}+\frac{\left(b^2C-a(aC+2bB)\right)\log(\sin(c+dx)}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^5*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (a^2*B - b^2*B - 2*a*b*C)*x + ((a^2*B - b^2*B - 2*a*b*C)*Cot[c + d*x])/d -
(a*(2*b*B + a*C)*Cot[c + d*x]^2)/(2*d) - (a^2*B*Cot[c + d*x]^3)/(3*d) + ((b
^2*C - a*(2*b*B + a*C))*Log[Sin[c + d*x]])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
```

```
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3604

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> -Simp
[((B*c - A*d)*(b*c - a*d)^2*(c + d*Tan[e + f*x])^(n + 1))/(f*d^2*(n + 1)*(c
^2 + d^2)), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*S
imp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c +
 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)*T
an[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c
- a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int \cot^4(c+dx)(a+b\tan(c+dx))^2(B$$
$$= -\frac{a^2B\cot^3(c+dx)}{3d} + \int \cot^3(c+dx)\left(a\right.$$
$$= -\frac{a(2bB+aC)\cot^2(c+dx)}{2d} - \frac{a^2B\cot^3}{3a}$$
$$= \frac{\left(a^2B-b^2B-2abC\right)\cot(c+dx)}{d} - \frac{a(2}{}$$
$$= \left(a^2B-b^2B-2abC\right)x + \frac{\left(a^2B-b^2B-\right.}{}$$
$$= \left(a^2B-b^2B-2abC\right)x + \frac{\left(a^2B-b^2B-\right.}{}$$

Mathematica [C] time = 1.17, size = 152, normalized size = 1.29

$$\frac{6\left(a^2B-2abC-b^2B\right)\cot(c+dx)-6\left(a^2C+2abB-b^2C\right)\log(\tan(c+dx))-2a^2B\cot^3(c+dx)-3a(aC+2bB)c}{6d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^5*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (6*(a^2*B - b^2*B - 2*a*b*C)*Cot[c + d*x] - 3*a*(2*b*B + a*C)*Cot[c + d*x]^
2 - 2*a^2*B*Cot[c + d*x]^3 + 3*(a + I*b)^2*((-I)*B + C)*Log[I - Tan[c + d*x
]] - 6*(2*a*b*B + a^2*C - b^2*C)*Log[Tan[c + d*x]] + 3*(a - I*b)^2*(I*B + C
)*Log[I + Tan[c + d*x]])/(6*d)
```

fricas [A] time = 0.62, size = 157, normalized size = 1.33

$$-\frac{3\left(Ca^2+2\,Bab-Cb^2\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^3+3\left(Ca^2+2\,Bab-2\left(Ba^2-2\,Cab-Bb^2\right)dx\right)\tan\left(dx+c\right)}{6\,d\tan\left(dx+c\right)^3}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] -1/6*(3*(C*a^2 + 2*B*a*b - C*b^2)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*
tan(d*x + c)^3 + 3*(C*a^2 + 2*B*a*b - 2*(B*a^2 - 2*C*a*b - B*b^2)*d*x)*tan(
```

```
d*x + c)^3 + 2*B*a^2 - 6*(B*a^2 - 2*C*a*b - B*b^2)*tan(d*x + c)^2 + 3*(C*a^
2 + 2*B*a*b)*tan(d*x + c))/(d*tan(d*x + c)^3)
```

giac [B] time = 11.09, size = 334, normalized size = 2.83

$$\frac{Ba^2\tan\left(\frac{1}{2}dx+\frac{1}{2}c\right)^3-3Ca^2\tan\left(\frac{1}{2}dx+\frac{1}{2}c\right)^2-6Bab\tan\left(\frac{1}{2}dx+\frac{1}{2}c\right)^2-15Ba^2\tan\left(\frac{1}{2}dx+\frac{1}{2}c\right)+24Cab\tan}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/24*(B*a^2*tan(1/2*d*x + 1/2*c)^3 - 3*C*a^2*tan(1/2*d*x + 1/2*c)^2 - 6*B*a
*b*tan(1/2*d*x + 1/2*c)^2 - 15*B*a^2*tan(1/2*d*x + 1/2*c) + 24*C*a*b*tan(1/
2*d*x + 1/2*c) + 12*B*b^2*tan(1/2*d*x + 1/2*c) + 24*(B*a^2 - 2*C*a*b - B*b^
2)*(d*x + c) + 24*(C*a^2 + 2*B*a*b - C*b^2)*log(tan(1/2*d*x + 1/2*c)^2 + 1)
 - 24*(C*a^2 + 2*B*a*b - C*b^2)*log(abs(tan(1/2*d*x + 1/2*c))) + (44*C*a^2*
tan(1/2*d*x + 1/2*c)^3 + 88*B*a*b*tan(1/2*d*x + 1/2*c)^3 - 44*C*b^2*tan(1/2
*d*x + 1/2*c)^3 + 15*B*a^2*tan(1/2*d*x + 1/2*c)^2 - 24*C*a*b*tan(1/2*d*x +
1/2*c)^2 - 12*B*b^2*tan(1/2*d*x + 1/2*c)^2 - 3*C*a^2*tan(1/2*d*x + 1/2*c) -
 6*B*a*b*tan(1/2*d*x + 1/2*c) - B*a^2)/tan(1/2*d*x + 1/2*c)^3)/d
```

maple [A] time = 0.46, size = 188, normalized size = 1.59

$$-\frac{a^2B\left(\cot^3(dx+c)\right)}{3d}+\frac{a^2B\cot(dx+c)}{d}+a^2Bx+\frac{Ba^2c}{d}-\frac{a^2C\left(\cot^2(dx+c)\right)}{2d}-\frac{a^2C\ln(\sin(dx+c))}{d}-\frac{Bab\left(\cot^2(dx\right.}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^5*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -1/3*a^2*B*cot(d*x+c)^3/d+a^2*B*cot(d*x+c)/d+a^2*B*x+1/d*B*a^2*c-1/2/d*a^2*
C*cot(d*x+c)^2-1/d*a^2*C*ln(sin(d*x+c))-1/d*B*a*b*cot(d*x+c)^2-2/d*B*a*b*ln
(sin(d*x+c))-2*a*b*C*x-2/d*C*cot(d*x+c)*a*b-2/d*C*a*b*c-B*x*b^2-1/d*B*cot(d
*x+c)*b^2-1/d*B*b^2*c+1/d*b^2*C*ln(sin(d*x+c))
```

maxima [A] time = 0.75, size = 149, normalized size = 1.26

$$\frac{6\left(Ba^2-2Cab-Bb^2\right)(dx+c)+3\left(Ca^2+2Bab-Cb^2\right)\log\left(\tan(dx+c)^2+1\right)-6\left(Ca^2+2Bab-Cb^2\right)\log(\tan}{6d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/6*(6*(B*a^2 - 2*C*a*b - B*b^2)*(d*x + c) + 3*(C*a^2 + 2*B*a*b - C*b^2)*lo
g(tan(d*x + c)^2 + 1) - 6*(C*a^2 + 2*B*a*b - C*b^2)*log(tan(d*x + c)) - (2*
B*a^2 - 6*(B*a^2 - 2*C*a*b - B*b^2)*tan(d*x + c)^2 + 3*(C*a^2 + 2*B*a*b)*ta
n(d*x + c))/tan(d*x + c)^3)/d
```

mupad [B] time = 9.08, size = 156, normalized size = 1.32

$$-\frac{\cot(c+d\,x)^3\,\left(\frac{B\,a^2}{3}+\tan(c+d\,x)^2\,\left(-B\,a^2+2\,C\,a\,b+B\,b^2\right)+\tan(c+d\,x)\,\left(\frac{C\,a^2}{2}+B\,b\,a\right)\right)}{d}-\frac{\ln(\tan(c+d\,x))\,\left(C\right.}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^5*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B*1i - C)*(a*1i - b)^2)/(2*d) - (log(tan(c + d*x))
*(C*a^2 - C*b^2 + 2*B*a*b))/d - (cot(c + d*x)^3*((B*a^2)/3 + tan(c + d*x)^2
*(B*b^2 - B*a^2 + 2*C*a*b) + tan(c + d*x)*((C*a^2)/2 + B*a*b)))/d - (log(ta
n(c + d*x) + 1i)*(B*1i + C)*(a*1i + b)^2)/(2*d)
```

sympy [A] time = 5.68, size = 258, normalized size = 2.19

$$\begin{cases} \text{NaN} \\ x\,(a+b\tan(c))^2\left(B\tan(c)+C\tan^2(c)\right)\cot^5(c) \\ Ba^2x+\frac{Ba^2}{d\tan(c+dx)}-\frac{Ba^2}{3d\tan^3(c+dx)}+\frac{Bab\log\left(\tan^2(c+dx)+1\right)}{d}-\frac{2Bab\log(\tan(c+dx))}{d}-\frac{Bab}{d\tan^2(c+dx)}-Bb^2x-\frac{Bb^2}{d\tan(c+dx)}+\frac{Ca^2\log}{} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**5*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, (Eq(c, 0) | Eq(c, -d*x)) & (Eq(d, 0) | Eq(c, -d*x))), (x*(a
 + b*tan(c))**2*(B*tan(c) + C*tan(c)**2)*cot(c)**5, Eq(d, 0)), (B*a**2*x +
B*a**2/(d*tan(c + d*x)) - B*a**2/(3*d*tan(c + d*x)**3) + B*a*b*log(tan(c +
d*x)**2 + 1)/d - 2*B*a*b*log(tan(c + d*x))/d - B*a*b/(d*tan(c + d*x)**2) -
B*b**2*x - B*b**2/(d*tan(c + d*x)) + C*a**2*log(tan(c + d*x)**2 + 1)/(2*d)
- C*a**2*log(tan(c + d*x))/d - C*a**2/(2*d*tan(c + d*x)**2) - 2*C*a*b*x - 2
*C*a*b/(d*tan(c + d*x)) - C*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + C*b**2*lo
g(tan(c + d*x))/d, True))
```

3.16 $\int \cot^6(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2($

Optimal. Leaf size=151

$$\frac{\left(a^2B-2abC-b^2B\right)\cot^2(c+dx)}{2d}+\frac{\left(a^2B-2abC-b^2B\right)\log(\sin(c+dx))}{d}+x\left(a^2C+2abB-b^2C\right)-\frac{a^2B\cot^4(c+d}{4d}$$

```
[Out] (2*B*a*b+C*a^2-C*b^2)*x-(b^2*C-a*(2*B*b+C*a))*cot(d*x+c)/d+1/2*(B*a^2-B*b^2
-2*C*a*b)*cot(d*x+c)^2/d-1/3*a*(2*B*b+C*a)*cot(d*x+c)^3/d-1/4*a^2*B*cot(d*x
+c)^4/d+(B*a^2-B*b^2-2*C*a*b)*ln(sin(d*x+c))/d
```

Rubi [A] time = 0.37, antiderivative size = 151, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3632, 3604, 3628, 3529, 3531, 3475}

$$\frac{\left(a^2B-2abC-b^2B\right)\cot^2(c+dx)}{2d}+\frac{\left(a^2B-2abC-b^2B\right)\log(\sin(c+dx))}{d}+x\left(a^2C+2abB-b^2C\right)-\frac{a^2B\cot^4(c+d}{4d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^6*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (2*a*b*B + a^2*C - b^2*C)*x - ((b^2*C - a*(2*b*B + a*C))*Cot[c + d*x])/d +
((a^2*B - b^2*B - 2*a*b*C)*Cot[c + d*x]^2)/(2*d) - (a*(2*b*B + a*C)*Cot[c +
 d*x]^3)/(3*d) - (a^2*B*Cot[c + d*x]^4)/(4*d) + ((a^2*B - b^2*B - 2*a*b*C)*
Log[Sin[c + d*x]])/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
```

```
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3604

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> -Simp
[((B*c - A*d)*(b*c - a*d)^2*(c + d*Tan[e + f*x])^(n + 1))/(f*d^2*(n + 1)*(c
^2 + d^2)), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*S
imp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c +
 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)*T
an[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c
- a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^2\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int \cot^5(c+dx)(a+b\tan(c+dx))^2(B$$

$$= -\frac{a^2B\cot^4(c+dx)}{4d} + \int \cot^4(c+dx)\,($$

$$= -\frac{a(2bB+aC)\cot^3(c+dx)}{3d} - \frac{a^2B\cot}{}$$

$$= \frac{\left(a^2B-b^2B-2abC\right)\cot^2(c+dx)}{2d} - \frac{a}{}$$

$$= -\frac{\left(b^2C-a(2bB+aC)\right)\cot(c+dx)}{d} +$$

$$= \left(2abB+a^2C-b^2C\right)x - \frac{\left(b^2C-a(2b\right.}{}$$

$$= \left(2abB+a^2C-b^2C\right)x - \frac{\left(b^2C-a(2b\right.}{}$$

Mathematica [C] time = 2.93, size = 180, normalized size = 1.19

$$\frac{6\left(a^2B-2abC-b^2B\right)\cot^2(c+dx)+12\left(a^2C+2abB-b^2C\right)\cot(c+dx)-6\left(\left(-2a^2B+4abC+2b^2B\right)\log(\tan(c\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^6*(a + b*Tan[c + d*x])^2*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (12*(2*a*b*B + a^2*C - b^2*C)*Cot[c + d*x] + 6*(a^2*B - b^2*B - 2*a*b*C)*Co
t[c + d*x]^2 - 4*a*(2*b*B + a*C)*Cot[c + d*x]^3 - 3*a^2*B*Cot[c + d*x]^4 -
6*((a + I*b)^2*(B + I*C)*Log[I - Tan[c + d*x]] + (-2*a^2*B + 2*b^2*B + 4*a*
b*C)*Log[Tan[c + d*x]] + (a - I*b)^2*(B - I*C)*Log[I + Tan[c + d*x]]))/(12*
d)
```

fricas [A] time = 1.24, size = 191, normalized size = 1.26

$$\frac{6\left(Ba^2-2\,Cab-Bb^2\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^4+3\left(3\,Ba^2-4\,Cab-2\,Bb^2+4\left(Ca^2+2\,Bab-Cb^2\right)dx\right)\tan}{1}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/12*(6*(B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*
tan(d*x + c)^4 + 3*(3*B*a^2 - 4*C*a*b - 2*B*b^2 + 4*(C*a^2 + 2*B*a*b - C*b^
2)*d*x)*tan(d*x + c)^4 + 12*(C*a^2 + 2*B*a*b - C*b^2)*tan(d*x + c)^3 - 3*B*
a^2 + 6*(B*a^2 - 2*C*a*b - B*b^2)*tan(d*x + c)^2 - 4*(C*a^2 + 2*B*a*b)*tan(
d*x + c))/(d*tan(d*x + c)^4)
```

giac [B] time = 21.20, size = 435, normalized size = 2.88

$$-\frac{3\,Ba^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^4-8\,Ca^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-16\,Bab\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-36\,Ba^2\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2+48\,Cab}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] -1/192*(3*B*a^2*tan(1/2*d*x + 1/2*c)^4 - 8*C*a^2*tan(1/2*d*x + 1/2*c)^3 - 1
6*B*a*b*tan(1/2*d*x + 1/2*c)^3 - 36*B*a^2*tan(1/2*d*x + 1/2*c)^2 + 48*C*a*b
*tan(1/2*d*x + 1/2*c)^2 + 24*B*b^2*tan(1/2*d*x + 1/2*c)^2 + 120*C*a^2*tan(1
/2*d*x + 1/2*c) + 240*B*a*b*tan(1/2*d*x + 1/2*c) - 96*C*b^2*tan(1/2*d*x + 1
/2*c) - 192*(C*a^2 + 2*B*a*b - C*b^2)*(d*x + c) + 192*(B*a^2 - 2*C*a*b - B*
b^2)*log(tan(1/2*d*x + 1/2*c)^2 + 1) - 192*(B*a^2 - 2*C*a*b - B*b^2)*log(ab
s(tan(1/2*d*x + 1/2*c))) + (400*B*a^2*tan(1/2*d*x + 1/2*c)^4 - 800*C*a*b*ta
n(1/2*d*x + 1/2*c)^4 - 400*B*b^2*tan(1/2*d*x + 1/2*c)^4 - 120*C*a^2*tan(1/2
*d*x + 1/2*c)^3 - 240*B*a*b*tan(1/2*d*x + 1/2*c)^3 + 96*C*b^2*tan(1/2*d*x +
 1/2*c)^3 - 36*B*a^2*tan(1/2*d*x + 1/2*c)^2 + 48*C*a*b*tan(1/2*d*x + 1/2*c)
^2 + 24*B*b^2*tan(1/2*d*x + 1/2*c)^2 + 8*C*a^2*tan(1/2*d*x + 1/2*c) + 16*B*
a*b*tan(1/2*d*x + 1/2*c) + 3*B*a^2)/tan(1/2*d*x + 1/2*c)^4)/d
```

maple [A] time = 0.56, size = 238, normalized size = 1.58

$$-\frac{a^2B\left(\cot^4\left(dx+c\right)\right)}{4d}+\frac{a^2B\left(\cot^2\left(dx+c\right)\right)}{2d}+\frac{a^2B\ln\left(\sin\left(dx+c\right)\right)}{d}-\frac{a^2C\left(\cot^3\left(dx+c\right)\right)}{3d}+\frac{C\cot\left(dx+c\right)a^2}{d}+a^2Cx+$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^6*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -1/4*a^2*B*cot(d*x+c)^4/d+1/2*a^2*B*cot(d*x+c)^2/d+a^2*B*ln(sin(d*x+c))/d-1
/3/d*a^2*C*cot(d*x+c)^3+1/d*C*cot(d*x+c)*a^2+a^2*C*x+1/d*C*a^2*c-2/3/d*B*a*
b*cot(d*x+c)^3+2/d*B*cot(d*x+c)*a*b+2*B*x*a*b+2/d*B*a*b*c-1/d*C*a*b*cot(d*x
+c)^2-2/d*C*a*b*ln(sin(d*x+c))-1/2/d*b^2*B*cot(d*x+c)^2-1/d*b^2*B*ln(sin(d*
x+c))-b^2*C*x-1/d*C*cot(d*x+c)*b^2-1/d*C*b^2*c
```

maxima [A] time = 0.86, size = 175, normalized size = 1.16

$$\frac{12\left(Ca^2+2\,Bab-Cb^2\right)(dx+c)-6\left(Ba^2-2\,Cab-Bb^2\right)\log\left(\tan\left(dx+c\right)^2+1\right)+12\left(Ba^2-2\,Cab-Bb^2\right)\log\left(\mathrm{t}\right.}{12\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^2*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/12*(12*(C*a^2 + 2*B*a*b - C*b^2)*(d*x + c) - 6*(B*a^2 - 2*C*a*b - B*b^2)*
log(tan(d*x + c)^2 + 1) + 12*(B*a^2 - 2*C*a*b - B*b^2)*log(tan(d*x + c)) +
(12*(C*a^2 + 2*B*a*b - C*b^2)*tan(d*x + c)^3 - 3*B*a^2 + 6*(B*a^2 - 2*C*a*b
 - B*b^2)*tan(d*x + c)^2 - 4*(C*a^2 + 2*B*a*b)*tan(d*x + c))/tan(d*x + c)^4
)/d
```

mupad [B] time = 8.86, size = 182, normalized size = 1.21

$$-\frac{\cot\left(c+d\,x\right)^4\,\left(\frac{B\,a^2}{4}+\tan\left(c+d\,x\right)^2\,\left(-\frac{B\,a^2}{2}+C\,a\,b+\frac{B\,b^2}{2}\right)-\tan\left(c+d\,x\right)^3\,\left(C\,a^2+2\,B\,a\,b-C\,b^2\right)+\tan\left(c+d\,x\right.}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^6*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)^2)/(2*d) - (log(tan(c + d*x))
*(B*b^2 - B*a^2 + 2*C*a*b))/d - (cot(c + d*x)^4*((B*a^2)/4 + tan(c + d*x)^2
*((B*b^2)/2 - (B*a^2)/2 + C*a*b) - tan(c + d*x)^3*(C*a^2 - C*b^2 + 2*B*a*b)
 + tan(c + d*x)*((C*a^2)/3 + (2*B*a*b)/3)))/d + (log(tan(c + d*x) - 1i)*(B
+ C*1i)*(a*1i - b)^2)/(2*d)
```

sympy [A] time = 8.80, size = 311, normalized size = 2.06

$$\begin{cases} \mathrm{NaN} \\ x\left(a+b\tan\left(c\right)\right)^2\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\cot^6\left(c\right) \\ -\frac{Ba^2\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+\frac{Ba^2\log\left(\tan\left(c+dx\right)\right)}{d}+\frac{Ba^2}{2d\tan^2\left(c+dx\right)}-\frac{Ba^2}{4d\tan^4\left(c+dx\right)}+2Babx+\frac{2Bab}{d\tan\left(c+dx\right)}-\frac{2Bab}{3d\tan^3\left(c+dx\right)}+\frac{Bb^2\log\ldots}{} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**6*(a+b*tan(d*x+c))**2*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, (Eq(c, 0) | Eq(c, -d*x)) & (Eq(d, 0) | Eq(c, -d*x))), (x*(a
 + b*tan(c))**2*(B*tan(c) + C*tan(c)**2)*cot(c)**6, Eq(d, 0)), (-B*a**2*log
(tan(c + d*x)**2 + 1)/(2*d) + B*a**2*log(tan(c + d*x))/d + B*a**2/(2*d*tan(
c + d*x)**2) - B*a**2/(4*d*tan(c + d*x)**4) + 2*B*a*b*x + 2*B*a*b/(d*tan(c
+ d*x)) - 2*B*a*b/(3*d*tan(c + d*x)**3) + B*b**2*log(tan(c + d*x)**2 + 1)/(
2*d) - B*b**2*log(tan(c + d*x))/d - B*b**2/(2*d*tan(c + d*x)**2) + C*a**2*x
 + C*a**2/(d*tan(c + d*x)) - C*a**2/(3*d*tan(c + d*x)**3) + C*a*b*log(tan(c
 + d*x)**2 + 1)/d - 2*C*a*b*log(tan(c + d*x))/d - C*a*b/(d*tan(c + d*x)**2)
 - C*b**2*x - C*b**2/(d*tan(c + d*x)), True))
```

3.17 $\int (a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx$

Optimal. Leaf size=165

$$\frac{b\left(a^2B-2abC-b^2B\right)\tan(c+dx)}{d}-\frac{\left(a^3B-3a^2bC-3ab^2B+b^3C\right)\log(\cos(c+dx))}{d}-x\left(a^3C+3a^2bB-3ab^2C-\right.$$

```
[Out] -(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*x-(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*ln(c
os(d*x+c))/d+b*(B*a^2-B*b^2-2*C*a*b)*tan(d*x+c)/d+1/2*(B*a-C*b)*(a+b*tan(d*
x+c))^2/d+1/3*B*(a+b*tan(d*x+c))^3/d+1/4*C*(a+b*tan(d*x+c))^4/b/d
```

Rubi [A] time = 0.18, antiderivative size = 165, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, integrand size = 32, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3630, 3528, 3525, 3475}

$$\frac{b\left(a^2B-2abC-b^2B\right)\tan(c+dx)}{d}-\frac{\left(-3a^2bC+a^3B-3ab^2B+b^3C\right)\log(\cos(c+dx))}{d}-x\left(3a^2bB+a^3C-3ab^2C\right.$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] -((3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*x) - ((a^3*B - 3*a*b^2*B - 3*a^2*
b*C + b^3*C)*Log[Cos[c + d*x]])/d + (b*(a^2*B - b^2*B - 2*a*b*C)*Tan[c + d*
x])/d + ((a*B - b*C)*(a + b*Tan[c + d*x])^2)/(2*d) + (B*(a + b*Tan[c + d*x]
)^3)/(3*d) + (C*(a + b*Tan[c + d*x])^4)/(4*b*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
```

```
0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int (a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx=\frac{C(a+b\tan(c+dx))^4}{4bd}+\int (a+b\tan(c+dx))^3(-C$$
$$=\frac{B(a+b\tan(c+dx))^3}{3d}+\frac{C(a+b\tan(c+dx))^4}{4bd}+\int$$
$$=\frac{(aB-bC)(a+b\tan(c+dx))^2}{2d}+\frac{B(a+b\tan(c+dx}{3d}$$
$$=-\left(3a^2bB-b^3B+a^3C-3ab^2C\right)x+\frac{b\left(a^2B-b^2B-\right.}{}$$
$$=-\left(3a^2bB-b^3B+a^3C-3ab^2C\right)x-\frac{\left(a^3B-3ab^2B\right.}{}$$

Mathematica [C] time = 1.63, size = 209, normalized size = 1.27

$$\frac{-12b^2B\left(b^2-6a^2\right)\tan(c+dx)+24ab^3B\tan^2(c+dx)-6(aB+bC)\left(6ab^2\tan(c+dx)+(-b+ia)^3\log(-\tan(c+\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2),x]
```

```
[Out] ((-6*I)*(a + I*b)^4*B*Log[I - Tan[c + d*x]] + (6*I)*(a - I*b)^4*B*Log[I + T
an[c + d*x]] - 12*b^2*(-6*a^2 + b^2)*B*Tan[c + d*x] + 24*a*b^3*B*Tan[c + d*
x]^2 + 4*b^4*B*Tan[c + d*x]^3 + 3*C*(a + b*Tan[c + d*x])^4 - 6*(a*B + b*C)*
((I*a - b)^3*Log[I - Tan[c + d*x]] - (I*a + b)^3*Log[I + Tan[c + d*x]] + 6*
a*b^2*Tan[c + d*x] + b^3*Tan[c + d*x]^2))/(12*b*d)
```

fricas [A] time = 0.59, size = 178, normalized size = 1.08

$$\frac{3\,Cb^3\tan\left(dx+c\right)^4+4\left(3\,Cab^2+Bb^3\right)\tan\left(dx+c\right)^3-12\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)dx+6\left(3\,Ca^2b+3\,Bab^2\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="fr
icas")
```

```
[Out] 1/12*(3*C*b^3*tan(d*x + c)^4 + 4*(3*C*a*b^2 + B*b^3)*tan(d*x + c)^3 - 12*(C
*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*d*x + 6*(3*C*a^2*b + 3*B*a*b^2 - C*b^
3)*tan(d*x + c)^2 - 6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*log(1/(tan(d*
x + c)^2 + 1)) + 12*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*tan(d*x + c))/d
```

giac [B] time = 12.65, size = 2870, normalized size = 17.39

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="gi
ac")
```

```
[Out] -1/12*(12*C*a^3*d*x*tan(d*x)^4*tan(c)^4 + 36*B*a^2*b*d*x*tan(d*x)^4*tan(c)^
4 - 36*C*a*b^2*d*x*tan(d*x)^4*tan(c)^4 - 12*B*b^3*d*x*tan(d*x)^4*tan(c)^4 +
 6*B*a^3*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(
c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^4*tan(c
)^4 - 18*C*a^2*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x
)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)
^4*tan(c)^4 - 18*B*a*b^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) +
 tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*
tan(d*x)^4*tan(c)^4 + 6*C*b^3*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan
(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 +
 1))*tan(d*x)^4*tan(c)^4 - 48*C*a^3*d*x*tan(d*x)^3*tan(c)^3 - 144*B*a^2*b*d
*x*tan(d*x)^3*tan(c)^3 + 144*C*a*b^2*d*x*tan(d*x)^3*tan(c)^3 + 48*B*b^3*d*x
*tan(d*x)^3*tan(c)^3 - 18*C*a^2*b*tan(d*x)^4*tan(c)^4 - 18*B*a*b^2*tan(d*x)
^4*tan(c)^4 + 9*C*b^3*tan(d*x)^4*tan(c)^4 - 24*B*a^3*log(4*(tan(d*x)^4*tan(
c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*
tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 + 72*C*a^2*b*log(4*(tan(d*x
)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*t
an(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 + 72*B*a*b^2*log(4*
(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)
^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 - 24*C*b^3*
log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + ta
n(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^3*tan(c)^3 + 12*
C*a^3*tan(d*x)^4*tan(c)^3 + 36*B*a^2*b*tan(d*x)^4*tan(c)^3 - 36*C*a*b^2*tan
(d*x)^4*tan(c)^3 - 12*B*b^3*tan(d*x)^4*tan(c)^3 + 12*C*a^3*tan(d*x)^3*tan(c
)^4 + 36*B*a^2*b*tan(d*x)^3*tan(c)^4 - 36*C*a*b^2*tan(d*x)^3*tan(c)^4 - 12*
B*b^3*tan(d*x)^3*tan(c)^4 + 72*C*a^3*d*x*tan(d*x)^2*tan(c)^2 + 216*B*a^2*b*
d*x*tan(d*x)^2*tan(c)^2 - 216*C*a*b^2*d*x*tan(d*x)^2*tan(c)^2 - 72*B*b^3*d*
```

x*tan(d*x)^2*tan(c)^2 - 18*C*a^2*b*tan(d*x)^4*tan(c)^2 - 18*B*a*b^2*tan(d*x
)^4*tan(c)^2 + 6*C*b^3*tan(d*x)^4*tan(c)^2 + 36*C*a^2*b*tan(d*x)^3*tan(c)^3
 + 36*B*a*b^2*tan(d*x)^3*tan(c)^3 - 24*C*b^3*tan(d*x)^3*tan(c)^3 - 18*C*a^2
*b*tan(d*x)^2*tan(c)^4 - 18*B*a*b^2*tan(d*x)^2*tan(c)^4 + 6*C*b^3*tan(d*x)^
2*tan(c)^4 + 12*C*a*b^2*tan(d*x)^4*tan(c) + 4*B*b^3*tan(d*x)^4*tan(c) + 36*
B*a^3*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^
2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2*tan(c)^2
 - 108*C*a^2*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^
2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*tan(d*x)^2
*tan(c)^2 - 108*B*a*b^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) +
tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*t
an(d*x)^2*tan(c)^2 + 36*C*b^3*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan
(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 +
 1))*tan(d*x)^2*tan(c)^2 - 36*C*a^3*tan(d*x)^3*tan(c)^2 - 108*B*a^2*b*tan(d
*x)^3*tan(c)^2 + 144*C*a*b^2*tan(d*x)^3*tan(c)^2 + 48*B*b^3*tan(d*x)^3*tan(
c)^2 - 36*C*a^3*tan(d*x)^2*tan(c)^3 - 108*B*a^2*b*tan(d*x)^2*tan(c)^3 + 144
*C*a*b^2*tan(d*x)^2*tan(c)^3 + 48*B*b^3*tan(d*x)^2*tan(c)^3 + 12*C*a*b^2*ta
n(d*x)*tan(c)^4 + 4*B*b^3*tan(d*x)*tan(c)^4 - 3*C*b^3*tan(d*x)^4 - 48*C*a^3
*d*x*tan(d*x)*tan(c) - 144*B*a^2*b*d*x*tan(d*x)*tan(c) + 144*C*a*b^2*d*x*ta
n(d*x)*tan(c) + 48*B*b^3*d*x*tan(d*x)*tan(c) + 36*C*a^2*b*tan(d*x)^3*tan(c)
 + 36*B*a*b^2*tan(d*x)^3*tan(c) - 24*C*b^3*tan(d*x)^3*tan(c) - 36*C*a^2*b*t
an(d*x)^2*tan(c)^2 - 36*B*a*b^2*tan(d*x)^2*tan(c)^2 + 12*C*b^3*tan(d*x)^2*t
an(c)^2 + 36*C*a^2*b*tan(d*x)*tan(c)^3 + 36*B*a*b^2*tan(d*x)*tan(c)^3 - 24*
C*b^3*tan(d*x)*tan(c)^3 - 3*C*b^3*tan(c)^4 - 12*C*a*b^2*tan(d*x)^3 - 4*B*b^
3*tan(d*x)^3 - 24*B*a^3*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) +
tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1))*t
an(d*x)*tan(c) + 72*C*a^2*b*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c
) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1
))*tan(d*x)*tan(c) + 72*B*a*b^2*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*t
an(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2
 + 1))*tan(d*x)*tan(c) - 24*C*b^3*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3
*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)
^2 + 1))*tan(d*x)*tan(c) + 36*C*a^3*tan(d*x)^2*tan(c) + 108*B*a^2*b*tan(d*x
)^2*tan(c) - 144*C*a*b^2*tan(d*x)^2*tan(c) - 48*B*b^3*tan(d*x)^2*tan(c) + 3
6*C*a^3*tan(d*x)*tan(c)^2 + 108*B*a^2*b*tan(d*x)*tan(c)^2 - 144*C*a*b^2*tan
(d*x)*tan(c)^2 - 48*B*b^3*tan(d*x)*tan(c)^2 - 12*C*a*b^2*tan(c)^3 - 4*B*b^3
*tan(c)^3 + 12*C*a^3*d*x + 36*B*a^2*b*d*x - 36*C*a*b^2*d*x - 12*B*b^3*d*x -
 18*C*a^2*b*tan(d*x)^2 - 18*B*a*b^2*tan(d*x)^2 + 6*C*b^3*tan(d*x)^2 + 36*C*
a^2*b*tan(d*x)*tan(c) + 36*B*a*b^2*tan(d*x)*tan(c) - 24*C*b^3*tan(d*x)*tan(
c) - 18*C*a^2*b*tan(c)^2 - 18*B*a*b^2*tan(c)^2 + 6*C*b^3*tan(c)^2 + 6*B*a^3
log(4(tan(d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + t
an(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1)) - 18*C*a^2*b*log(4*(tan(
d*x)^4*tan(c)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 -
2*tan(d*x)*tan(c) + 1)/(tan(c)^2 + 1)) - 18*B*a*b^2*log(4*(tan(d*x)^4*tan(c
)^2 - 2*tan(d*x)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*t

```
an(c) + 1)/(tan(c)^2 + 1)) + 6*C*b^3*log(4*(tan(d*x)^4*tan(c)^2 - 2*tan(d*x
)^3*tan(c) + tan(d*x)^2*tan(c)^2 + tan(d*x)^2 - 2*tan(d*x)*tan(c) + 1)/(tan
(c)^2 + 1)) - 12*C*a^3*tan(d*x) - 36*B*a^2*b*tan(d*x) + 36*C*a*b^2*tan(d*x)
 + 12*B*b^3*tan(d*x) - 12*C*a^3*tan(c) - 36*B*a^2*b*tan(c) + 36*C*a*b^2*tan
(c) + 12*B*b^3*tan(c) - 18*C*a^2*b - 18*B*a*b^2 + 9*C*b^3)/(d*tan(d*x)^4*ta
n(c)^4 - 4*d*tan(d*x)^3*tan(c)^3 + 6*d*tan(d*x)^2*tan(c)^2 - 4*d*tan(d*x)*t
an(c) + d)
```

maple [A] time = 0.03, size = 314, normalized size = 1.90

$$\frac{b^3C\left(\tan^4(dx+c)\right)}{4d}+\frac{B\left(\tan^3(dx+c)\right)b^3}{3d}+\frac{C\left(\tan^3(dx+c)\right)a\,b^2}{d}+\frac{3B\left(\tan^2(dx+c)\right)a\,b^2}{2d}+\frac{3C\left(\tan^2(dx+c)\right)a^2}{2d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] 1/4/d*b^3*C*tan(d*x+c)^4+1/3/d*B*tan(d*x+c)^3*b^3+1/d*C*tan(d*x+c)^3*a*b^2+
3/2/d*B*tan(d*x+c)^2*a*b^2+3/2/d*C*tan(d*x+c)^2*a^2*b-1/2/d*b^3*C*tan(d*x+c
)^2+3/d*B*tan(d*x+c)*a^2*b-1/d*B*tan(d*x+c)*b^3+1/d*C*tan(d*x+c)*a^3-3/d*C*
a*b^2*tan(d*x+c)+1/2/d*ln(1+tan(d*x+c)^2)*a^3*B-3/2/d*ln(1+tan(d*x+c)^2)*B*
a*b^2-3/2/d*ln(1+tan(d*x+c)^2)*C*a^2*b+1/2/d*ln(1+tan(d*x+c)^2)*b^3*C-3/d*B
*arctan(tan(d*x+c))*a^2*b+1/d*B*arctan(tan(d*x+c))*b^3-1/d*C*arctan(tan(d*x
+c))*a^3+3/d*C*arctan(tan(d*x+c))*a*b^2
```

maxima [A] time = 0.52, size = 179, normalized size = 1.08

$$\frac{3\,Cb^3\tan(dx+c)^4+4\left(3\,Cab^2+Bb^3\right)\tan(dx+c)^3+6\left(3\,Ca^2b+3\,Bab^2-Cb^3\right)\tan(dx+c)^2-12\left(Ca^3+3\,Ba\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, algorithm="ma
xima")
```

```
[Out] 1/12*(3*C*b^3*tan(d*x + c)^4 + 4*(3*C*a*b^2 + B*b^3)*tan(d*x + c)^3 + 6*(3*
C*a^2*b + 3*B*a*b^2 - C*b^3)*tan(d*x + c)^2 - 12*(C*a^3 + 3*B*a^2*b - 3*C*a
*b^2 - B*b^3)*(d*x + c) + 6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*log(tan
(d*x + c)^2 + 1) + 12*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*tan(d*x + c))
/d
```

mupad [B] time = 8.83, size = 181, normalized size = 1.10

$$x\,\left(-C\,a^3-3\,B\,a^2\,b+3\,C\,a\,b^2+B\,b^3\right)-\frac{\tan\left(c+d\,x\right)^2\,\left(\frac{C\,b^3}{2}-\frac{3\,a\,b\,\left(B\,b+C\,a\right)}{2}\right)}{d}-\frac{\tan\left(c+d\,x\right)\,\left(-C\,a^3-3\,B\,a^2\,b+3\,C\right.}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^3,x)
```

```
[Out] x*(B*b^3 - C*a^3 - 3*B*a^2*b + 3*C*a*b^2) - (tan(c + d*x)^2*((C*b^3)/2 - (3
*a*b*(B*b + C*a))/2))/d - (tan(c + d*x)*(B*b^3 - C*a^3 - 3*B*a^2*b + 3*C*a*
b^2))/d + (log(tan(c + d*x)^2 + 1)*((B*a^3)/2 + (C*b^3)/2 - (3*B*a*b^2)/2 -
 (3*C*a^2*b)/2))/d + (tan(c + d*x)^3*((B*b^3)/3 + C*a*b^2))/d + (C*b^3*tan(
c + d*x)^4)/(4*d)
```

sympy [A] time = 0.67, size = 313, normalized size = 1.90

$$\begin{cases} \frac{Ba^3 \log\left(\tan^2(c+dx)+1\right)}{2d} - 3Ba^2bx + \frac{3Ba^2b\tan(c+dx)}{d} - \frac{3Bab^2 \log\left(\tan^2(c+dx)+1\right)}{2d} + \frac{3Bab^2\tan^2(c+dx)}{2d} + Bb^3x + \frac{Bb^3\tan^3(c+dx)}{3d} - \frac{B \\ x\left(a+b\tan(c)\right)^3\left(B\tan(c)+C\tan^2(c)\right) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a**3*log(tan(c + d*x)**2 + 1)/(2*d) - 3*B*a**2*b*x + 3*B*a**2*
b*tan(c + d*x)/d - 3*B*a*b**2*log(tan(c + d*x)**2 + 1)/(2*d) + 3*B*a*b**2*t
an(c + d*x)**2/(2*d) + B*b**3*x + B*b**3*tan(c + d*x)**3/(3*d) - B*b**3*tan
(c + d*x)/d - C*a**3*x + C*a**3*tan(c + d*x)/d - 3*C*a**2*b*log(tan(c + d*x
)**2 + 1)/(2*d) + 3*C*a**2*b*tan(c + d*x)**2/(2*d) + 3*C*a*b**2*x + C*a*b**
2*tan(c + d*x)**3/d - 3*C*a*b**2*tan(c + d*x)/d + C*b**3*log(tan(c + d*x)**
2 + 1)/(2*d) + C*b**3*tan(c + d*x)**4/(4*d) - C*b**3*tan(c + d*x)**2/(2*d),
 Ne(d, 0)), (x*(a + b*tan(c))**3*(B*tan(c) + C*tan(c)**2), True))
```

3.18 $\int \cot(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(\right.$

Optimal. Leaf size=140

$$\frac{b\left(a^2C+2abB-b^2C\right)\tan(c+dx)}{d}-\frac{\left(a^3C+3a^2bB-3ab^2C-b^3B\right)\log(\cos(c+dx))}{d}+x\left(a^3B-3a^2bC-3ab^2B+\right.$$

```
[Out] (B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*x-(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*ln(co
s(d*x+c))/d+b*(2*B*a*b+C*a^2-C*b^2)*tan(d*x+c)/d+1/2*(B*b+C*a)*(a+b*tan(d*x
+c))^2/d+1/3*C*(a+b*tan(d*x+c))^3/d
```

Rubi [A] time = 0.21, antiderivative size = 140, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3632, 3528, 3525, 3475}

$$\frac{b\left(a^2C+2abB-b^2C\right)\tan(c+dx)}{d}-\frac{\left(3a^2bB+a^3C-3ab^2C-b^3B\right)\log(\cos(c+dx))}{d}+x\left(-3a^2bC+a^3B-3ab^2B\right.$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2)
,x]
```

```
[Out] (a^3*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*x - ((3*a^2*b*B - b^3*B + a^3*C - 3
*a*b^2*C)*Log[Cos[c + d*x]])/d + (b*(2*a*b*B + a^2*C - b^2*C)*Tan[c + d*x])
/d + ((b*B + a*C)*(a + b*Tan[c + d*x])^2)/(2*d) + (C*(a + b*Tan[c + d*x])^3
)/(3*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
```

```
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \cot(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \int (a+b\tan(c+dx))^3(B+C\tan(c+dx$$

$$= \frac{C(a+b\tan(c+dx))^3}{3d} + \int (a+b\tan(c$$

$$= \frac{(bB+aC)(a+b\tan(c+dx))^2}{2d} + \frac{C(a+$$

$$= \left(a^3B-3ab^2B-3a^2bC+b^3C\right)x + \frac{b\left(2a\right.}{}$$

$$= \left(a^3B-3ab^2B-3a^2bC+b^3C\right)x - \frac{\left(3a^2\right.}{}$$

Mathematica [C] time = 1.07, size = 130, normalized size = 0.93

$$\frac{6b\left(3a^2C+3abB-b^2C\right)\tan(c+dx)+3b^2(3aC+bB)\tan^2(c+dx)+3(a-ib)^3(C+iB)\log(\tan(c+dx)+i)+3(}{6d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d
*x]^2),x]
```

```
[Out] (3*(a + I*b)^3*((-I)*B + C)*Log[I - Tan[c + d*x]] + 3*(a - I*b)^3*(I*B + C)
*Log[I + Tan[c + d*x]] + 6*b*(3*a*b*B + 3*a^2*C - b^2*C)*Tan[c + d*x] + 3*b
^2*(b*B + 3*a*C)*Tan[c + d*x]^2 + 2*b^3*C*Tan[c + d*x]^3)/(6*d)
```

fricas [A] time = 0.68, size = 142, normalized size = 1.01

$$\frac{2\,Cb^3\tan(dx+c)^3+6\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)dx+3\left(3\,Cab^2+Bb^3\right)\tan(dx+c)^2-3\left(Ca^3+3\,Ba^2b-3\right.}{6\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="fricas")
```

```
[Out] 1/6*(2*C*b^3*tan(d*x + c)^3 + 6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*d*x
 + 3*(3*C*a*b^2 + B*b^3)*tan(d*x + c)^2 - 3*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2
- B*b^3)*log(1/(tan(d*x + c)^2 + 1)) + 6*(3*C*a^2*b + 3*B*a*b^2 - C*b^3)*ta
n(d*x + c))/d
```

giac [A] time = 5.96, size = 158, normalized size = 1.13

$$\frac{2\,Cb^3\tan\left(dx+c\right)^3+9\,Cab^2\tan\left(dx+c\right)^2+3\,Bb^3\tan\left(dx+c\right)^2+18\,Ca^2b\tan\left(dx+c\right)+18\,Bab^2\tan\left(dx+c\right)-$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="giac")
```

```
[Out] 1/6*(2*C*b^3*tan(d*x + c)^3 + 9*C*a*b^2*tan(d*x + c)^2 + 3*B*b^3*tan(d*x +
c)^2 + 18*C*a^2*b*tan(d*x + c) + 18*B*a*b^2*tan(d*x + c) - 6*C*b^3*tan(d*x
+ c) + 6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*(d*x + c) + 3*(C*a^3 + 3*B
*a^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2 + 1))/d
```

maple [A] time = 0.47, size = 234, normalized size = 1.67

$$a^3Bx+\frac{a^3Bc}{d}-\frac{C\,a^3\ln\left(\cos\left(dx+c\right)\right)}{d}-\frac{3a^2bB\ln\left(\cos\left(dx+c\right)\right)}{d}-3Cx\,a^2b+\frac{3C\tan\left(dx+c\right)a^2b}{d}-\frac{3C\,a^2bc}{d}-3Bxa\,b^2+$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] a^3*B*x+1/d*a^3*B*c-1/d*C*a^3*ln(cos(d*x+c))-3/d*a^2*b*B*ln(cos(d*x+c))-3*C
*x*a^2*b+3/d*C*tan(d*x+c)*a^2*b-3/d*C*a^2*b*c-3*B*x*a*b^2+3/d*B*tan(d*x+c)*
a*b^2-3/d*B*a*b^2*c+3/2/d*C*a*b^2*tan(d*x+c)^2+3/d*C*a*b^2*ln(cos(d*x+c))+1
/2/d*b^3*B*tan(d*x+c)^2+b^3*B*ln(cos(d*x+c))/d+1/3/d*b^3*C*tan(d*x+c)^3-1/d
*b^3*C*tan(d*x+c)+b^3*C*x+1/d*b^3*C*c
```

maxima [A] time = 1.02, size = 143, normalized size = 1.02

$$\frac{2\,Cb^3\tan\left(dx+c\right)^3+3\left(3\,Cab^2+Bb^3\right)\tan\left(dx+c\right)^2+6\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)\left(dx+c\right)+3\left(Ca^3+3\,Ba\right.}{6\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x, al
gorithm="maxima")
```

```
[Out] 1/6*(2*C*b^3*tan(d*x + c)^3 + 3*(3*C*a*b^2 + B*b^3)*tan(d*x + c)^2 + 6*(B*a
^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*(d*x + c) + 3*(C*a^3 + 3*B*a^2*b - 3*C*
a*b^2 - B*b^3)*log(tan(d*x + c)^2 + 1) + 6*(3*C*a^2*b + 3*B*a*b^2 - C*b^3)*
tan(d*x + c))/d
```

mupad [B] time = 8.96, size = 142, normalized size = 1.01

$$x\left(B\,a^3-3\,C\,a^2\,b-3\,B\,a\,b^2+C\,b^3\right)-\frac{\ln\left(\tan\left(c+d\,x\right)^2+1\right)\,\left(-\frac{C\,a^3}{2}-\frac{3\,B\,a^2\,b}{2}+\frac{3\,C\,a\,b^2}{2}+\frac{B\,b^3}{2}\right)}{d}+\frac{\tan\left(c+d\,x\right)^2\,\left(\frac{B\,b^3}{2}\right.}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))^3
,x)
```

```
[Out] x*(B*a^3 + C*b^3 - 3*B*a*b^2 - 3*C*a^2*b) - (log(tan(c + d*x)^2 + 1)*((B*b^
3)/2 - (C*a^3)/2 - (3*B*a^2*b)/2 + (3*C*a*b^2)/2))/d + (tan(c + d*x)^2*((B*
b^3)/2 + (3*C*a*b^2)/2))/d - (tan(c + d*x)*(C*b^3 - 3*a*b*(B*b + C*a)))/d +
 (C*b^3*tan(c + d*x)^3)/(3*d)
```

sympy [A] time = 1.82, size = 248, normalized size = 1.77

$$\begin{cases} Ba^3x+\frac{3Ba^2b\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}-3Bab^2x+\frac{3Bab^2\tan\left(c+dx\right)}{d}-\frac{Bb^3\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+\frac{Bb^3\tan^2\left(c+dx\right)}{2d}+\frac{Ca^3\log\left(\tan^2\left(c+dx\right)+1\right.}{2d} \\ x\left(a+b\tan\left(c\right)\right)^3\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\cot\left(c\right) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),x)
```

```
[Out] Piecewise((B*a**3*x + 3*B*a**2*b*log(tan(c + d*x)**2 + 1)/(2*d) - 3*B*a*b**
2*x + 3*B*a*b**2*tan(c + d*x)/d - B*b**3*log(tan(c + d*x)**2 + 1)/(2*d) + B
*b**3*tan(c + d*x)**2/(2*d) + C*a**3*log(tan(c + d*x)**2 + 1)/(2*d) - 3*C*a
**2*b*x + 3*C*a**2*b*tan(c + d*x)/d - 3*C*a*b**2*log(tan(c + d*x)**2 + 1)/(
2*d) + 3*C*a*b**2*tan(c + d*x)**2/(2*d) + C*b**3*x + C*b**3*tan(c + d*x)**3
/(3*d) - C*b**3*tan(c + d*x)/d, Ne(d, 0)), (x*(a + b*tan(c))**3*(B*tan(c) +
 C*tan(c)**2)*cot(c), True))
```

3.19 $\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(\right.$

Optimal. Leaf size=117

$$\frac{a^3B\log(\sin(c+dx))}{d}-\frac{b\left(3a^2C+3abB-b^2C\right)\log(\cos(c+dx))}{d}+x\left(a^3C+3a^2bB-3ab^2C-b^3B\right)+\frac{b^2(2aC+bB)}{d}$$

```
[Out] (3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*x-b*(3*B*a*b+3*C*a^2-C*b^2)*ln(cos(d*x+c)
)/d+a^3*B*ln(sin(d*x+c))/d+b^2*(B*b+2*C*a)*tan(d*x+c)/d+1/2*b*C*(a+b*tan(d*
x+c))^2/d
```

Rubi [A] time = 0.34, antiderivative size = 117, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3632, 3607, 3637, 3624, 3475}

$$-\frac{b\left(3a^2C+3abB-b^2C\right)\log(\cos(c+dx))}{d}+x\left(3a^2bB+a^3C-3ab^2C-b^3B\right)+\frac{a^3B\log(\sin(c+dx))}{d}+\frac{b^2(2aC+b}{}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^2*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*x - (b*(3*a*b*B + 3*a^2*C - b^2*C)*
Log[Cos[c + d*x]])/d + (a^3*B*Log[Sin[c + d*x]])/d + (b^2*(b*B + 2*a*C)*Tan
[c + d*x])/d + (b*C*(a + b*Tan[c + d*x])^2)/(2*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3607

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*B*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(m
+ n)), x] + Dist[1/(d*(m + n)), Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan
[e + f*x])^n*Simp[a^2*A*d*(m + n) - b*B*(b*c*(m - 1) + a*d*(n + 1)) + d*(m
+ n)*(2*a*A*b + B*(a^2 - b^2))*Tan[e + f*x] - (b*B*(b*c - a*d)*(m - 1) - b*
(A*b + a*B)*d*(m + n))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e,
f, A, B, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2,
 0] && GtQ[m, 1] && (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(IGtQ[n, 1] &
& ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3624

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/tan[(e_.) + (f_.)*(x_)], x_Symbol] :> Simp[B*x, x] + (Dist[A, Int[1/Tan[e
 + f*x], x], x] + Dist[C, Int[Tan[e + f*x], x], x]) /; FreeQ[{e, f, A, B, C
}, x] && NeQ[A, C]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int \cot^2(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int \cot(c+dx)(a+b\tan(c+dx))^3(B+$$

$$= \frac{bC(a+b\tan(c+dx))^2}{2d} + \frac{1}{2}\int \cot(c+$$

$$= \frac{b^2(bB+2aC)\tan(c+dx)}{d} + \frac{bC(a+b\,t}{}$$

$$= \left(3a^2bB-b^3B+a^3C-3ab^2C\right)x + \frac{b^2(l}{}$$

$$= \left(3a^2bB-b^3B+a^3C-3ab^2C\right)x - \frac{b\left(3\right.}{}$$

Mathematica [C] time = 0.47, size = 113, normalized size = 0.97

$$\frac{2a^3B\log(\tan(c+dx))+2b^2(3aC+bB)\tan(c+dx)-(a+ib)^3(B+iC)\log(-\tan(c+dx)+i)-(a-ib)^3(B-iC}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^2*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-((a + I*b)^3*(B + I*C)*Log[I - Tan[c + d*x]]) + 2*a^3*B*Log[Tan[c + d*x]]
 - (a - I*b)^3*(B - I*C)*Log[I + Tan[c + d*x]] + 2*b^2*(b*B + 3*a*C)*Tan[c
+ d*x] + b^3*C*Tan[c + d*x]^2)/(2*d)
```

fricas [A] time = 0.65, size = 133, normalized size = 1.14

$$\frac{Cb^3\tan(dx+c)^2+Ba^3\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)+2\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)dx-\left(3\,Ca^2b+3\,Bab^2-Cb^3\right)\log\left(\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/2*(C*b^3*tan(d*x + c)^2 + B*a^3*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))
+ 2*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*d*x - (3*C*a^2*b + 3*B*a*b^2 -
C*b^3)*log(1/(tan(d*x + c)^2 + 1)) + 2*(3*C*a*b^2 + B*b^3)*tan(d*x + c))/d
```

giac [A] time = 8.84, size = 129, normalized size = 1.10

$$\frac{Cb^3\tan(dx+c)^2+2\,Ba^3\log(|\tan(dx+c)|)+6\,Cab^2\tan(dx+c)+2\,Bb^3\tan(dx+c)+2\left(Ca^3+3\,Ba^2b-3\,C\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/2*(C*b^3*tan(d*x + c)^2 + 2*B*a^3*log(abs(tan(d*x + c))) + 6*C*a*b^2*tan(
d*x + c) + 2*B*b^3*tan(d*x + c) + 2*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)
*(d*x + c) - (B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)^2 + 1
))/d
```

maple [A] time = 0.57, size = 183, normalized size = 1.56

$$\frac{a^3B\ln(\sin(dx+c))}{d}+a^3Cx+\frac{C\,a^3c}{d}+3Bx\,a^2b+\frac{3B\,a^2bc}{d}-\frac{3C\,a^2b\ln(\cos(dx+c))}{d}-\frac{3Ba\,b^2\ln(\cos(dx+c))}{d}-3a\,b^2($$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^2*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] a^3*B*ln(sin(d*x+c))/d+a^3*C*x+1/d*C*a^3*c+3*B*x*a^2*b+3/d*B*a^2*b*c-3/d*C*
a^2*b*ln(cos(d*x+c))-3/d*B*a*b^2*ln(cos(d*x+c))-3*a*b^2*C*x+3/d*C*a*b^2*tan
(d*x+c)-3/d*C*a*b^2*c-B*x*b^3+1/d*B*tan(d*x+c)*b^3-1/d*B*b^3*c+1/2/d*b^3*C*
tan(d*x+c)^2+b^3*C*ln(cos(d*x+c))/d
```

maxima [A] time = 0.60, size = 124, normalized size = 1.06

$$\frac{Cb^3 \tan(dx+c)^2 + 2\,Ba^3 \log(\tan(dx+c)) + 2\left(Ca^3 + 3\,Ba^2b - 3\,Cab^2 - Bb^3\right)(dx+c) - \left(Ba^3 - 3\,Ca^2b - 3\,Bab^2\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/2*(C*b^3*tan(d*x + c)^2 + 2*B*a^3*log(tan(d*x + c)) + 2*(C*a^3 + 3*B*a^2*
b - 3*C*a*b^2 - B*b^3)*(d*x + c) - (B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*
log(tan(d*x + c)^2 + 1) + 2*(3*C*a*b^2 + B*b^3)*tan(d*x + c))/d
```

mupad [B] time = 8.96, size = 118, normalized size = 1.01

$$\frac{\tan(c+d\,x)\,\left(B\,b^3+3\,C\,a\,b^2\right)}{d}+\frac{B\,a^3\,\ln(\tan(c+d\,x))}{d}+\frac{C\,b^3\,\tan(c+d\,x)^2}{2\,d}-\frac{\ln(\tan(c+d\,x)+1\mathrm{i})\,(B-C\,1\mathrm{i})\,(b+}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (tan(c + d*x)*(B*b^3 + 3*C*a*b^2))/d + (B*a^3*log(tan(c + d*x)))/d - (log(t
an(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)^3*1i)/(2*d) - (log(tan(c + d*x) - 1
i)*(B + C*1i)*(a*1i - b)^3*1i)/(2*d) + (C*b^3*tan(c + d*x)^2)/(2*d)
```

sympy [A] time = 2.32, size = 211, normalized size = 1.80

$$\begin{cases} -\frac{Ba^3\log\left(\tan^2(c+dx)+1\right)}{2d} + \frac{Ba^3\log(\tan(c+dx))}{d} + 3Ba^2bx + \frac{3Bab^2\log\left(\tan^2(c+dx)+1\right)}{2d} - Bb^3x + \frac{Bb^3\tan(c+dx)}{d} + Ca^3x + \frac{3Ca^2b\log}{} \\ x\,(a+b\tan(c))^3\left(B\tan(c)+C\tan^2(c)\right)\cot^2(c) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**2*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((-B*a**3*log(tan(c + d*x)**2 + 1)/(2*d) + B*a**3*log(tan(c + d*x)
)/d + 3*B*a**2*b*x + 3*B*a*b**2*log(tan(c + d*x)**2 + 1)/(2*d) - B*b**3*x +
 B*b**3*tan(c + d*x)/d + C*a**3*x + 3*C*a**2*b*log(tan(c + d*x)**2 + 1)/(2*
d) - 3*C*a*b**2*x + 3*C*a*b**2*tan(c + d*x)/d - C*b**3*log(tan(c + d*x)**2
+ 1)/(2*d) + C*b**3*tan(c + d*x)**2/(2*d), Ne(d, 0)), (x*(a + b*tan(c))**3*
(B*tan(c) + C*tan(c)**2)*cot(c)**2, True))
```

3.20 $\int \cot^3(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(\right.$

Optimal. Leaf size=119

$$\frac{a^2(aC+3bB)\log(\sin(c+dx))}{d}-x\left(a^3B-3a^2bC-3ab^2B+b^3C\right)+\frac{b^2(aB+bC)\tan(c+dx)}{d}-\frac{b^2(3aC+bB)\log(\cos}{d}$$

```
[Out] -(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*x-b^2*(B*b+3*C*a)*ln(cos(d*x+c))/d+a^2*(
3*B*b+C*a)*ln(sin(d*x+c))/d+b^2*(B*a+C*b)*tan(d*x+c)/d-a*B*cot(d*x+c)*(a+b*
tan(d*x+c))^2/d
```

Rubi [A] time = 0.33, antiderivative size = 119, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3632, 3605, 3637, 3624, 3475}

$$-x\left(-3a^2bC+a^3B-3ab^2B+b^3C\right)+\frac{a^2(aC+3bB)\log(\sin(c+dx))}{d}+\frac{b^2(aB+bC)\tan(c+dx)}{d}-\frac{b^2(3aC+bB)\log}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^3*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] -((a^3*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*x) - (b^2*(b*B + 3*a*C)*Log[Cos[c
 + d*x]])/d + (a^2*(3*b*B + a*C)*Log[Sin[c + d*x]])/d + (b^2*(a*B + b*C)*Ta
n[c + d*x])/d - (a*B*Cot[c + d*x]*(a + b*Tan[c + d*x])^2)/d
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3605

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[((b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x
])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3624

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/tan[(e_.) + (f_.)*(x_)], x_Symbol] :> Simp[B*x, x] + (Dist[A, Int[1/Tan[e
 + f*x], x], x] + Dist[C, Int[Tan[e + f*x], x], x]) /; FreeQ[{e, f, A, B, C
}, x] && NeQ[A, C]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int \cot^3(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int \cot^2(c+dx)(a+b\tan(c+dx))^3(B$$

$$= -\frac{aB\cot(c+dx)(a+b\tan(c+dx))^2}{d}$$

$$= \frac{b^2(aB+bC)\tan(c+dx)}{d} - \frac{aB\cot(c$$

$$= -\left(a^3B-3ab^2B-3a^2bC+b^3C\right)x +$$

$$= -\left(a^3B-3ab^2B-3a^2bC+b^3C\right)x -$$

Mathematica [C] time = 0.49, size = 113, normalized size = 0.95

$$\frac{-2a^3B\cot(c+dx)+2a^2(aC+3bB)\log(\tan(c+dx))+i(a+ib)^3(B+iC)\log(-\tan(c+dx)+i)+(b+ia)^3(B-i}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^3*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-2*a^3*B*Cot[c + d*x] + I*(a + I*b)^3*(B + I*C)*Log[I - Tan[c + d*x]] + 2*
a^2*(3*b*B + a*C)*Log[Tan[c + d*x]] + (I*a + b)^3*(B - I*C)*Log[I + Tan[c +
 d*x]] + 2*b^3*C*Tan[c + d*x])/(2*d)
```

fricas [A] time = 0.65, size = 145, normalized size = 1.22

$$\frac{2\,Cb^3\tan(dx+c)^2-2\,Ba^3-2\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)dx\tan(dx+c)+\left(Ca^3+3\,Ba^2b\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)}{2\,d\tan(dx+c)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/2*(2*C*b^3*tan(d*x + c)^2 - 2*B*a^3 - 2*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 +
C*b^3)*d*x*tan(d*x + c) + (C*a^3 + 3*B*a^2*b)*log(tan(d*x + c)^2/(tan(d*x +
 c)^2 + 1))*tan(d*x + c) - (3*C*a*b^2 + B*b^3)*log(1/(tan(d*x + c)^2 + 1))*
tan(d*x + c))/(d*tan(d*x + c))
```

giac [A] time = 12.04, size = 152, normalized size = 1.28

$$\frac{2\,Cb^3\tan(dx+c)-2\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)(dx+c)-\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)\log\left(\tan(dx+c)^2\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/2*(2*C*b^3*tan(d*x + c) - 2*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*(d*x
+ c) - (C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2 + 1) + 2*
(C*a^3 + 3*B*a^2*b)*log(abs(tan(d*x + c))) - 2*(C*a^3*tan(d*x + c) + 3*B*a^
2*b*tan(d*x + c) + B*a^3)/tan(d*x + c))/d
```

maple [A] time = 0.45, size = 168, normalized size = 1.41

$$-a^3Bx+3Bxa\,b^2+3Cx\,a^2b-b^3Cx-\frac{B\cot(dx+c)\,a^3}{d}+\frac{3a^2bB\ln(\sin(dx+c))}{d}-\frac{b^3B\ln(\cos(dx+c))}{d}-\frac{a^3Bc}{d}+\frac{3Ba\,b^2}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^3*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -a^3*B*x+3*B*x*a*b^2+3*C*x*a^2*b-b^3*C*x-1/d*B*cot(d*x+c)*a^3+3/d*a^2*b*B*l
n(sin(d*x+c))-b^3*B*ln(cos(d*x+c))/d-1/d*a^3*B*c+3/d*B*a*b^2*c+1/d*b^3*C*ta
n(d*x+c)+1/d*C*a^3*ln(sin(d*x+c))-3/d*C*a*b^2*ln(cos(d*x+c))+3/d*C*a^2*b*c-
1/d*b^3*C*c
```

maxima [A] time = 0.78, size = 125, normalized size = 1.05

$$\frac{2\,Cb^3\tan\left(dx+c\right)-\frac{2\,Ba^3}{\tan(dx+c)}-2\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)(dx+c)-\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)\log\left(t\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/2*(2*C*b^3*tan(d*x + c) - 2*B*a^3/tan(d*x + c) - 2*(B*a^3 - 3*C*a^2*b - 3
*B*a*b^2 + C*b^3)*(d*x + c) - (C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(t
an(d*x + c)^2 + 1) + 2*(C*a^3 + 3*B*a^2*b)*log(tan(d*x + c)))/d
```

mupad [B] time = 8.86, size = 114, normalized size = 0.96

$$\frac{\ln\left(\tan\left(c+d\,x\right)\right)\,\left(C\,a^3+3\,B\,b\,a^2\right)}{d}-\frac{B\,a^3\cot\left(c+d\,x\right)}{d}+\frac{C\,b^3\tan\left(c+d\,x\right)}{d}+\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)\,\left(a+b\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(tan(c + d*x))*(C*a^3 + 3*B*a^2*b))/d + (log(tan(c + d*x) - 1i)*(B + C*
1i)*(a + b*1i)^3*1i)/(2*d) - (log(tan(c + d*x) + 1i)*(B - C*1i)*(a - b*1i)^
3*1i)/(2*d) - (B*a^3*cot(c + d*x))/d + (C*b^3*tan(c + d*x))/d
```

sympy [A] time = 4.42, size = 214, normalized size = 1.80

$$\begin{cases} \mathrm{NaN} \\ x\left(a+b\tan\left(c\right)\right)^3\left(B\tan\left(c\right)+C\tan^2\left(c\right)\right)\cot^3\left(c\right) \\ \mathrm{NaN} \\ -Ba^3x-\frac{Ba^3}{d\tan\left(c+dx\right)}-\frac{3Ba^2b\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}+\frac{3Ba^2b\log\left(\tan\left(c+dx\right)\right)}{d}+3Bab^2x+\frac{Bb^3\log\left(\tan^2\left(c+dx\right)+1\right)}{2d}-\frac{Ca^3\log\left(\tan^2\left(c+\right.\right.}{2d} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**3*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, Eq(c, 0) & Eq(d, 0)), (x*(a + b*tan(c))**3*(B*tan(c) + C*ta
n(c)**2)*cot(c)**3, Eq(d, 0)), (nan, Eq(c, -d*x)), (-B*a**3*x - B*a**3/(d*t
an(c + d*x)) - 3*B*a**2*b*log(tan(c + d*x)**2 + 1)/(2*d) + 3*B*a**2*b*log(t
an(c + d*x))/d + 3*B*a*b**2*x + B*b**3*log(tan(c + d*x)**2 + 1)/(2*d) - C*a
**3*log(tan(c + d*x)**2 + 1)/(2*d) + C*a**3*log(tan(c + d*x))/d + 3*C*a**2*
b*x + 3*C*a*b**2*log(tan(c + d*x)**2 + 1)/(2*d) - C*b**3*x + C*b**3*tan(c +
 d*x)/d, True))
```

3.21 $\int \cot^4(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2\right.$

Optimal. Leaf size=127

$$-\frac{a\left(a^2B-3abC-3b^2B\right)\log(\sin(c+dx))}{d}-\frac{a^2(aC+2bB)\cot(c+dx)}{d}-x\left(a^3C+3a^2bB-3ab^2C-b^3B\right)-\frac{aB\cot^2}{}$$

```
[Out] -(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*x-a^2*(2*B*b+C*a)*cot(d*x+c)/d-b^3*C*ln(
cos(d*x+c))/d-a*(B*a^2-3*B*b^2-3*C*a*b)*ln(sin(d*x+c))/d-1/2*a*B*cot(d*x+c)
^2*(a+b*tan(d*x+c))^2/d
```

Rubi [A] time = 0.35, antiderivative size = 127, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3632, 3605, 3635, 3624, 3475}

$$-\frac{a\left(a^2B-3abC-3b^2B\right)\log(\sin(c+dx))}{d}-x\left(3a^2bB+a^3C-3ab^2C-b^3B\right)-\frac{a^2(aC+2bB)\cot(c+dx)}{d}-\frac{aB\cot^2}{}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^4*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] -((3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*x) - (a^2*(2*b*B + a*C)*Cot[c + d
*x])/d - (b^3*C*Log[Cos[c + d*x]])/d - (a*(a^2*B - 3*b^2*B - 3*a*b*C)*Log[S
in[c + d*x]])/d - (a*B*Cot[c + d*x]^2*(a + b*Tan[c + d*x])^2)/(2*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3605

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[((b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x
])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3624

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/tan[(e_.) + (f_.)*(x_)], x_Symbol] :> Simp[B*x, x] + (Dist[A, Int[1/Tan[e
 + f*x], x], x] + Dist[C, Int[Tan[e + f*x], x], x]) /; FreeQ[{e, f, A, B, C
}, x] && NeQ[A, C]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \cot^4(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx=\int \cot^3(c+dx)(a+b\tan(c+dx))^3(B$$

$$=-\frac{aB\cot^2(c+dx)(a+b\tan(c+dx))^2}{2d}$$

$$=-\frac{a^2(2bB+aC)\cot(c+dx)}{d}-\frac{aB\cot^2(}{}$$

$$=-\left(3a^2bB-b^3B+a^3C-3ab^2C\right)x-\frac{a^2}{}$$

$$=-\left(3a^2bB-b^3B+a^3C-3ab^2C\right)x-\frac{a^2}{}$$

Mathematica [C] time = 0.46, size = 126, normalized size = 0.99

$$\frac{a^3(-B)\cot^2(c+dx)-2a\left(a^2B-3abC-3b^2B\right)\log(\tan(c+dx))-2a^2(aC+3bB)\cot(c+dx)+(a+ib)^3(B+iC}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^4*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-2*a^2*(3*b*B + a*C)*Cot[c + d*x] - a^3*B*Cot[c + d*x]^2 + (a + I*b)^3*(B
+ I*C)*Log[I - Tan[c + d*x]] - 2*a*(a^2*B - 3*b^2*B - 3*a*b*C)*Log[Tan[c +
d*x]] + (a - I*b)^3*(B - I*C)*Log[I + Tan[c + d*x]])/(2*d)
```

fricas [A] time = 0.92, size = 162, normalized size = 1.28

$$-\frac{Cb^3\log\left(\frac{1}{\tan(dx+c)^2+1}\right)\tan(dx+c)^2+Ba^3+\left(Ba^3-3Ca^2b-3Bab^2\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan(dx+c)^2+\left(Ba^3+2\right.}{2d\tan(dx+c)^2}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] -1/2*(C*b^3*log(1/(tan(d*x + c)^2 + 1))*tan(d*x + c)^2 + B*a^3 + (B*a^3 - 3
*C*a^2*b - 3*B*a*b^2)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*tan(d*x + c)
^2 + (B*a^3 + 2*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*d*x)*tan(d*x + c)^2
 + 2*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/(d*tan(d*x + c)^2)
```

giac [A] time = 22.95, size = 193, normalized size = 1.52

$$-\frac{2\left(Ca^3+3Ba^2b-3Cab^2-Bb^3\right)(dx+c)-\left(Ba^3-3Ca^2b-3Bab^2+Cb^3\right)\log\left(\tan(dx+c)^2+1\right)+2\left(Ba^3-3\right.}{2}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] -1/2*(2*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*(d*x + c) - (B*a^3 - 3*C*a^
2*b - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)^2 + 1) + 2*(B*a^3 - 3*C*a^2*b - 3
*B*a*b^2)*log(abs(tan(d*x + c))) - (3*B*a^3*tan(d*x + c)^2 - 9*C*a^2*b*tan(
d*x + c)^2 - 9*B*a*b^2*tan(d*x + c)^2 - 2*C*a^3*tan(d*x + c) - 6*B*a^2*b*ta
n(d*x + c) - B*a^3)/tan(d*x + c)^2)/d
```

maple [A] time = 0.57, size = 186, normalized size = 1.46

$$-\frac{a^3B\left(\cot^2(dx+c)\right)}{2d}-\frac{a^3B\ln(\sin(dx+c))}{d}-a^3Cx-\frac{C\cot(dx+c)\,a^3}{d}-\frac{C\,a^3c}{d}-3Bx\,a^2b-\frac{3B\cot(dx+c)\,a^2b}{d}-\frac{3B\,a^2b}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^4*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -1/2/d*a^3*B*cot(d*x+c)^2-a^3*B*ln(sin(d*x+c))/d-a^3*C*x-1/d*C*cot(d*x+c)*a
^3-1/d*C*a^3*c-3*B*x*a^2*b-3/d*B*cot(d*x+c)*a^2*b-3/d*B*a^2*b*c+3/d*C*a^2*b
*ln(sin(d*x+c))+3/d*B*a*b^2*ln(sin(d*x+c))+3*a*b^2*C*x+3/d*C*a*b^2*c+B*x*b^
3+1/d*B*b^3*c-b^3*C*ln(cos(d*x+c))/d
```

maxima [A] time = 0.60, size = 142, normalized size = 1.12

$$-\frac{2\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)(dx+c)-\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)\log\left(\tan(dx+c)^2+1\right)+2\left(Ba^3-3\,C\right.}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] -1/2*(2*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*(d*x + c) - (B*a^3 - 3*C*a^
2*b - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)^2 + 1) + 2*(B*a^3 - 3*C*a^2*b - 3
*B*a*b^2)*log(tan(d*x + c)) + (B*a^3 + 2*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/
tan(d*x + c)^2)/d
```

mupad [B] time = 8.97, size = 135, normalized size = 1.06

$$\frac{\ln(\tan(c+d\,x))\,\left(-B\,a^3+3\,C\,a^2\,b+3\,B\,a\,b^2\right)}{d}-\frac{\cot(c+d\,x)^2\,\left(\tan(c+d\,x)\,\left(C\,a^3+3\,B\,b\,a^2\right)+\frac{B\,a^3}{2}\right)}{d}+\frac{\ln(\tan(c+$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^4*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(tan(c + d*x))*(3*B*a*b^2 - B*a^3 + 3*C*a^2*b))/d - (cot(c + d*x)^2*(ta
n(c + d*x)*(C*a^3 + 3*B*a^2*b) + (B*a^3)/2))/d + (log(tan(c + d*x) + 1i)*(B
 - C*1i)*(a*1i + b)^3*1i)/(2*d) + (log(tan(c + d*x) - 1i)*(B + C*1i)*(a*1i
- b)^3*1i)/(2*d)
```

sympy [A] time = 5.60, size = 260, normalized size = 2.05

$$\begin{cases} \text{NaN} \\ x(a+b\tan(c))^3 \left(B\tan(c)+C\tan^2(c)\right)\cot^4(c) \\ \frac{Ba^3\log\left(\tan^2(c+dx)+1\right)}{2d} - \frac{Ba^3\log(\tan(c+dx))}{d} - \frac{Ba^3}{2d\tan^2(c+dx)} - 3Ba^2bx - \frac{3Ba^2b}{d\tan(c+dx)} - \frac{3Bab^2\log\left(\tan^2(c+dx)+1\right)}{2d} + \frac{3Bab^2\log(\tan}{d} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**4*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, (Eq(c, 0) | Eq(c, -d*x)) & (Eq(d, 0) | Eq(c, -d*x))), (x*(a
 + b*tan(c))**3*(B*tan(c) + C*tan(c)**2)*cot(c)**4, Eq(d, 0)), (B*a**3*log(
tan(c + d*x)**2 + 1)/(2*d) - B*a**3*log(tan(c + d*x))/d - B*a**3/(2*d*tan(c
 + d*x)**2) - 3*B*a**2*b*x - 3*B*a**2*b/(d*tan(c + d*x)) - 3*B*a*b**2*log(t
an(c + d*x)**2 + 1)/(2*d) + 3*B*a*b**2*log(tan(c + d*x))/d + B*b**3*x - C*a
**3*x - C*a**3/(d*tan(c + d*x)) - 3*C*a**2*b*log(tan(c + d*x)**2 + 1)/(2*d)
 + 3*C*a**2*b*log(tan(c + d*x))/d + 3*C*a*b**2*x + C*b**3*log(tan(c + d*x)*
*2 + 1)/(2*d), True))
```

3.22 $\int \cot^5(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(\right.$

Optimal. Leaf size=154

$$\frac{a\left(3a^2B-9abC-8b^2B\right)\cot(c+dx)}{3d}-\frac{a^2(3aC+5bB)\cot^2(c+dx)}{6d}-\frac{\left(a^3C+3a^2bB-3ab^2C-b^3B\right)\log(\sin(c+dx)}{d}$$

```
[Out] (B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*x+1/3*a*(3*B*a^2-8*B*b^2-9*C*a*b)*cot(d*x
+c)/d-1/6*a^2*(5*B*b+3*C*a)*cot(d*x+c)^2/d-(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2
)*ln(sin(d*x+c))/d-1/3*a*B*cot(d*x+c)^3*(a+b*tan(d*x+c))^2/d
```

Rubi [A] time = 0.43, antiderivative size = 154, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3632, 3605, 3635, 3628, 3531, 3475}

$$\frac{a\left(3a^2B-9abC-8b^2B\right)\cot(c+dx)}{3d}-\frac{\left(3a^2bB+a^3C-3ab^2C-b^3B\right)\log(\sin(c+dx))}{d}+x\left(-3a^2bC+a^3B-3ab^2B\right.$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^5*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (a^3*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*x + (a*(3*a^2*B - 8*b^2*B - 9*a*b*C
)*Cot[c + d*x])/(3*d) - (a^2*(5*b*B + 3*a*C)*Cot[c + d*x]^2)/(6*d) - ((3*a^
2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*Log[Sin[c + d*x]])/d - (a*B*Cot[c + d*x]
^3*(a + b*Tan[c + d*x])^2)/(3*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3605

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
```

```
mp[((b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x
])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \cot^5(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int \cot^4(c+dx)(a+b\tan(c+dx))^3(B$$
$$= -\frac{aB\cot^3(c+dx)(a+b\tan(c+dx))^2}{3d}$$
$$= -\frac{a^2(5bB+3aC)\cot^2(c+dx)}{6d} - \frac{aB\cot}{}$$
$$= \frac{a\left(3a^2B-8b^2B-9abC\right)\cot(c+dx)}{3d} -$$
$$= \left(a^3B-3ab^2B-3a^2bC+b^3C\right)x + \frac{a\left(3\right.}{}$$
$$= \left(a^3B-3ab^2B-3a^2bC+b^3C\right)x + \frac{a\left(3\right.}{}$$

Mathematica [C] time = 1.28, size = 164, normalized size = 1.06

$$\frac{-2a^3B\cot^3(c+dx)+6a\left(a^2B-3abC-3b^2B\right)\cot(c+dx)-3a^2(aC+3bB)\cot^2(c+dx)-6\left(a^3C+3a^2bB-3ab^2\right.}{6d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^5*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (6*a*(a^2*B - 3*b^2*B - 3*a*b*C)*Cot[c + d*x] - 3*a^2*(3*b*B + a*C)*Cot[c +
 d*x]^2 - 2*a^3*B*Cot[c + d*x]^3 + 3*(a + I*b)^3*((-I)*B + C)*Log[I - Tan[c
 + d*x]] - 6*(3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*Log[Tan[c + d*x]] + 3*
(a - I*b)^3*(I*B + C)*Log[I + Tan[c + d*x]])/(6*d)
```

fricas [A] time = 0.63, size = 181, normalized size = 1.18

$$-\frac{3\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan(dx+c)^3+2\,Ba^3+3\left(Ca^3+3\,Ba^2b-2\left(Ba^3-3\,Ca^2b-\right.\right.}{6\,d\tan(dx}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] -1/6*(3*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2/(tan(d*x
 + c)^2 + 1))*tan(d*x + c)^3 + 2*B*a^3 + 3*(C*a^3 + 3*B*a^2*b - 2*(B*a^3 -
```

```
3*C*a^2*b - 3*B*a*b^2 + C*b^3)*d*x)*tan(d*x + c)^3 - 6*(B*a^3 - 3*C*a^2*b -
 3*B*a*b^2)*tan(d*x + c)^2 + 3*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/(d*tan(d*x
 + c)^3)
```

giac [B] time = 27.33, size = 390, normalized size = 2.53

$$Ba^3 \tan\left(\frac{1}{2}\,dx + \frac{1}{2}\,c\right)^3 - 3\,Ca^3 \tan\left(\frac{1}{2}\,dx + \frac{1}{2}\,c\right)^2 - 9\,Ba^2b \tan\left(\frac{1}{2}\,dx + \frac{1}{2}\,c\right)^2 - 15\,Ba^3 \tan\left(\frac{1}{2}\,dx + \frac{1}{2}\,c\right) + 36\,Ca^2b\,\mathrm{t}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/24*(B*a^3*tan(1/2*d*x + 1/2*c)^3 - 3*C*a^3*tan(1/2*d*x + 1/2*c)^2 - 9*B*a
^2*b*tan(1/2*d*x + 1/2*c)^2 - 15*B*a^3*tan(1/2*d*x + 1/2*c) + 36*C*a^2*b*ta
n(1/2*d*x + 1/2*c) + 36*B*a*b^2*tan(1/2*d*x + 1/2*c) + 24*(B*a^3 - 3*C*a^2*
b - 3*B*a*b^2 + C*b^3)*(d*x + c) + 24*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^
3)*log(tan(1/2*d*x + 1/2*c)^2 + 1) - 24*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*
b^3)*log(abs(tan(1/2*d*x + 1/2*c))) + (44*C*a^3*tan(1/2*d*x + 1/2*c)^3 + 13
2*B*a^2*b*tan(1/2*d*x + 1/2*c)^3 - 132*C*a*b^2*tan(1/2*d*x + 1/2*c)^3 - 44*
B*b^3*tan(1/2*d*x + 1/2*c)^3 + 15*B*a^3*tan(1/2*d*x + 1/2*c)^2 - 36*C*a^2*b
*tan(1/2*d*x + 1/2*c)^2 - 36*B*a*b^2*tan(1/2*d*x + 1/2*c)^2 - 3*C*a^3*tan(1
/2*d*x + 1/2*c) - 9*B*a^2*b*tan(1/2*d*x + 1/2*c) - B*a^3)/tan(1/2*d*x + 1/2
*c)^3)/d
```

maple [A] time = 0.52, size = 233, normalized size = 1.51

$$-\frac{a^3B\left(\cot^3(dx+c)\right)}{3d}+\frac{B\cot(dx+c)\,a^3}{d}+a^3Bx+\frac{a^3Bc}{d}-\frac{C\,a^3\left(\cot^2(dx+c)\right)}{2d}-\frac{C\,a^3\ln(\sin(dx+c))}{d}-\frac{3a^2bB\left(\cot^2(\right.}{2d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^5*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -1/3/d*a^3*B*cot(d*x+c)^3+1/d*B*cot(d*x+c)*a^3+a^3*B*x+1/d*a^3*B*c-1/2/d*C*
a^3*cot(d*x+c)^2-1/d*C*a^3*ln(sin(d*x+c))-3/2/d*a^2*b*B*cot(d*x+c)^2-3/d*a^
2*b*B*ln(sin(d*x+c))-3*C*x*a^2*b-3/d*C*cot(d*x+c)*a^2*b-3/d*C*a^2*b*c-3*B*x
*a*b^2-3/d*B*cot(d*x+c)*a*b^2-3/d*B*a*b^2*c+3/d*C*a*b^2*ln(sin(d*x+c))+1/d*
b^3*B*ln(sin(d*x+c))+b^3*C*x+1/d*b^3*C*c
```

maxima [A] time = 0.57, size = 180, normalized size = 1.17

$$\frac{6\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)(dx+c)+3\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)\log\left(\tan(dx+c)^2+1\right)-6\left(Ca^3+\right.}{6\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^5*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/6*(6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*(d*x + c) + 3*(C*a^3 + 3*B*a
^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2 + 1) - 6*(C*a^3 + 3*B*a^2*b -
3*C*a*b^2 - B*b^3)*log(tan(d*x + c)) - (2*B*a^3 - 6*(B*a^3 - 3*C*a^2*b - 3*
B*a*b^2)*tan(d*x + c)^2 + 3*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/tan(d*x + c)^
3)/d
```

mupad [B] time = 9.00, size = 169, normalized size = 1.10

$$\frac{\ln(\tan(c+d\,x))\,\left(-C\,a^3-3\,B\,a^2\,b+3\,C\,a\,b^2+B\,b^3\right)}{d}-\frac{\cot(c+d\,x)^3\,\left(\tan(c+d\,x)\,\left(\frac{C\,a^3}{2}+\frac{3\,B\,b\,a^2}{2}\right)+\frac{B\,a^3}{3}+\tan(c\right.}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^5*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(tan(c + d*x))*(B*b^3 - C*a^3 - 3*B*a^2*b + 3*C*a*b^2))/d - (cot(c + d*
x)^3*(tan(c + d*x)*((C*a^3)/2 + (3*B*a^2*b)/2) + (B*a^3)/3 + tan(c + d*x)^2
*(3*B*a*b^2 - B*a^3 + 3*C*a^2*b)))/d - (log(tan(c + d*x) - 1i)*(B + C*1i)*(
a + b*1i)^3*1i)/(2*d) + (log(tan(c + d*x) + 1i)*(B - C*1i)*(a - b*1i)^3*1i)
/(2*d)
```

sympy [A] time = 8.56, size = 330, normalized size = 2.14

$$\begin{cases} \text{NaN} \\ x\,(a+b\tan(c))^3\left(B\tan(c)+C\tan^2(c)\right)\cot^5(c) \\ Ba^3x+\frac{Ba^3}{d\tan(c+dx)}-\frac{Ba^3}{3d\tan^3(c+dx)}+\frac{3Ba^2b\log\left(\tan^2(c+dx)+1\right)}{2d}-\frac{3Ba^2b\log(\tan(c+dx))}{d}-\frac{3Ba^2b}{2d\tan^2(c+dx)}-3Bab^2x-\frac{3Bab^2}{d\tan(c+dx)} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**5*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, (Eq(c, 0) | Eq(c, -d*x)) & (Eq(d, 0) | Eq(c, -d*x))), (x*(a
 + b*tan(c))**3*(B*tan(c) + C*tan(c)**2)*cot(c)**5, Eq(d, 0)), (B*a**3*x +
B*a**3/(d*tan(c + d*x)) - B*a**3/(3*d*tan(c + d*x)**3) + 3*B*a**2*b*log(tan
(c + d*x)**2 + 1)/(2*d) - 3*B*a**2*b*log(tan(c + d*x))/d - 3*B*a**2*b/(2*d*
tan(c + d*x)**2) - 3*B*a*b**2*x - 3*B*a*b**2/(d*tan(c + d*x)) - B*b**3*log(
```

```
tan(c + d*x)**2 + 1)/(2*d) + B*b**3*log(tan(c + d*x))/d + C*a**3*log(tan(c
+ d*x)**2 + 1)/(2*d) - C*a**3*log(tan(c + d*x))/d - C*a**3/(2*d*tan(c + d*x
)**2) - 3*C*a**2*b*x - 3*C*a**2*b/(d*tan(c + d*x)) - 3*C*a*b**2*log(tan(c +
 d*x)**2 + 1)/(2*d) + 3*C*a*b**2*log(tan(c + d*x))/d + C*b**3*x, True))
```

3.23 $\int \cot^6(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(\right.$

Optimal. Leaf size=191

$$\frac{a\left(2a^2B-6abC-5b^2B\right)\cot^2(c+dx)}{4d}-\frac{a^2(2aC+3bB)\cot^3(c+dx)}{6d}+\frac{\left(a^3C+3a^2bB-3ab^2C-b^3B\right)\cot(c+dx)}{d}+$$

```
[Out] (3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*x+(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*cot(d
*x+c)/d+1/4*a*(2*B*a^2-5*B*b^2-6*C*a*b)*cot(d*x+c)^2/d-1/6*a^2*(3*B*b+2*C*a
)*cot(d*x+c)^3/d+(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*ln(sin(d*x+c))/d-1/4*a*B
*cot(d*x+c)^4*(a+b*tan(d*x+c))^2/d
```

Rubi [A] time = 0.51, antiderivative size = 191, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.175, Rules used = {3632, 3605, 3635, 3628, 3529, 3531, 3475}

$$\frac{a\left(2a^2B-6abC-5b^2B\right)\cot^2(c+dx)}{4d}+\frac{\left(3a^2bB+a^3C-3ab^2C-b^3B\right)\cot(c+dx)}{d}+\frac{\left(-3a^2bC+a^3B-3ab^2B+b^3C\right.}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^6*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*x + ((3*a^2*b*B - b^3*B + a^3*C - 3
*a*b^2*C)*Cot[c + d*x])/d + (a*(2*a^2*B - 5*b^2*B - 6*a*b*C)*Cot[c + d*x]^2
)/(4*d) - (a^2*(3*b*B + 2*a*C)*Cot[c + d*x]^3)/(6*d) + ((a^3*B - 3*a*b^2*B
- 3*a^2*b*C + b^3*C)*Log[Sin[c + d*x]])/d - (a*B*Cot[c + d*x]^4*(a + b*Tan[
c + d*x])^2)/(4*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3605

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[((b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x
])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
```

```
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \cot^6(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(c+dx)\right) dx = \int \cot^5(c+dx)(a+b\tan(c+dx))^3(B -$$

$$= -\frac{aB\cot^4(c+dx)(a+b\tan(c+dx))^2}{4d} +$$

$$= -\frac{a^2(3bB+2aC)\cot^3(c+dx)}{6d} - \frac{aB\cot}{}$$

$$= \frac{a\left(2a^2B-5b^2B-6abC\right)\cot^2(c+dx)}{4d} -$$

$$= \frac{\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\cot(c+}{d}$$

$$= \left(3a^2bB-b^3B+a^3C-3ab^2C\right)x + \frac{(3a}{}$$

$$= \left(3a^2bB-b^3B+a^3C-3ab^2C\right)x + \frac{(3a}{}$$

Mathematica [C] time = 0.79, size = 199, normalized size = 1.04

$$\frac{-3a^3B\cot^4(c+dx)+6a\left(a^2B-3abC-3b^2B\right)\cot^2(c+dx)-4a^2(aC+3bB)\cot^3(c+dx)+12\left(a^3C+3a^2bB-3a\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^6*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (12*(3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*Cot[c + d*x] + 6*a*(a^2*B - 3*b
^2*B - 3*a*b*C)*Cot[c + d*x]^2 - 4*a^2*(3*b*B + a*C)*Cot[c + d*x]^3 - 3*a^3
*B*Cot[c + d*x]^4 - 6*(a + I*b)^3*(B + I*C)*Log[I - Tan[c + d*x]] + 12*(a^3
*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*Log[Tan[c + d*x]] - 6*(a - I*b)^3*(B -
I*C)*Log[I + Tan[c + d*x]])/(12*d)
```

fricas [A] time = 0.74, size = 225, normalized size = 1.18

$$\frac{6\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan(dx+c)^4+3\left(3\,Ba^3-6\,Ca^2b-6\,Bab^2+4\left(Ca^3+3\,Ba^2b-\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/12*(6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)^2/(tan(d*x
 + c)^2 + 1))*tan(d*x + c)^4 + 3*(3*B*a^3 - 6*C*a^2*b - 6*B*a*b^2 + 4*(C*a^
3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*d*x)*tan(d*x + c)^4 - 3*B*a^3 + 12*(C*a^
3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*tan(d*x + c)^3 + 6*(B*a^3 - 3*C*a^2*b -
3*B*a*b^2)*tan(d*x + c)^2 - 4*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/(d*tan(d*x
+ c)^4)
```

giac [B] time = 90.46, size = 528, normalized size = 2.76

$$-\frac{3\,Ba^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^4-8\,Ca^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-24\,Ba^2b\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3-36\,Ba^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^2+72\,C}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] -1/192*(3*B*a^3*tan(1/2*d*x + 1/2*c)^4 - 8*C*a^3*tan(1/2*d*x + 1/2*c)^3 - 2
4*B*a^2*b*tan(1/2*d*x + 1/2*c)^3 - 36*B*a^3*tan(1/2*d*x + 1/2*c)^2 + 72*C*a
^2*b*tan(1/2*d*x + 1/2*c)^2 + 72*B*a*b^2*tan(1/2*d*x + 1/2*c)^2 + 120*C*a^3
*tan(1/2*d*x + 1/2*c) + 360*B*a^2*b*tan(1/2*d*x + 1/2*c) - 288*C*a*b^2*tan(
1/2*d*x + 1/2*c) - 96*B*b^3*tan(1/2*d*x + 1/2*c) - 192*(C*a^3 + 3*B*a^2*b -
 3*C*a*b^2 - B*b^3)*(d*x + c) + 192*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)
*log(tan(1/2*d*x + 1/2*c)^2 + 1) - 192*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b
^3)*log(abs(tan(1/2*d*x + 1/2*c))) + (400*B*a^3*tan(1/2*d*x + 1/2*c)^4 - 12
00*C*a^2*b*tan(1/2*d*x + 1/2*c)^4 - 1200*B*a*b^2*tan(1/2*d*x + 1/2*c)^4 + 4
00*C*b^3*tan(1/2*d*x + 1/2*c)^4 - 120*C*a^3*tan(1/2*d*x + 1/2*c)^3 - 360*B*
a^2*b*tan(1/2*d*x + 1/2*c)^3 + 288*C*a*b^2*tan(1/2*d*x + 1/2*c)^3 + 96*B*b^
3*tan(1/2*d*x + 1/2*c)^3 - 36*B*a^3*tan(1/2*d*x + 1/2*c)^2 + 72*C*a^2*b*tan
(1/2*d*x + 1/2*c)^2 + 72*B*a*b^2*tan(1/2*d*x + 1/2*c)^2 + 8*C*a^3*tan(1/2*d
*x + 1/2*c) + 24*B*a^2*b*tan(1/2*d*x + 1/2*c) + 3*B*a^3)/tan(1/2*d*x + 1/2*
c)^4)/d
```

maple [A] time = 0.55, size = 302, normalized size = 1.58

$$-\frac{a^3B\left(\cot^4\left(dx+c\right)\right)}{4d}+\frac{a^3B\left(\cot^2\left(dx+c\right)\right)}{2d}+\frac{a^3B\ln\left(\sin\left(dx+c\right)\right)}{d}-\frac{C\,a^3\left(\cot^3\left(dx+c\right)\right)}{3d}+\frac{C\cot\left(dx+c\right)a^3}{d}+a^3Cx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^6*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -1/4/d*a^3*B*cot(d*x+c)^4+1/2/d*a^3*B*cot(d*x+c)^2+a^3*B*ln(sin(d*x+c))/d-1
/3/d*C*a^3*cot(d*x+c)^3+1/d*C*cot(d*x+c)*a^3+a^3*C*x+1/d*C*a^3*c-1/d*a^2*b*
B*cot(d*x+c)^3+3*B*x*a^2*b+3/d*B*cot(d*x+c)*a^2*b+3/d*B*a^2*b*c-3/2/d*C*a^2
*b*cot(d*x+c)^2-3/d*C*a^2*b*ln(sin(d*x+c))-3/2/d*B*a*b^2*cot(d*x+c)^2-3/d*B
*a*b^2*ln(sin(d*x+c))-3*a*b^2*C*x-3/d*C*cot(d*x+c)*a*b^2-3/d*C*a*b^2*c-B*x*
b^3-1/d*B*cot(d*x+c)*b^3-1/d*B*b^3*c+1/d*b^3*C*ln(sin(d*x+c))
```

maxima [A] time = 0.74, size = 215, normalized size = 1.13

$$\frac{12\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)(dx+c)-6\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)\log\left(\tan\left(dx+c\right)^2+1\right)+12\left(Ba^3-\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^6*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] 1/12*(12*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*(d*x + c) - 6*(B*a^3 - 3*C
*a^2*b - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)^2 + 1) + 12*(B*a^3 - 3*C*a^2*b
 - 3*B*a*b^2 + C*b^3)*log(tan(d*x + c)) - (3*B*a^3 - 12*(C*a^3 + 3*B*a^2*b
- 3*C*a*b^2 - B*b^3)*tan(d*x + c)^3 - 6*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2)*tan
(d*x + c)^2 + 4*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/tan(d*x + c)^4)/d
```

mupad [B] time = 8.94, size = 204, normalized size = 1.07

$$\frac{\ln\left(\tan\left(c+d\,x\right)\right)\,\left(B\,a^3-3\,C\,a^2\,b-3\,B\,a\,b^2+C\,b^3\right)}{d}-\frac{\cot\left(c+d\,x\right)^4\,\left(\tan\left(c+d\,x\right)\,\left(\frac{C\,a^3}{3}+B\,b\,a^2\right)+\frac{B\,a^3}{4}+\tan\left(c+\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^6*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(tan(c + d*x))*(B*a^3 + C*b^3 - 3*B*a*b^2 - 3*C*a^2*b))/d - (cot(c + d*
x)^4*(tan(c + d*x)*((C*a^3)/3 + B*a^2*b) + (B*a^3)/4 + tan(c + d*x)^2*((3*B
*a*b^2)/2 - (B*a^3)/2 + (3*C*a^2*b)/2) + tan(c + d*x)^3*(B*b^3 - C*a^3 - 3*
B*a^2*b + 3*C*a*b^2)))/d - (log(tan(c + d*x) + 1i)*(B - C*1i)*(a*1i + b)^3*
1i)/(2*d) - (log(tan(c + d*x) - 1i)*(B + C*1i)*(a*1i - b)^3*1i)/(2*d)
```

sympy [A] time = 11.01, size = 398, normalized size = 2.08

$$\begin{cases} \text{NaN} \\ x\left(a+b\tan\left(c\right)\right)^{3}\left(B\tan\left(c\right)+C\tan^{2}\left(c\right)\right)\cot^{6}\left(c\right) \\ -\frac{Ba^{3}\log\left(\tan^{2}\left(c+dx\right)+1\right)}{2d}+\frac{Ba^{3}\log\left(\tan\left(c+dx\right)\right)}{d}+\frac{Ba^{3}}{2d\tan^{2}\left(c+dx\right)}-\frac{Ba^{3}}{4d\tan^{4}\left(c+dx\right)}+3Ba^{2}bx+\frac{3Ba^{2}b}{d\tan\left(c+dx\right)}-\frac{Ba^{2}b}{d\tan^{3}\left(c+dx\right)}+\frac{3Bab^{2}}{} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**6*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, (Eq(c, 0) | Eq(c, -d*x)) & (Eq(d, 0) | Eq(c, -d*x))), (x*(a
 + b*tan(c))**3*(B*tan(c) + C*tan(c)**2)*cot(c)**6, Eq(d, 0)), (-B*a**3*log
(tan(c + d*x)**2 + 1)/(2*d) + B*a**3*log(tan(c + d*x))/d + B*a**3/(2*d*tan(
c + d*x)**2) - B*a**3/(4*d*tan(c + d*x)**4) + 3*B*a**2*b*x + 3*B*a**2*b/(d*
tan(c + d*x)) - B*a**2*b/(d*tan(c + d*x)**3) + 3*B*a*b**2*log(tan(c + d*x)*
*2 + 1)/(2*d) - 3*B*a*b**2*log(tan(c + d*x))/d - 3*B*a*b**2/(2*d*tan(c + d*
x)**2) - B*b**3*x - B*b**3/(d*tan(c + d*x)) + C*a**3*x + C*a**3/(d*tan(c +
d*x)) - C*a**3/(3*d*tan(c + d*x)**3) + 3*C*a**2*b*log(tan(c + d*x)**2 + 1)/
(2*d) - 3*C*a**2*b*log(tan(c + d*x))/d - 3*C*a**2*b/(2*d*tan(c + d*x)**2) -
 3*C*a*b**2*x - 3*C*a*b**2/(d*tan(c + d*x)) - C*b**3*log(tan(c + d*x)**2 +
1)/(2*d) + C*b**3*log(tan(c + d*x))/d, True))
```

3.24 $\int \cot^7(c+dx)(a+b\tan(c+dx))^3 \left(B\tan(c+dx)+C\tan^2(\right.$

Optimal. Leaf size=233

$$\frac{a\left(5a^2B-15abC-12b^2B\right)\cot^3(c+dx)}{15d}-\frac{a^2(5aC+7bB)\cot^4(c+dx)}{20d}+\frac{\left(a^3C+3a^2bB-3ab^2C-b^3B\right)\cot^2(c+dx)}{2d}$$

```
[Out] -(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*x-(B*a^3-3*B*a*b^2-3*C*a^2*b+C*b^3)*cot(
d*x+c)/d+1/2*(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*cot(d*x+c)^2/d+1/15*a*(5*B*a
^2-12*B*b^2-15*C*a*b)*cot(d*x+c)^3/d-1/20*a^2*(7*B*b+5*C*a)*cot(d*x+c)^4/d+
(3*B*a^2*b-B*b^3+C*a^3-3*C*a*b^2)*ln(sin(d*x+c))/d-1/5*a*B*cot(d*x+c)^5*(a+
b*tan(d*x+c))^2/d
```

Rubi [A] time = 0.56, antiderivative size = 233, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 7, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.175, Rules used = {3632, 3605, 3635, 3628, 3529, 3531, 3475}

$$\frac{a\left(5a^2B-15abC-12b^2B\right)\cot^3(c+dx)}{15d}+\frac{\left(3a^2bB+a^3C-3ab^2C-b^3B\right)\cot^2(c+dx)}{2d}-\frac{\left(-3a^2bC+a^3B-3ab^2B+\right.}{d}$$

Antiderivative was successfully verified.

```
[In] Int[Cot[c + d*x]^7*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] -((a^3*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*x) - ((a^3*B - 3*a*b^2*B - 3*a^2*
b*C + b^3*C)*Cot[c + d*x])/d + ((3*a^2*b*B - b^3*B + a^3*C - 3*a*b^2*C)*Cot
[c + d*x]^2)/(2*d) + (a*(5*a^2*B - 12*b^2*B - 15*a*b*C)*Cot[c + d*x]^3)/(15
*d) - (a^2*(7*b*B + 5*a*C)*Cot[c + d*x]^4)/(20*d) + ((3*a^2*b*B - b^3*B + a
^3*C - 3*a*b^2*C)*Log[Sin[c + d*x]])/d - (a*B*Cot[c + d*x]^5*(a + b*Tan[c +
 d*x])^2)/(5*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3605

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[((b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x
])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
```

```
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \cot^7(c+dx)(a+b\tan(c+dx))^3\left(B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \int \cot^6(c+dx)(a+b\tan(c+dx))^3(B$$

$$= -\frac{aB\cot^5(c+dx)(a+b\tan(c+dx))^2}{5d}$$

$$= -\frac{a^2(7bB+5aC)\cot^4(c+dx)}{20d} - \frac{aB\cot}{}$$

$$= \frac{a\left(5a^2B-12b^2B-15abC\right)\cot^3(c+dx}{15d}$$

$$= \frac{\left(3a^2bB-b^3B+a^3C-3ab^2C\right)\cot^2(c+}{2d}$$

$$= -\frac{\left(a^3B-3ab^2B-3a^2bC+b^3C\right)\cot(c}{d}$$

$$= -\left(a^3B-3ab^2B-3a^2bC+b^3C\right)x - \frac{(a}{}$$

$$= -\left(a^3B-3ab^2B-3a^2bC+b^3C\right)x - \frac{(a}{}$$

Mathematica [C] time = 1.21, size = 237, normalized size = 1.02

$$\frac{-12a^3B\cot^5(c+dx)+20a\left(a^2B-3abC-3b^2B\right)\cot^3(c+dx)-15a^2(aC+3bB)\cot^4(c+dx)+30\left(a^3C+3a^2bB\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[Cot[c + d*x]^7*(a + b*Tan[c + d*x])^3*(B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (-60*(a^3*B - 3*a*b^2*B - 3*a^2*b*C + b^3*C)*Cot[c + d*x] + 30*(3*a^2*b*B -
 b^3*B + a^3*C - 3*a*b^2*C)*Cot[c + d*x]^2 + 20*a*(a^2*B - 3*b^2*B - 3*a*b*
C)*Cot[c + d*x]^3 - 15*a^2*(3*b*B + a*C)*Cot[c + d*x]^4 - 12*a^3*B*Cot[c +
d*x]^5 + (30*I)*(a + I*b)^3*(B + I*C)*Log[I - Tan[c + d*x]] + 60*(3*a^2*b*B
```

```
 - b^3*B + a^3*C - 3*a*b^2*C)*Log[Tan[c + d*x]] + 30*(I*a + b)^3*(B - I*C)*
Log[I + Tan[c + d*x]])/(60*d)
```

fricas [A] time = 1.41, size = 266, normalized size = 1.14

$$30\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)^5+15\left(3\,Ca^3+9\,Ba^2b-6\,Cab^2-2\,Bb^3-4\left(Ba\right.\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^7*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] 1/60*(30*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2/(tan(d*
x + c)^2 + 1))*tan(d*x + c)^5 + 15*(3*C*a^3 + 9*B*a^2*b - 6*C*a*b^2 - 2*B*b
^3 - 4*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*d*x)*tan(d*x + c)^5 - 60*(B*
a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*tan(d*x + c)^4 - 12*B*a^3 + 30*(C*a^3
+ 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*tan(d*x + c)^3 + 20*(B*a^3 - 3*C*a^2*b - 3
*B*a*b^2)*tan(d*x + c)^2 - 15*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/(d*tan(d*x
+ c)^5)
```

giac [B] time = 61.54, size = 670, normalized size = 2.88

$$6\,Ba^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^5-15\,Ca^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^4-45\,Ba^2b\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^4-70\,Ba^3\tan\left(\frac{1}{2}\,dx+\frac{1}{2}\,c\right)^3+120$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^7*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] 1/960*(6*B*a^3*tan(1/2*d*x + 1/2*c)^5 - 15*C*a^3*tan(1/2*d*x + 1/2*c)^4 - 4
5*B*a^2*b*tan(1/2*d*x + 1/2*c)^4 - 70*B*a^3*tan(1/2*d*x + 1/2*c)^3 + 120*C*
a^2*b*tan(1/2*d*x + 1/2*c)^3 + 120*B*a*b^2*tan(1/2*d*x + 1/2*c)^3 + 180*C*a
^3*tan(1/2*d*x + 1/2*c)^2 + 540*B*a^2*b*tan(1/2*d*x + 1/2*c)^2 - 360*C*a*b^
2*tan(1/2*d*x + 1/2*c)^2 - 120*B*b^3*tan(1/2*d*x + 1/2*c)^2 + 660*B*a^3*tan
(1/2*d*x + 1/2*c) - 1800*C*a^2*b*tan(1/2*d*x + 1/2*c) - 1800*B*a*b^2*tan(1/
2*d*x + 1/2*c) + 480*C*b^3*tan(1/2*d*x + 1/2*c) - 960*(B*a^3 - 3*C*a^2*b -
3*B*a*b^2 + C*b^3)*(d*x + c) - 960*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^3)*
log(tan(1/2*d*x + 1/2*c)^2 + 1) + 960*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2 - B*b^
3)*log(abs(tan(1/2*d*x + 1/2*c))) - (2192*C*a^3*tan(1/2*d*x + 1/2*c)^5 + 65
76*B*a^2*b*tan(1/2*d*x + 1/2*c)^5 - 6576*C*a*b^2*tan(1/2*d*x + 1/2*c)^5 - 2
192*B*b^3*tan(1/2*d*x + 1/2*c)^5 + 660*B*a^3*tan(1/2*d*x + 1/2*c)^4 - 1800*
```

```
C*a^2*b*tan(1/2*d*x + 1/2*c)^4 - 1800*B*a*b^2*tan(1/2*d*x + 1/2*c)^4 + 480*
C*b^3*tan(1/2*d*x + 1/2*c)^4 - 180*C*a^3*tan(1/2*d*x + 1/2*c)^3 - 540*B*a^2
*b*tan(1/2*d*x + 1/2*c)^3 + 360*C*a*b^2*tan(1/2*d*x + 1/2*c)^3 + 120*B*b^3*
tan(1/2*d*x + 1/2*c)^3 - 70*B*a^3*tan(1/2*d*x + 1/2*c)^2 + 120*C*a^2*b*tan(
1/2*d*x + 1/2*c)^2 + 120*B*a*b^2*tan(1/2*d*x + 1/2*c)^2 + 15*C*a^3*tan(1/2*
d*x + 1/2*c) + 45*B*a^2*b*tan(1/2*d*x + 1/2*c) + 6*B*a^3)/tan(1/2*d*x + 1/2
*c)^5)/d
```

maple [A] time = 0.54, size = 376, normalized size = 1.61

$$-a^3Bx-\frac{3Ca\,b^2\left(\cot^2(dx+c)\right)}{2d}-\frac{C\,a^2b\left(\cot^3(dx+c)\right)}{d}-\frac{3a^2bB\left(\cot^4(dx+c)\right)}{4d}-\frac{Ba\,b^2\left(\cot^3(dx+c)\right)}{d}-\frac{a^3Bc}{d}+\frac{C\,a^3\,(}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^7*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] -a^3*B*x+3/2/d*a^2*b*B*cot(d*x+c)^2-1/d*a^3*B*c+1/d*C*a^3*ln(sin(d*x+c))-1/
d*b^3*C*c+3*C*x*a^2*b+3*B*x*a*b^2-1/d*B*cot(d*x+c)*a^3+1/3/d*a^3*B*cot(d*x+
c)^3+1/2/d*C*a^3*cot(d*x+c)^2-1/d*b^3*B*ln(sin(d*x+c))-1/4/d*C*a^3*cot(d*x+
c)^4-1/5/d*a^3*B*cot(d*x+c)^5-1/2/d*b^3*B*cot(d*x+c)^2-1/d*C*cot(d*x+c)*b^3
+3/d*C*cot(d*x+c)*a^2*b+3/d*B*cot(d*x+c)*a*b^2-3/d*C*a*b^2*ln(sin(d*x+c))+3
/d*C*a^2*b*c+3/d*B*a*b^2*c-b^3*C*x-3/2/d*C*a*b^2*cot(d*x+c)^2-1/d*C*a^2*b*c
ot(d*x+c)^3-3/4/d*a^2*b*B*cot(d*x+c)^4-1/d*B*a*b^2*cot(d*x+c)^3+3/d*a^2*b*B
*ln(sin(d*x+c))
```

maxima [A] time = 0.76, size = 250, normalized size = 1.07

$$-\frac{60\left(Ba^3-3\,Ca^2b-3\,Bab^2+Cb^3\right)(dx+c)+30\left(Ca^3+3\,Ba^2b-3\,Cab^2-Bb^3\right)\log\left(\tan(dx+c)^2+1\right)-60\left(Ca\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^7*(a+b*tan(d*x+c))^3*(B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] -1/60*(60*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2 + C*b^3)*(d*x + c) + 30*(C*a^3 + 3
*B*a^2*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)^2 + 1) - 60*(C*a^3 + 3*B*a^2
*b - 3*C*a*b^2 - B*b^3)*log(tan(d*x + c)) + (60*(B*a^3 - 3*C*a^2*b - 3*B*a*
b^2 + C*b^3)*tan(d*x + c)^4 + 12*B*a^3 - 30*(C*a^3 + 3*B*a^2*b - 3*C*a*b^2
- B*b^3)*tan(d*x + c)^3 - 20*(B*a^3 - 3*C*a^2*b - 3*B*a*b^2)*tan(d*x + c)^2
 + 15*(C*a^3 + 3*B*a^2*b)*tan(d*x + c))/tan(d*x + c)^5)/d
```

mupad [B] time = 9.12, size = 238, normalized size = 1.02

$$-\frac{\cot(c+d\,x)^5\,\left(\tan(c+d\,x)\,\left(\frac{C\,a^3}{4}+\frac{3\,B\,b\,a^2}{4}\right)+\frac{B\,a^3}{5}+\tan(c+d\,x)^2\,\left(-\frac{B\,a^3}{3}+C\,a^2\,b+B\,a\,b^2\right)+\tan(c+d\,x)^4\,\left(B\,a\right.\right.}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(c + d*x)^7*(B*tan(c + d*x) + C*tan(c + d*x)^2)*(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B + C*1i)*(a + b*1i)^3*1i)/(2*d) - (log(tan(c + d*
x))*(B*b^3 - C*a^3 - 3*B*a^2*b + 3*C*a*b^2))/d - (cot(c + d*x)^5*(tan(c + d
*x)*((C*a^3)/4 + (3*B*a^2*b)/4) + (B*a^3)/5 + tan(c + d*x)^2*(B*a*b^2 - (B*
a^3)/3 + C*a^2*b) + tan(c + d*x)^4*(B*a^3 + C*b^3 - 3*B*a*b^2 - 3*C*a^2*b)
+ tan(c + d*x)^3*((B*b^3)/2 - (C*a^3)/2 - (3*B*a^2*b)/2 + (3*C*a*b^2)/2)))/
d - (log(tan(c + d*x) + 1i)*(B - C*1i)*(a - b*1i)^3*1i)/(2*d)
```

sympy [A] time = 27.65, size = 469, normalized size = 2.01

$$\begin{cases} \text{NaN} \\ x\left(a+b\tan\left(c\right)\right)^{3}\left(B\tan\left(c\right)+C\tan^{2}\left(c\right)\right)\cot^{7}\left(c\right) \\ -Ba^{3}x-\frac{Ba^{3}}{d\tan\left(c+dx\right)}+\frac{Ba^{3}}{3d\tan^{3}\left(c+dx\right)}-\frac{Ba^{3}}{5d\tan^{5}\left(c+dx\right)}-\frac{3Ba^{2}b\log\left(\tan^{2}\left(c+dx\right)+1\right)}{2d}+\frac{3Ba^{2}b\log\left(\tan\left(c+dx\right)\right)}{d}+\frac{3Ba^{2}b}{2d\tan^{2}\left(c+dx\right)}-\frac{\quad}{4d\,\mathrm{ta}} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**7*(a+b*tan(d*x+c))**3*(B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Piecewise((nan, (Eq(c, 0) | Eq(c, -d*x)) & (Eq(d, 0) | Eq(c, -d*x))), (x*(a
 + b*tan(c))**3*(B*tan(c) + C*tan(c)**2)*cot(c)**7, Eq(d, 0)), (-B*a**3*x -
 B*a**3/(d*tan(c + d*x)) + B*a**3/(3*d*tan(c + d*x)**3) - B*a**3/(5*d*tan(c
 + d*x)**5) - 3*B*a**2*b*log(tan(c + d*x)**2 + 1)/(2*d) + 3*B*a**2*b*log(ta
n(c + d*x))/d + 3*B*a**2*b/(2*d*tan(c + d*x)**2) - 3*B*a**2*b/(4*d*tan(c +
d*x)**4) + 3*B*a*b**2*x + 3*B*a*b**2/(d*tan(c + d*x)) - B*a*b**2/(d*tan(c +
 d*x)**3) + B*b**3*log(tan(c + d*x)**2 + 1)/(2*d) - B*b**3*log(tan(c + d*x)
)/d - B*b**3/(2*d*tan(c + d*x)**2) - C*a**3*log(tan(c + d*x)**2 + 1)/(2*d)
+ C*a**3*log(tan(c + d*x))/d + C*a**3/(2*d*tan(c + d*x)**2) - C*a**3/(4*d*t
an(c + d*x)**4) + 3*C*a**2*b*x + 3*C*a**2*b/(d*tan(c + d*x)) - C*a**2*b/(d*
tan(c + d*x)**3) + 3*C*a*b**2*log(tan(c + d*x)**2 + 1)/(2*d) - 3*C*a*b**2*l
og(tan(c + d*x))/d - 3*C*a*b**2/(2*d*tan(c + d*x)**2) - C*b**3*x - C*b**3/(
d*tan(c + d*x)), True))
```

3.25 $\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$

Optimal. Leaf size=127

$$\frac{(aB+bC)\log(\cos(c+dx))}{d\left(a^2+b^2\right)}-\frac{x(bB-aC)}{a^2+b^2}-\frac{a^3(bB-aC)\log(a+b\tan(c+dx))}{b^3d\left(a^2+b^2\right)}+\frac{(bB-aC)\tan(c+dx)}{b^2d}+\frac{C\tan^2(c+dx)}{2bd}$$

```
[Out] -(B*b-C*a)*x/(a^2+b^2)+(B*a+C*b)*ln(cos(d*x+c))/(a^2+b^2)/d-a^3*(B*b-C*a)*l
n(a+b*tan(d*x+c))/b^3/(a^2+b^2)/d+(B*b-C*a)*tan(d*x+c)/b^2/d+1/2*C*tan(d*x+
c)^2/b/d
```

Rubi [A] time = 0.47, antiderivative size = 127, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.175, Rules used = {3632, 3607, 3647, 3626, 3617, 31, 3475}

$$-\frac{a^3(bB-aC)\log(a+b\tan(c+dx))}{b^3d\left(a^2+b^2\right)}+\frac{(aB+bC)\log(\cos(c+dx))}{d\left(a^2+b^2\right)}-\frac{x(bB-aC)}{a^2+b^2}+\frac{(bB-aC)\tan(c+dx)}{b^2d}+\frac{C\tan^2(c+dx)}{2bd}$$

Antiderivative was successfully verified.

```
[In] Int[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
]),x]
```

```
[Out] -(((b*B - a*C)*x)/(a^2 + b^2)) + ((a*B + b*C)*Log[Cos[c + d*x]])/((a^2 + b^
2)*d) - (a^3*(b*B - a*C)*Log[a + b*Tan[c + d*x]])/(b^3*(a^2 + b^2)*d) + ((b
*B - a*C)*Tan[c + d*x])/(b^2*d) + (C*Tan[c + d*x]^2)/(2*b*d)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3607

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*B*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(m
+ n)), x] + Dist[1/(d*(m + n)), Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan
[e + f*x])^n*Simp[a^2*A*d*(m + n) - b*B*(b*c*(m - 1) + a*d*(n + 1)) + d*(m
```

```
+ n)*(2*a*A*b + B*(a^2 - b^2))*Tan[e + f*x] - (b*B*(b*c - a*d)*(m - 1) - b*
(A*b + a*B)*d*(m + n))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e,
f, A, B, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2,
 0] && GtQ[m, 1] && (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(IGtQ[n, 1] &
& ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx=\int \frac{\tan^3(c+dx)(B+C\tan(c+dx))}{a+b\tan(c+dx)}\,dx$$

$$=\frac{C\tan^2(c+dx)}{2bd}+\frac{\int \frac{\tan(c+dx)\left(-2aC-2bC\tan(c+dx)+2(bB-aC)\tan^2(\right.}{a+b\tan(c+dx)}}{2b}$$

$$=\frac{(bB-aC)\tan(c+dx)}{b^2d}+\frac{C\tan^2(c+dx)}{2bd}+\frac{\int \frac{-2a(bB-aC)-2b}{}}{}$$

$$=-\frac{(bB-aC)x}{a^2+b^2}+\frac{(bB-aC)\tan(c+dx)}{b^2d}+\frac{C\tan^2(c+dx)}{2bd}-$$

$$=-\frac{(bB-aC)x}{a^2+b^2}+\frac{(aB+bC)\log(\cos(c+dx))}{\left(a^2+b^2\right)d}+\frac{(bB-aC)\,\mathrm{t}}{b^2}$$

$$=-\frac{(bB-aC)x}{a^2+b^2}+\frac{(aB+bC)\log(\cos(c+dx))}{\left(a^2+b^2\right)d}-\frac{a^3(bB-aC}{b}$$

Mathematica [C] time = 1.45, size = 138, normalized size = 1.09

$$\frac{\frac{2a^3(aC-bB)\log(a+b\tan(c+dx))}{b^2\left(a^2+b^2\right)}+\frac{2(bB-aC)\tan(c+dx)}{b}-\frac{b(B+iC)\log(-\tan(c+dx)+i)}{a+ib}-\frac{b(B-iC)\log(\tan(c+dx)+i)}{a-ib}+C\tan^2(c+dx)}{2bd}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x]),x]
```

```
[Out] (-((b*(B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)) - (b*(B - I*C)*Log[I + Ta
n[c + d*x]])/(a - I*b) + (2*a^3*(-(b*B) + a*C)*Log[a + b*Tan[c + d*x]])/(b^
2*(a^2 + b^2)) + (2*(b*B - a*C)*Tan[c + d*x])/b + C*Tan[c + d*x]^2)/(2*b*d)
```

fricas [A] time = 0.87, size = 190, normalized size = 1.50

$$\frac{2\left(Cab^3-Bb^4\right)dx+\left(Ca^2b^2+Cb^4\right)\tan\left(dx+c\right)^2+\left(Ca^4-Ba^3b\right)\log\left(\frac{b^2\tan(dx+c)^2+2\,ab\tan(dx+c)+a^2}{\tan(dx+c)^2+1}\right)-\left(Ca^4-Ba^3b\right.}{2\left(a^2b^3+b^5\right)d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="fricas")
```

```
[Out] 1/2*(2*(C*a*b^3 - B*b^4)*d*x + (C*a^2*b^2 + C*b^4)*tan(d*x + c)^2 + (C*a^4
- B*a^3*b)*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c
)^2 + 1)) - (C*a^4 - B*a^3*b - B*a*b^3 - C*b^4)*log(1/(tan(d*x + c)^2 + 1))
 - 2*(C*a^3*b - B*a^2*b^2 + C*a*b^3 - B*b^4)*tan(d*x + c))/((a^2*b^3 + b^5)
*d)
```

giac [A] time = 2.11, size = 135, normalized size = 1.06

$$\frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{2\left(Ca^4-Ba^3b\right)\log(|b\tan(dx+c)+a|)}{a^2b^3+b^5}+\frac{Cb\tan(dx+c)^2-2\,Ca\tan(dx+c)+2\,Bb\tan(dx+c)}{b^2}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="giac")
```

```
[Out] 1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - (B*a + C*b)*log(tan(d*x + c)^2 +
 1)/(a^2 + b^2) + 2*(C*a^4 - B*a^3*b)*log(abs(b*tan(d*x + c) + a))/(a^2*b^3
 + b^5) + (C*b*tan(d*x + c)^2 - 2*C*a*tan(d*x + c) + 2*B*b*tan(d*x + c))/b^
2)/d
```

maple [A] time = 0.26, size = 211, normalized size = 1.66

$$\frac{C\left(\tan^2(dx+c)\right)}{2bd}+\frac{B\tan(dx+c)}{bd}-\frac{C\tan(dx+c)\,a}{d\,b^2}-\frac{a^3B\ln(a+b\tan(dx+c))}{b^2\left(a^2+b^2\right)d}+\frac{a^4\ln(a+b\tan(dx+c))\,C}{d\,b^3\left(a^2+b^2\right)}-\frac{\ln}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x)
```

```
[Out] 1/2*C*tan(d*x+c)^2/b/d+B*tan(d*x+c)/b/d-1/d/b^2*C*tan(d*x+c)*a-a^3*B*ln(a+b
*tan(d*x+c))/b^2/(a^2+b^2)/d+1/d/b^3*a^4/(a^2+b^2)*ln(a+b*tan(d*x+c))*C-1/2
/d/(a^2+b^2)*ln(1+tan(d*x+c)^2)*a*B-1/2/d/(a^2+b^2)*ln(1+tan(d*x+c)^2)*C*b-
1/d/(a^2+b^2)*B*arctan(tan(d*x+c))*b+1/d/(a^2+b^2)*C*arctan(tan(d*x+c))*a
```

maxima [A] time = 0.72, size = 130, normalized size = 1.02

$$\frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}+\frac{2\left(Ca^4-Ba^3b\right)\log(b\tan(dx+c)+a)}{a^2b^3+b^5}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{Cb\tan(dx+c)^2-2\,(Ca-Bb)\tan(dx+c)}{b^2}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="maxima")
```

```
[Out] 1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) + 2*(C*a^4 - B*a^3*b)*log(b*tan(d*
x + c) + a)/(a^2*b^3 + b^5) - (B*a + C*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b^
2) + (C*b*tan(d*x + c)^2 - 2*(C*a - B*b)*tan(d*x + c))/b^2)/d
```

mupad [B] time = 9.07, size = 144, normalized size = 1.13

$$\frac{\tan(c+dx)\left(\frac{B}{b}-\frac{Ca}{b^2}\right)}{d}-\frac{\ln(\tan(c+dx)-\mathrm{i})\,(-C+B\,1\mathrm{i})}{2d\,(-b+a\,1\mathrm{i})}+\frac{\ln(a+b\tan(c+dx))\,\left(C\,a^4-B\,a^3\,b\right)}{d\,\left(a^2\,b^3+b^5\right)}-\frac{\ln(\tan(c+dx)}{2d\,(}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((tan(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
)),x)
```

```
[Out] (tan(c + d*x)*(B/b - (C*a)/b^2))/d - (log(tan(c + d*x) - 1i)*(B*1i - C))/(2
*d*(a*1i - b)) + (log(a + b*tan(c + d*x))*(C*a^4 - B*a^3*b))/(d*(b^5 + a^2*
b^3)) - (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a - b*1i)) + (C*tan(c + d
*x)^2)/(2*b*d)
```

sympy [A] time = 2.03, size = 1309, normalized size = 10.31

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)*tan(c), Eq(a, 0) & Eq(b, 0) & Eq(
d, 0)), (-3*I*B*d*x*tan(c + d*x)/(2*I*b*d*tan(c + d*x) + 2*b*d) - 3*B*d*x/(
2*I*b*d*tan(c + d*x) + 2*b*d) - B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*
I*b*d*tan(c + d*x) + 2*b*d) + I*B*log(tan(c + d*x)**2 + 1)/(2*I*b*d*tan(c +
 d*x) + 2*b*d) + 2*I*B*tan(c + d*x)**2/(2*I*b*d*tan(c + d*x) + 2*b*d) + 3*I
*B/(2*I*b*d*tan(c + d*x) + 2*b*d) + 3*C*d*x*tan(c + d*x)/(2*I*b*d*tan(c + d
*x) + 2*b*d) - 3*I*C*d*x/(2*I*b*d*tan(c + d*x) + 2*b*d) - 2*I*C*log(tan(c +
 d*x)**2 + 1)*tan(c + d*x)/(2*I*b*d*tan(c + d*x) + 2*b*d) - 2*C*log(tan(c +
 d*x)**2 + 1)/(2*I*b*d*tan(c + d*x) + 2*b*d) + I*C*tan(c + d*x)**3/(2*I*b*d
*tan(c + d*x) + 2*b*d) - C*tan(c + d*x)**2/(2*I*b*d*tan(c + d*x) + 2*b*d) -
 3*C/(2*I*b*d*tan(c + d*x) + 2*b*d), Eq(a, -I*b)), (3*I*B*d*x*tan(c + d*x)/
(-2*I*b*d*tan(c + d*x) + 2*b*d) - 3*B*d*x/(-2*I*b*d*tan(c + d*x) + 2*b*d) -
 B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(-2*I*b*d*tan(c + d*x) + 2*b*d) -
I*B*log(tan(c + d*x)**2 + 1)/(-2*I*b*d*tan(c + d*x) + 2*b*d) - 2*I*B*tan(c
+ d*x)**2/(-2*I*b*d*tan(c + d*x) + 2*b*d) - 3*I*B/(-2*I*b*d*tan(c + d*x) +
2*b*d) + 3*C*d*x*tan(c + d*x)/(-2*I*b*d*tan(c + d*x) + 2*b*d) + 3*I*C*d*x/(
```

```
-2*I*b*d*tan(c + d*x) + 2*b*d) + 2*I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x
)/(-2*I*b*d*tan(c + d*x) + 2*b*d) - 2*C*log(tan(c + d*x)**2 + 1)/(-2*I*b*d*
tan(c + d*x) + 2*b*d) - I*C*tan(c + d*x)**3/(-2*I*b*d*tan(c + d*x) + 2*b*d)
 - C*tan(c + d*x)**2/(-2*I*b*d*tan(c + d*x) + 2*b*d) - 3*C/(-2*I*b*d*tan(c
+ d*x) + 2*b*d), Eq(a, I*b)), ((-B*log(tan(c + d*x)**2 + 1)/(2*d) + B*tan(c
 + d*x)**2/(2*d) + C*x + C*tan(c + d*x)**3/(3*d) - C*tan(c + d*x)/d)/a, Eq(
b, 0)), (x*(B*tan(c) + C*tan(c)**2)*tan(c)**2/(a + b*tan(c)), Eq(d, 0)), (-
2*B*a**3*b*log(a/b + tan(c + d*x))/(2*a**2*b**3*d + 2*b**5*d) + 2*B*a**2*b*
*2*tan(c + d*x)/(2*a**2*b**3*d + 2*b**5*d) - B*a*b**3*log(tan(c + d*x)**2 +
 1)/(2*a**2*b**3*d + 2*b**5*d) - 2*B*b**4*d*x/(2*a**2*b**3*d + 2*b**5*d) +
2*B*b**4*tan(c + d*x)/(2*a**2*b**3*d + 2*b**5*d) + 2*C*a**4*log(a/b + tan(c
 + d*x))/(2*a**2*b**3*d + 2*b**5*d) - 2*C*a**3*b*tan(c + d*x)/(2*a**2*b**3*
d + 2*b**5*d) + C*a**2*b**2*tan(c + d*x)**2/(2*a**2*b**3*d + 2*b**5*d) + 2*
C*a*b**3*d*x/(2*a**2*b**3*d + 2*b**5*d) - 2*C*a*b**3*tan(c + d*x)/(2*a**2*b
**3*d + 2*b**5*d) - C*b**4*log(tan(c + d*x)**2 + 1)/(2*a**2*b**3*d + 2*b**5
*d) + C*b**4*tan(c + d*x)**2/(2*a**2*b**3*d + 2*b**5*d), True))
```

3.26 $\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$

Optimal. Leaf size=101

$$\frac{a^2(bB-aC)\log(a+b\tan(c+dx))}{b^2d\left(a^2+b^2\right)}-\frac{(bB-aC)\log(\cos(c+dx))}{d\left(a^2+b^2\right)}-\frac{x(aB+bC)}{a^2+b^2}+\frac{C\tan(c+dx)}{bd}$$

```
[Out] -(B*a+C*b)*x/(a^2+b^2)-(B*b-C*a)*ln(cos(d*x+c))/(a^2+b^2)/d+a^2*(B*b-C*a)*l
n(a+b*tan(d*x+c))/b^2/(a^2+b^2)/d+C*tan(d*x+c)/b/d
```

Rubi [A] time = 0.24, antiderivative size = 101, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.158, Rules used = {3632, 3606, 3626, 3617, 31, 3475}

$$\frac{a^2(bB-aC)\log(a+b\tan(c+dx))}{b^2d\left(a^2+b^2\right)}-\frac{(bB-aC)\log(\cos(c+dx))}{d\left(a^2+b^2\right)}-\frac{x(aB+bC)}{a^2+b^2}+\frac{C\tan(c+dx)}{bd}$$

Antiderivative was successfully verified.

```
[In] Int[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
,x]
```

```
[Out] -(((a*B + b*C)*x)/(a^2 + b^2)) - ((b*B - a*C)*Log[Cos[c + d*x]])/((a^2 + b^
2)*d) + (a^2*(b*B - a*C)*Log[a + b*Tan[c + d*x]])/(b^2*(a^2 + b^2)*d) + (C*
Tan[c + d*x])/(b*d)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3606

```
Int[(((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (
f_.)*(x_)]))/((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[(b^
2*B*Tan[e + f*x])/(d*f), x] + Dist[1/d, Int[(a^2*A*d - b^2*B*c + (2*a*A*b +
 B*(a^2 - b^2))*d*Tan[e + f*x] + (A*b^2*d - b*B*(b*c - 2*a*d))*Tan[e + f*x]
^2)/(c + d*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && N
```

```
eQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \int \frac{\tan^2(c+dx)(B+C\tan(c+dx))}{a+b\tan(c+dx)}\,dx$$

$$= \frac{C\tan(c+dx)}{bd} + \frac{\int \frac{-aC-bC\tan(c+dx)+(bB-aC)\tan^2(c+dx)}{a+b\tan(c+dx)}dx}{b}$$

$$= -\frac{(aB+bC)x}{a^2+b^2} + \frac{C\tan(c+dx)}{bd} + \frac{(bB-aC)\int\tan(c+dx)\,d$$

$$= -\frac{(aB+bC)x}{a^2+b^2} - \frac{(bB-aC)\log(\cos(c+dx))}{\left(a^2+b^2\right)d} + \frac{C\tan(c+dx$$

$$= -\frac{(aB+bC)x}{a^2+b^2} - \frac{(bB-aC)\log(\cos(c+dx))}{\left(a^2+b^2\right)d} + \frac{a^2(bB-aC)}{b^2$$

Mathematica [C] time = 0.70, size = 118, normalized size = 1.17

$$\frac{\frac{2a^2(bB-aC)\log(a+b\tan(c+dx))}{b^2\left(a^2+b^2\right)} + \frac{i(B+iC)\log(-\tan(c+dx)+i)}{a+ib} - \frac{(C+iB)\log(\tan(c+dx)+i)}{a-ib} + \frac{2C\tan(c+dx)}{b}}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x]),x]
```

```
[Out] ((I*(B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b) - ((I*B + C)*Log[I + Tan[c +
 d*x]])/(a - I*b) + (2*a^2*(b*B - a*C)*Log[a + b*Tan[c + d*x]])/(b^2*(a^2 +
 b^2)) + (2*C*Tan[c + d*x])/b)/(2*d)
```

fricas [A] time = 0.67, size = 149, normalized size = 1.48

$$-\frac{2\left(Bab^2+Cb^3\right)dx + \left(Ca^3-Ba^2b\right)\log\left(\frac{b^2\tan(dx+c)^2+2ab\tan(dx+c)+a^2}{\tan(dx+c)^2+1}\right) - \left(Ca^3-Ba^2b+Cab^2-Bb^3\right)\log\left(\frac{1}{\tan(dx+c)^2+}\right.}{2\left(a^2b^2+b^4\right)d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="fricas")
```

```
[Out] -1/2*(2*(B*a*b^2 + C*b^3)*d*x + (C*a^3 - B*a^2*b)*log((b^2*tan(d*x + c)^2 +
 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1)) - (C*a^3 - B*a^2*b + C*a*b
```

```
^2 - B*b^3)*log(1/(tan(d*x + c)^2 + 1)) - 2*(C*a^2*b + C*b^3)*tan(d*x + c))
/((a^2*b^2 + b^4)*d)
```

giac [A] time = 1.81, size = 110, normalized size = 1.09

$$-\frac{\frac{2\,(Ba+Cb)(dx+c)}{a^2+b^2}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{2\left(Ca^3-Ba^2b\right)\log(|b\tan(dx+c)+a|)}{a^2b^2+b^4}-\frac{2\,C\tan(dx+c)}{b}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="giac")
```

```
[Out] -1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) + (C*a - B*b)*log(tan(d*x + c)^2
+ 1)/(a^2 + b^2) + 2*(C*a^3 - B*a^2*b)*log(abs(b*tan(d*x + c) + a))/(a^2*b^
2 + b^4) - 2*C*tan(d*x + c)/b)/d
```

maple [A] time = 0.26, size = 179, normalized size = 1.77

$$\frac{C\tan(dx+c)}{bd}+\frac{a^2\ln(a+b\tan(dx+c))\,B}{db\left(a^2+b^2\right)}-\frac{a^3\ln(a+b\tan(dx+c))\,C}{d\,b^2\left(a^2+b^2\right)}+\frac{\ln\left(1+\tan^2(dx+c)\right)Bb}{2d\left(a^2+b^2\right)}-\frac{\ln\left(1+\tan^2\right.}{2d\left(a\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x)
```

```
[Out] C*tan(d*x+c)/b/d+1/d/b*a^2/(a^2+b^2)*ln(a+b*tan(d*x+c))*B-1/d/b^2*a^3/(a^2+
b^2)*ln(a+b*tan(d*x+c))*C+1/2/d/(a^2+b^2)*ln(1+tan(d*x+c)^2)*B*b-1/2/d/(a^2
+b^2)*ln(1+tan(d*x+c)^2)*a*C-1/d/(a^2+b^2)*B*arctan(tan(d*x+c))*a-1/d/(a^2+
b^2)*C*arctan(tan(d*x+c))*b
```

maxima [A] time = 0.55, size = 109, normalized size = 1.08

$$-\frac{\frac{2\,(Ba+Cb)(dx+c)}{a^2+b^2}+\frac{2\left(Ca^3-Ba^2b\right)\log(b\tan(dx+c)+a)}{a^2b^2+b^4}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}-\frac{2\,C\tan(dx+c)}{b}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="maxima")
```

```
[Out] -1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) + 2*(C*a^3 - B*a^2*b)*log(b*tan(d
*x + c) + a)/(a^2*b^2 + b^4) + (C*a - B*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b
^2) - 2*C*tan(d*x + c)/b)/d
```

mupad [B] time = 8.77, size = 117, normalized size = 1.16

$$\frac{C\tan(c+dx)}{bd}+\frac{\ln(\tan(c+dx)+1\mathrm{i})\,(B-C\,1\mathrm{i})}{2d\,(b+a\,1\mathrm{i})}-\frac{\ln(a+b\tan(c+dx))\,\left(C\,a^3-B\,a^2\,b\right)}{d\,\left(a^2\,b^2+b^4\right)}+\frac{\ln(\tan(c+dx)-\mathrm{i})\,(-$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((tan(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a*1i + b)) - (log(a + b*tan(c + d
*x))*(C*a^3 - B*a^2*b))/(d*(b^4 + a^2*b^2)) + (C*tan(c + d*x))/(b*d) + (log
(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a + b*1i))
```

sympy [A] time = 1.45, size = 1020, normalized size = 10.10

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2), Eq(a, 0) & Eq(b, 0) & Eq(d, 0)),
 (I*B*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) + B*d*x/(2*b*d*tan(c
+ d*x) - 2*I*b*d) + B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c +
d*x) - 2*I*b*d) - I*B*log(tan(c + d*x)**2 + 1)/(2*b*d*tan(c + d*x) - 2*I*b*
d) - I*B/(2*b*d*tan(c + d*x) - 2*I*b*d) - 3*C*d*x*tan(c + d*x)/(2*b*d*tan(c
 + d*x) - 2*I*b*d) + 3*I*C*d*x/(2*b*d*tan(c + d*x) - 2*I*b*d) + I*C*log(tan
(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) + C*log(tan(c
 + d*x)**2 + 1)/(2*b*d*tan(c + d*x) - 2*I*b*d) + 2*C*tan(c + d*x)**2/(2*b*d
*tan(c + d*x) - 2*I*b*d) + 3*C/(2*b*d*tan(c + d*x) - 2*I*b*d), Eq(a, -I*b))
, (-I*B*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d) + B*d*x/(2*b*d*tan(
c + d*x) + 2*I*b*d) + B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c
+ d*x) + 2*I*b*d) + I*B*log(tan(c + d*x)**2 + 1)/(2*b*d*tan(c + d*x) + 2*I*
b*d) + I*B/(2*b*d*tan(c + d*x) + 2*I*b*d) - 3*C*d*x*tan(c + d*x)/(2*b*d*tan
(c + d*x) + 2*I*b*d) - 3*I*C*d*x/(2*b*d*tan(c + d*x) + 2*I*b*d) - I*C*log(t
an(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d) + C*log(tan
(c + d*x)**2 + 1)/(2*b*d*tan(c + d*x) + 2*I*b*d) + 2*C*tan(c + d*x)**2/(2*b
*d*tan(c + d*x) + 2*I*b*d) + 3*C/(2*b*d*tan(c + d*x) + 2*I*b*d), Eq(a, I*b)
), ((-B*x + B*tan(c + d*x)/d - C*log(tan(c + d*x)**2 + 1)/(2*d) + C*tan(c +
 d*x)**2/(2*d))/a, Eq(b, 0)), (x*(B*tan(c) + C*tan(c)**2)*tan(c)/(a + b*tan
(c)), Eq(d, 0)), (2*B*a**2*b*log(a/b + tan(c + d*x))/(2*a**2*b**2*d + 2*b**
4*d) - 2*B*a*b**2*d*x/(2*a**2*b**2*d + 2*b**4*d) + B*b**3*log(tan(c + d*x)*
*2 + 1)/(2*a**2*b**2*d + 2*b**4*d) - 2*C*a**3*log(a/b + tan(c + d*x))/(2*a*
*2*b**2*d + 2*b**4*d) + 2*C*a**2*b*tan(c + d*x)/(2*a**2*b**2*d + 2*b**4*d)
```

```
- C*a*b**2*log(tan(c + d*x)**2 + 1)/(2*a**2*b**2*d + 2*b**4*d) - 2*C*b**3*d
*x/(2*a**2*b**2*d + 2*b**4*d) + 2*C*b**3*tan(c + d*x)/(2*a**2*b**2*d + 2*b*
*4*d), True))
```

3.27 $\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx$

Optimal. Leaf size=85

$$-\frac{a(bB-aC)\log(a+b\tan(c+dx))}{bd\left(a^2+b^2\right)}-\frac{(aB+bC)\log(\cos(c+dx))}{d\left(a^2+b^2\right)}+\frac{x(bB-aC)}{a^2+b^2}$$

```
[Out] (B*b-C*a)*x/(a^2+b^2)-(B*a+C*b)*ln(cos(d*x+c))/(a^2+b^2)/d-a*(B*b-C*a)*ln(a
+b*tan(d*x+c))/b/(a^2+b^2)/d
```

Rubi [A] time = 0.16, antiderivative size = 85, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 4, integrand size = 32, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {1629, 635, 203, 260}

$$-\frac{a(bB-aC)\log(a+b\tan(c+dx))}{bd\left(a^2+b^2\right)}-\frac{(aB+bC)\log(\cos(c+dx))}{d\left(a^2+b^2\right)}+\frac{x(bB-aC)}{a^2+b^2}$$

Antiderivative was successfully verified.

```
[In] Int[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x]),x]
```

```
[Out] ((b*B - a*C)*x)/(a^2 + b^2) - ((a*B + b*C)*Log[Cos[c + d*x]])/((a^2 + b^2)*
d) - (a*(b*B - a*C)*Log[a + b*Tan[c + d*x]])/(b*(a^2 + b^2)*d)
```

Rule 203

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTan[(Rt[b, 2]*x)/Rt
[a, 2]])/(Rt[a, 2]*Rt[b, 2]), x] /; FreeQ[{a, b}, x] && PosQ[a/b] && (GtQ[a
, 0] || GtQ[b, 0])
```

Rule 260

```
Int[(x_)^(m_.)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> Simp[Log[RemoveConten
t[a + b*x^n, x]]/(b*n), x] /; FreeQ[{a, b, m, n}, x] && EqQ[m, n - 1]
```

Rule 635

```
Int[((d_) + (e_.)*(x_))/((a_) + (c_.)*(x_)^2), x_Symbol] :> Dist[d, Int[1/(
a + c*x^2), x], x] + Dist[e, Int[x/(a + c*x^2), x], x] /; FreeQ[{a, c, d, e
}, x] && !NiceSqrtQ[-(a*c)]
```

Rule 1629

```
Int[(Pq_)*((d_) + (e_.)*(x_))^(m_.)*((a_) + (c_.)*(x_)^2)^(p_.), x_Symbol]
:> Int[ExpandIntegrand[(d + e*x)^m*Pq*(a + c*x^2)^p, x], x] /; FreeQ[{a, c,
```

```
d, e, m}, x] && PolyQ[Pq, x] && IGtQ[p, -2]
```

Rubi steps

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{a+b\tan(c+dx)}\,dx = \frac{\text{Subst}\left(\int \frac{x(B+Cx)}{(a+bx)(1+x^2)}\,dx, x, \tan(c+dx)\right)}{d}$$
$$= \frac{\text{Subst}\left(\int \left(\frac{a(-bB+aC)}{(a^2+b^2)(a+bx)}+\frac{bB-aC+(aB+bC)x}{(a^2+b^2)(1+x^2)}\right)dx, x, \tan(c+dx)\right)}{d}$$
$$= -\frac{a(bB-aC)\log(a+b\tan(c+dx))}{b(a^2+b^2)d}+\frac{\text{Subst}\left(\int \frac{bB-aC+(aB+bC)x}{1+x^2}\,dx, x, \text{ta}$$
$$= -\frac{a(bB-aC)\log(a+b\tan(c+dx))}{b(a^2+b^2)d}+\frac{(bB-aC)\,\text{Subst}\left(\int \frac{1}{1+x^2}\,dx, x, \text{ta}$$
$$= \frac{(bB-aC)x}{a^2+b^2}-\frac{(aB+bC)\log(\cos(c+dx))}{(a^2+b^2)d}-\frac{a(bB-aC)\log(a+b\tan(c}{b(a^2+b^2)d}$$

Mathematica [C] time = 0.20, size = 98, normalized size = 1.15

$$\frac{b(a-ib)(B+iC)\log(-\tan(c+dx)+i)+b(a+ib)(B-iC)\log(\tan(c+dx)+i)+2a(aC-bB)\log(a+b\tan(c}{2bd(a^2+b^2)}$$

Antiderivative was successfully verified.

```
[In] Integrate[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x]),x]
```

```
[Out] ((a - I*b)*b*(B + I*C)*Log[I - Tan[c + d*x]] + (a + I*b)*b*(B - I*C)*Log[I
+ Tan[c + d*x]] + 2*a*(-(b*B) + a*C)*Log[a + b*Tan[c + d*x]])/(2*b*(a^2 + b
^2)*d)
```

fricas [A] time = 2.31, size = 110, normalized size = 1.29

$$-\frac{2\left(Cab-Bb^2\right)dx-\left(Ca^2-Bab\right)\log\left(\frac{b^2\tan(dx+c)^2+2\,ab\tan(dx+c)+a^2}{\tan(dx+c)^2+1}\right)+\left(Ca^2+Cb^2\right)\log\left(\frac{1}{\tan(dx+c)^2+1}\right)}{2\left(a^2b+b^3\right)d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algorithm="fric
as")
```

```
[Out] -1/2*(2*(C*a*b - B*b^2)*d*x - (C*a^2 - B*a*b)*log((b^2*tan(d*x + c)^2 + 2*a
*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1)) + (C*a^2 + C*b^2)*log(1/(tan(d
*x + c)^2 + 1)))/((a^2*b + b^3)*d)
```

giac [A] time = 1.52, size = 95, normalized size = 1.12

$$-\frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}-\frac{2\left(Ca^2-Bab\right)\log(|b\tan(dx+c)+a|)}{a^2b+b^3}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algorithm="giac
")
```

```
[Out] -1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - (B*a + C*b)*log(tan(d*x + c)^2
+ 1)/(a^2 + b^2) - 2*(C*a^2 - B*a*b)*log(abs(b*tan(d*x + c) + a))/(a^2*b +
b^3))/d
```

maple [A] time = 0.28, size = 159, normalized size = 1.87

$$-\frac{a\ln\left(a+b\tan\left(dx+c\right)\right)B}{d\left(a^2+b^2\right)}+\frac{a^2\ln\left(a+b\tan\left(dx+c\right)\right)C}{d\left(a^2+b^2\right)b}+\frac{\ln\left(1+\tan^2\left(dx+c\right)\right)aB}{2d\left(a^2+b^2\right)}+\frac{\ln\left(1+\tan^2\left(dx+c\right)\right)Cb}{2d\left(a^2+b^2\right)}+\frac{B}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x)
```

```
[Out] -1/d*a/(a^2+b^2)*ln(a+b*tan(d*x+c))*B+1/d*a^2/(a^2+b^2)/b*ln(a+b*tan(d*x+c)
)*C+1/2/d/(a^2+b^2)*ln(1+tan(d*x+c)^2)*a*B+1/2/d/(a^2+b^2)*ln(1+tan(d*x+c)^
2)*C*b+1/d/(a^2+b^2)*B*arctan(tan(d*x+c))*b-1/d/(a^2+b^2)*C*arctan(tan(d*x+
c))*a
```

maxima [A] time = 0.77, size = 94, normalized size = 1.11

$$-\frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}-\frac{2\left(Ca^2-Bab\right)\log(b\tan(dx+c)+a)}{a^2b+b^3}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algorithm="maxi
ma")
```

```
[Out] -1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - 2*(C*a^2 - B*a*b)*log(b*tan(d*x
 + c) + a)/(a^2*b + b^3) - (B*a + C*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b^2))
/d
```

mupad [B] time = 9.07, size = 100, normalized size = 1.18

$$\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-b+a\,1\mathrm{i}\right)}+\frac{\ln\left(\tan\left(c+d\,x\right)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(a-b\,1\mathrm{i}\right)}-\frac{a\,\ln\left(a+b\tan\left(c+d\,x\right)\right)\,\left(B\,b-C\,a\right)}{b\,d\,\left(a^2+b^2\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)/(a + b*tan(c + d*x)),x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*1i - b)) + (log(tan(c + d*x) +
1i)*(B - C*1i))/(2*d*(a - b*1i)) - (a*log(a + b*tan(c + d*x))*(B*b - C*a))/
(b*d*(a^2 + b^2))
```

sympy [A] time = 1.12, size = 724, normalized size = 8.52

$$\begin{cases} \frac{\tilde{\infty} x\left(B\tan(c)+C\tan^2(c)\right)}{\tan(c)} \\ -\frac{Bdx\tan(c+dx)}{-2bd\tan(c+dx)+2ibd}+\frac{iBdx}{-2bd\tan(c+dx)+2ibd}+\frac{B}{-2bd\tan(c+dx)+2ibd}-\frac{iCdx\tan(c+dx)}{-2bd\tan(c+dx)+2ibd}-\frac{Cdx}{-2bd\tan(c+dx)+2ibd}-\frac{C\log\left(\tan^2(c\cdots\right.}{-2bd\tan\cdots} \\ -\frac{Bdx\tan(c+dx)}{-2bd\tan(c+dx)-2ibd}-\frac{iBdx}{-2bd\tan(c+dx)-2ibd}+\frac{B}{-2bd\tan(c+dx)-2ibd}+\frac{iCdx\tan(c+dx)}{-2bd\tan(c+dx)-2ibd}-\frac{Cdx}{-2bd\tan(c+dx)-2ibd}-\frac{C\log\left(\tan^2(c+\cdots\right.}{-2bd\tan\cdots} \\ \frac{\frac{B\log\left(\tan^2(c+dx)+1\right)}{2d}-Cx+\frac{C\tan(c+dx)}{d}}{a} \\ \frac{x\left(B\tan(c)+C\tan^2(c)\right)}{a+b\tan(c)} \\ -\frac{2Bab\log\left(\frac{a}{b}+\tan(c+dx)\right)}{2a^2bd+2b^3d}+\frac{Bab\log\left(\tan^2(c+dx)+1\right)}{2a^2bd+2b^3d}+\frac{2Bb^2dx}{2a^2bd+2b^3d}+\frac{2Ca^2\log\left(\frac{a}{b}+\tan(c+dx)\right)}{2a^2bd+2b^3d}-\frac{2Cabdx}{2a^2bd+2b^3d}+\frac{Cb^2\log\left(\tan^2(c+dx)+1\right)}{2a^2bd+2b^3d} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)/tan(c), Eq(a, 0) & Eq(b, 0) & Eq(
d, 0)), (-B*d*x*tan(c + d*x)/(-2*b*d*tan(c + d*x) + 2*I*b*d) + I*B*d*x/(-2*
b*d*tan(c + d*x) + 2*I*b*d) + B/(-2*b*d*tan(c + d*x) + 2*I*b*d) - I*C*d*x*t
an(c + d*x)/(-2*b*d*tan(c + d*x) + 2*I*b*d) - C*d*x/(-2*b*d*tan(c + d*x) +
2*I*b*d) - C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(-2*b*d*tan(c + d*x) + 2
*I*b*d) + I*C*log(tan(c + d*x)**2 + 1)/(-2*b*d*tan(c + d*x) + 2*I*b*d) + I*
C/(-2*b*d*tan(c + d*x) + 2*I*b*d), Eq(a, -I*b)), (-B*d*x*tan(c + d*x)/(-2*b
```

```
*d*tan(c + d*x) - 2*I*b*d) - I*B*d*x/(-2*b*d*tan(c + d*x) - 2*I*b*d) + B/(-
2*b*d*tan(c + d*x) - 2*I*b*d) + I*C*d*x*tan(c + d*x)/(-2*b*d*tan(c + d*x) -
 2*I*b*d) - C*d*x/(-2*b*d*tan(c + d*x) - 2*I*b*d) - C*log(tan(c + d*x)**2 +
 1)*tan(c + d*x)/(-2*b*d*tan(c + d*x) - 2*I*b*d) - I*C*log(tan(c + d*x)**2
+ 1)/(-2*b*d*tan(c + d*x) - 2*I*b*d) - I*C/(-2*b*d*tan(c + d*x) - 2*I*b*d),
 Eq(a, I*b)), ((B*log(tan(c + d*x)**2 + 1)/(2*d) - C*x + C*tan(c + d*x)/d)/
a, Eq(b, 0)), (x*(B*tan(c) + C*tan(c)**2)/(a + b*tan(c)), Eq(d, 0)), (-2*B*
a*b*log(a/b + tan(c + d*x))/(2*a**2*b*d + 2*b**3*d) + B*a*b*log(tan(c + d*x
)**2 + 1)/(2*a**2*b*d + 2*b**3*d) + 2*B*b**2*d*x/(2*a**2*b*d + 2*b**3*d) +
2*C*a**2*log(a/b + tan(c + d*x))/(2*a**2*b*d + 2*b**3*d) - 2*C*a*b*d*x/(2*a
**2*b*d + 2*b**3*d) + C*b**2*log(tan(c + d*x)**2 + 1)/(2*a**2*b*d + 2*b**3*
d), True))
```

3.28 $\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$

Optimal. Leaf size=58

$$\frac{(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{d\left(a^2+b^2\right)}+\frac{x(aB+bC)}{a^2+b^2}$$

```
[Out] (B*a+C*b)*x/(a^2+b^2)+(B*b-C*a)*ln(a*cos(d*x+c)+b*sin(d*x+c))/(a^2+b^2)/d
```

Rubi [A] time = 0.14, antiderivative size = 58, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.079, Rules used = {3632, 3531, 3530}

$$\frac{(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{d\left(a^2+b^2\right)}+\frac{x(aB+bC)}{a^2+b^2}$$

Antiderivative was successfully verified.

```
[In] Int[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
,x]
```

```
[Out] ((a*B + b*C)*x)/(a^2 + b^2) + ((b*B - a*C)*Log[a*Cos[c + d*x] + b*Sin[c + d
*x]])/((a^2 + b^2)*d)
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
```

```
1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \int \frac{B+C\tan(c+dx)}{a+b\tan(c+dx)}\,dx$$
$$= \frac{(aB+bC)x}{a^2+b^2} + \frac{(bB-aC)\int \frac{b-a\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{a^2+b^2}$$
$$= \frac{(aB+bC)x}{a^2+b^2} + \frac{(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{\left(a^2+b^2\right)d}$$

Mathematica [A] time = 0.13, size = 67, normalized size = 1.16

$$\frac{(bB-aC)\left(2\log(a\cot(c+dx)+b)-\log\left(\csc^2(c+dx)\right)\right)-2(aB+bC)\tan^{-1}(\cot(c+dx))}{2d\left(a^2+b^2\right)}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x]),x]
```

```
[Out] (-2*(a*B + b*C)*ArcTan[Cot[c + d*x]] + (b*B - a*C)*(2*Log[b + a*Cot[c + d*x
]] - Log[Csc[c + d*x]^2]))/(2*(a^2 + b^2)*d)
```

fricas [A] time = 0.75, size = 76, normalized size = 1.31

$$\frac{2\,(Ba+Cb)dx-(Ca-Bb)\log\left(\frac{b^2\tan(dx+c)^2+2\,ab\tan(dx+c)+a^2}{\tan(dx+c)^2+1}\right)}{2\left(a^2+b^2\right)d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="fricas")
```

```
[Out] 1/2*(2*(B*a + C*b)*d*x - (C*a - B*b)*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*
x + c) + a^2)/(tan(d*x + c)^2 + 1)))/((a^2 + b^2)*d)
```

giac [A] time = 2.15, size = 94, normalized size = 1.62

$$\frac{\frac{2\,(Ba+Cb)(dx+c)}{a^2+b^2}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}-\frac{2\left(Cab-Bb^2\right)\log(|b\tan(dx+c)+a|)}{a^2b+b^3}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="giac")
```

```
[Out] 1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) + (C*a - B*b)*log(tan(d*x + c)^2 +
 1)/(a^2 + b^2) - 2*(C*a*b - B*b^2)*log(abs(b*tan(d*x + c) + a))/(a^2*b + b
^3))/d
```

maple [B] time = 0.73, size = 153, normalized size = 2.64

$$\frac{\ln\left(a+b\tan\left(dx+c\right)\right)Bb}{d\left(a^2+b^2\right)}-\frac{\ln\left(a+b\tan\left(dx+c\right)\right)aC}{d\left(a^2+b^2\right)}-\frac{\ln\left(1+\tan^2\left(dx+c\right)\right)Bb}{2d\left(a^2+b^2\right)}+\frac{\ln\left(1+\tan^2\left(dx+c\right)\right)aC}{2d\left(a^2+b^2\right)}+\frac{B\,\mathrm{ar}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x)
```

```
[Out] 1/d/(a^2+b^2)*ln(a+b*tan(d*x+c))*B*b-1/d/(a^2+b^2)*ln(a+b*tan(d*x+c))*a*C-1
/2/d/(a^2+b^2)*ln(1+tan(d*x+c)^2)*B*b+1/2/d/(a^2+b^2)*ln(1+tan(d*x+c)^2)*a*
C+1/d/(a^2+b^2)*B*arctan(tan(d*x+c))*a+1/d/(a^2+b^2)*C*arctan(tan(d*x+c))*b
```

maxima [A] time = 0.63, size = 88, normalized size = 1.52

$$\frac{\frac{2\,(Ba+Cb)(dx+c)}{a^2+b^2}-\frac{2\,(Ca-Bb)\log(b\tan(dx+c)+a)}{a^2+b^2}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, algo
rithm="maxima")
```

```
[Out] 1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) - 2*(C*a - B*b)*log(b*tan(d*x + c)
 + a)/(a^2 + b^2) + (C*a - B*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b^2))/d
```

mupad [B] time = 9.12, size = 93, normalized size = 1.60

$$\frac{\ln\left(a+b\tan\left(c+d\,x\right)\right)\,\left(B\,b-C\,a\right)}{d\,\left(a^2+b^2\right)}-\frac{\ln\left(\tan\left(c+d\,x\right)+1{}\mathrm{i}\right)\,\left(B-C\,1{}\mathrm{i}\right)}{2\,d\,\left(b+a\,1{}\mathrm{i}\right)}-\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(-C+B\,1{}\mathrm{i}\right)}{2\,d\,\left(a+b\,1{}\mathrm{i}\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
,x)
```

```
[Out] (log(a + b*tan(c + d*x))*(B*b - C*a))/(d*(a^2 + b^2)) - (log(tan(c + d*x) +
 1i)*(B - C*1i))/(2*d*(a*1i + b)) - (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*
d*(a + b*1i))
```

sympy [A] time = 2.95, size = 541, normalized size = 9.33

$$\begin{cases} \frac{\tilde{\infty} x \left(B \tan{\left(c \right)} + C \tan^{2}{\left(c \right)}\right) \cot{\left(c \right)}}{\tan{\left(c \right)}} \\ \frac{i B d x \tan{\left(c + d x \right)}}{2 b d \tan{\left(c + d x \right)} - 2 i b d} + \frac{B d x}{2 b d \tan{\left(c + d x \right)} - 2 i b d} + \frac{i B}{2 b d \tan{\left(c + d x \right)} - 2 i b d} + \frac{C d x \tan{\left(c + d x \right)}}{2 b d \tan{\left(c + d x \right)} - 2 i b d} - \frac{i C d x}{2 b d \tan{\left(c + d x \right)} - 2 i b d} - \frac{C}{2 b d \tan{\left(c + d x \right)} - 2 i b d} \\ - \frac{i B d x \tan{\left(c + d x \right)}}{2 b d \tan{\left(c + d x \right)} + 2 i b d} + \frac{B d x}{2 b d \tan{\left(c + d x \right)} + 2 i b d} - \frac{i B}{2 b d \tan{\left(c + d x \right)} + 2 i b d} + \frac{C d x \tan{\left(c + d x \right)}}{2 b d \tan{\left(c + d x \right)} + 2 i b d} + \frac{i C d x}{2 b d \tan{\left(c + d x \right)} + 2 i b d} - \frac{C}{2 b d \tan{\left(c + d x \right)} + 2 i b d} \\ \frac{x \left(B \tan{\left(c \right)} + C \tan^{2}{\left(c \right)}\right) \cot{\left(c \right)}}{a + b \tan{\left(c \right)}} \\ \frac{B x + \frac{C \log{\left(\tan^{2}{\left(c + d x \right)} + 1 \right)}}{2 d}}{a} \\ \frac{2 B a d x}{2 a^{2} d + 2 b^{2} d} + \frac{2 B b \log{\left(\frac{a}{b} + \tan{\left(c + d x \right)} \right)}}{2 a^{2} d + 2 b^{2} d} - \frac{B b \log{\left(\tan^{2}{\left(c + d x \right)} + 1 \right)}}{2 a^{2} d + 2 b^{2} d} - \frac{2 C a \log{\left(\frac{a}{b} + \tan{\left(c + d x \right)} \right)}}{2 a^{2} d + 2 b^{2} d} + \frac{C a \log{\left(\tan^{2}{\left(c + d x \right)} + 1 \right)}}{2 a^{2} d + 2 b^{2} d} + \frac{2 C b d x}{2 a^{2} d + 2 b^{2} d} \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)*cot(c)/tan(c), Eq(a, 0) & Eq(b, 0
) & Eq(d, 0)), (I*B*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) + B*d*x
/(2*b*d*tan(c + d*x) - 2*I*b*d) + I*B/(2*b*d*tan(c + d*x) - 2*I*b*d) + C*d*
x*tan(c + d*x)/(2*b*d*tan(c + d*x) - 2*I*b*d) - I*C*d*x/(2*b*d*tan(c + d*x)
 - 2*I*b*d) - C/(2*b*d*tan(c + d*x) - 2*I*b*d), Eq(a, -I*b)), (-I*B*d*x*tan
(c + d*x)/(2*b*d*tan(c + d*x) + 2*I*b*d) + B*d*x/(2*b*d*tan(c + d*x) + 2*I*
b*d) - I*B/(2*b*d*tan(c + d*x) + 2*I*b*d) + C*d*x*tan(c + d*x)/(2*b*d*tan(c
 + d*x) + 2*I*b*d) + I*C*d*x/(2*b*d*tan(c + d*x) + 2*I*b*d) - C/(2*b*d*tan(
c + d*x) + 2*I*b*d), Eq(a, I*b)), (x*(B*tan(c) + C*tan(c)**2)*cot(c)/(a + b
*tan(c)), Eq(d, 0)), ((B*x + C*log(tan(c + d*x)**2 + 1)/(2*d))/a, Eq(b, 0))
, (2*B*a*d*x/(2*a**2*d + 2*b**2*d) + 2*B*b*log(a/b + tan(c + d*x))/(2*a**2*
d + 2*b**2*d) - B*b*log(tan(c + d*x)**2 + 1)/(2*a**2*d + 2*b**2*d) - 2*C*a*
log(a/b + tan(c + d*x))/(2*a**2*d + 2*b**2*d) + C*a*log(tan(c + d*x)**2 + 1
)/(2*a**2*d + 2*b**2*d) + 2*C*b*d*x/(2*a**2*d + 2*b**2*d), True))
```

3.29 $\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$

Optimal. Leaf size=80

$$-\frac{b(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{ad\left(a^2+b^2\right)}-\frac{x(bB-aC)}{a^2+b^2}+\frac{B\log(\sin(c+dx))}{ad}$$

```
[Out] -(B*b-C*a)*x/(a^2+b^2)+B*ln(sin(d*x+c))/a/d-b*(B*b-C*a)*ln(a*cos(d*x+c)+b*s
in(d*x+c))/a/(a^2+b^2)/d
```

Rubi [A] time = 0.20, antiderivative size = 80, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.100, Rules used = {3632, 3611, 3530, 3475}

$$-\frac{b(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{ad\left(a^2+b^2\right)}-\frac{x(bB-aC)}{a^2+b^2}+\frac{B\log(\sin(c+dx))}{ad}$$

Antiderivative was successfully verified.

```
[In] Int[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
]),x]
```

```
[Out] -(((b*B - a*C)*x)/(a^2 + b^2)) + (B*Log[Sin[c + d*x]])/(a*d) - (b*(b*B - a*
C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/(a*(a^2 + b^2)*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3611

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)])/(((a_) + (b_.)*tan[(e_.) + (f_.
)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])), x_Symbol] :> Simp[((B*(b
*c + a*d) + A*(a*c - b*d))*x)/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(b*(A*b
 - a*B))/((b*c - a*d)*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e +
 f*x]), x], x] + Dist[(d*(B*c - A*d))/((b*c - a*d)*(c^2 + d^2)), Int[(d - c
```

```
*Tan[e + f*x])/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A,
 B}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \int \frac{\cot(c+dx)(B+C\tan(c+dx))}{a+b\tan(c+dx)}\,dx$$

$$= -\frac{(bB-aC)x}{a^2+b^2} + \frac{B\int\cot(c+dx)\,dx}{a} - \frac{(b(bB-aC))\int\frac{b-a\tan}{a+b\tan}}{a\left(a^2+b^2\right)}$$

$$= -\frac{(bB-aC)x}{a^2+b^2} + \frac{B\log(\sin(c+dx))}{ad} - \frac{b(bB-aC)\log(a\cos}{a\left(a^2\right.}$$

Mathematica [C] time = 0.37, size = 113, normalized size = 1.41

$$-\frac{\frac{2b(bB-aC)\log(a+b\tan(c+dx))}{a\left(a^2+b^2\right)}+\frac{(B+iC)\log(-\tan(c+dx)+i)}{a+ib}+\frac{(B-iC)\log(\tan(c+dx)+i)}{a-ib}-\frac{2B\log(\tan(c+dx))}{a}}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x]),x]
```

```
[Out] -1/2*(((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b) - (2*B*Log[Tan[c + d*x]])
/a + ((B - I*C)*Log[I + Tan[c + d*x]])/(a - I*b) + (2*b*(b*B - a*C)*Log[a +
 b*Tan[c + d*x]])/(a*(a^2 + b^2)))/d
```

fricas [A] time = 0.67, size = 118, normalized size = 1.48

$$\frac{2\left(Ca^2-Bab\right)dx+\left(Ba^2+Bb^2\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)+\left(Cab-Bb^2\right)\log\left(\frac{b^2\tan(dx+c)^2+2ab\tan(dx+c)+a^2}{\tan(dx+c)^2+1}\right)}{2\left(a^3+ab^2\right)d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="fricas")
```

```
[Out] 1/2*(2*(C*a^2 - B*a*b)*d*x + (B*a^2 + B*b^2)*log(tan(d*x + c)^2/(tan(d*x +
c)^2 + 1)) + (C*a*b - B*b^2)*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) +
 a^2)/(tan(d*x + c)^2 + 1)))/((a^3 + a*b^2)*d)
```

giac [A] time = 4.11, size = 113, normalized size = 1.41

$$\frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{2\left(Cab^2-Bb^3\right)\log(|b\tan(dx+c)+a|)}{a^3b+ab^3}+\frac{2\,B\log(|\tan(dx+c)|)}{a}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="giac")
```

```
[Out] 1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - (B*a + C*b)*log(tan(d*x + c)^2 +
 1)/(a^2 + b^2) + 2*(C*a*b^2 - B*b^3)*log(abs(b*tan(d*x + c) + a))/(a^3*b +
 a*b^3) + 2*B*log(abs(tan(d*x + c)))/a)/d
```

maple [B] time = 0.94, size = 174, normalized size = 2.18

$$-\frac{b^2\ln(a+b\tan(dx+c))\,B}{da\left(a^2+b^2\right)}+\frac{b\ln(a+b\tan(dx+c))\,C}{d\left(a^2+b^2\right)}+\frac{B\ln(\tan(dx+c))}{da}-\frac{\ln\left(1+\tan^2(dx+c)\right)aB}{2d\left(a^2+b^2\right)}-\frac{\ln\left(1+\right.}{2}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x)
```

```
[Out] -1/d*b^2/a/(a^2+b^2)*ln(a+b*tan(d*x+c))*B+1/d*b/(a^2+b^2)*ln(a+b*tan(d*x+c)
)*C+1/d*B/a*ln(tan(d*x+c))-1/2/d/(a^2+b^2)*ln(1+tan(d*x+c)^2)*a*B-1/2/d/(a^
2+b^2)*ln(1+tan(d*x+c)^2)*C*b-1/d/(a^2+b^2)*B*arctan(tan(d*x+c))*b+1/d/(a^2
+b^2)*C*arctan(tan(d*x+c))*a
```

maxima [A] time = 0.46, size = 107, normalized size = 1.34

$$\frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}+\frac{2\left(Cab-Bb^2\right)\log(b\tan(dx+c)+a)}{a^3+ab^2}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{2\,B\log(\tan(dx+c))}{a}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="maxima")
```

```
[Out] 1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) + 2*(C*a*b - B*b^2)*log(b*tan(d*x
+ c) + a)/(a^3 + a*b^2) - (B*a + C*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b^2) +
 2*B*log(tan(d*x + c))/a)/d
```

mupad [B] time = 9.46, size = 115, normalized size = 1.44

$$\frac{B\,\ln\left(\tan\left(c+d\,x\right)\right)}{a\,d}-\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-b+a\,1\mathrm{i}\right)}-\frac{\ln\left(\tan\left(c+d\,x\right)+1\mathrm{i}\right)\,\left(B-C\,1\mathrm{i}\right)}{2\,d\,\left(a-b\,1\mathrm{i}\right)}-\frac{b\,\ln\left(a+b\tan\left(c+d\,x\right)\right)}{a\,d\,\left(a^2+b^2\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
)),x)
```

```
[Out] (B*log(tan(c + d*x)))/(a*d) - (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*1
i - b)) - (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a - b*1i)) - (b*log(a +
 b*tan(c + d*x))*(B*b - C*a))/(a*d*(a^2 + b^2))
```

sympy [A] time = 5.75, size = 966, normalized size = 12.08

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)*cot(c)**2/tan(c), Eq(a, 0) & Eq(b
, 0) & Eq(d, 0)), ((-B*x - B/(d*tan(c + d*x)) - C*log(tan(c + d*x)**2 + 1)/
(2*d) + C*log(tan(c + d*x))/d)/b, Eq(a, 0)), (I*B*d*x*tan(c + d*x)/(2*I*b*d
*tan(c + d*x) + 2*b*d) + B*d*x/(2*I*b*d*tan(c + d*x) + 2*b*d) + B*log(tan(c
 + d*x)**2 + 1)*tan(c + d*x)/(2*I*b*d*tan(c + d*x) + 2*b*d) - I*B*log(tan(c
 + d*x)**2 + 1)/(2*I*b*d*tan(c + d*x) + 2*b*d) - 2*B*log(tan(c + d*x))*tan(
c + d*x)/(2*I*b*d*tan(c + d*x) + 2*b*d) + 2*I*B*log(tan(c + d*x))/(2*I*b*d*
tan(c + d*x) + 2*b*d) + I*B/(2*I*b*d*tan(c + d*x) + 2*b*d) - C*d*x*tan(c +
d*x)/(2*I*b*d*tan(c + d*x) + 2*b*d) + I*C*d*x/(2*I*b*d*tan(c + d*x) + 2*b*d
) - C/(2*I*b*d*tan(c + d*x) + 2*b*d), Eq(a, -I*b)), (-I*B*d*x*tan(c + d*x)/
(-2*I*b*d*tan(c + d*x) + 2*b*d) + B*d*x/(-2*I*b*d*tan(c + d*x) + 2*b*d) + B
*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(-2*I*b*d*tan(c + d*x) + 2*b*d) + I*
B*log(tan(c + d*x)**2 + 1)/(-2*I*b*d*tan(c + d*x) + 2*b*d) - 2*B*log(tan(c
+ d*x))*tan(c + d*x)/(-2*I*b*d*tan(c + d*x) + 2*b*d) - 2*I*B*log(tan(c + d*
x))/(-2*I*b*d*tan(c + d*x) + 2*b*d) - I*B/(-2*I*b*d*tan(c + d*x) + 2*b*d) -
 C*d*x*tan(c + d*x)/(-2*I*b*d*tan(c + d*x) + 2*b*d) - I*C*d*x/(-2*I*b*d*tan
(c + d*x) + 2*b*d) - C/(-2*I*b*d*tan(c + d*x) + 2*b*d), Eq(a, I*b)), (x*(B*
tan(c) + C*tan(c)**2)*cot(c)**2/(a + b*tan(c)), Eq(d, 0)), ((-B*log(tan(c +
 d*x)**2 + 1)/(2*d) + B*log(tan(c + d*x))/d + C*x)/a, Eq(b, 0)), (-B*a**2*l
og(tan(c + d*x)**2 + 1)/(2*a**3*d + 2*a*b**2*d) + 2*B*a**2*log(tan(c + d*x)
```

```
)/(2*a**3*d + 2*a*b**2*d) - 2*B*a*b*d*x/(2*a**3*d + 2*a*b**2*d) - 2*B*b**2*
log(a/b + tan(c + d*x))/(2*a**3*d + 2*a*b**2*d) + 2*B*b**2*log(tan(c + d*x)
)/(2*a**3*d + 2*a*b**2*d) + 2*C*a**2*d*x/(2*a**3*d + 2*a*b**2*d) + 2*C*a*b*
log(a/b + tan(c + d*x))/(2*a**3*d + 2*a*b**2*d) - C*a*b*log(tan(c + d*x)**2
 + 1)/(2*a**3*d + 2*a*b**2*d), True))
```

3.30 $\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$

Optimal. Leaf size=103

$$\frac{b^2(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a^2d\left(a^2+b^2\right)}-\frac{x(aB+bC)}{a^2+b^2}-\frac{(bB-aC)\log(\sin(c+dx))}{a^2d}-\frac{B\cot(c+dx)}{ad}$$

```
[Out] -(B*a+C*b)*x/(a^2+b^2)-B*cot(d*x+c)/a/d-(B*b-C*a)*ln(sin(d*x+c))/a^2/d+b^2*
(B*b-C*a)*ln(a*cos(d*x+c)+b*sin(d*x+c))/a^2/(a^2+b^2)/d
```

Rubi [A] time = 0.34, antiderivative size = 103, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3632, 3609, 3651, 3530, 3475}

$$\frac{b^2(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a^2d\left(a^2+b^2\right)}-\frac{x(aB+bC)}{a^2+b^2}-\frac{(bB-aC)\log(\sin(c+dx))}{a^2d}-\frac{B\cot(c+dx)}{ad}$$

Antiderivative was successfully verified.

```
[In] Int[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
]),x]
```

```
[Out] -(((a*B + b*C)*x)/(a^2 + b^2)) - (B*Cot[c + d*x])/(a*d) - ((b*B - a*C)*Log[
Sin[c + d*x]])/(a^2*d) + (b^2*(b*B - a*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*
x]])/(a^2*(a^2 + b^2)*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1)
)/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
```

```
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \int \frac{\cot^2(c+dx)(B+C\tan(c+dx))}{a+b\tan(c+dx)}\,dx$$

$$= -\frac{B\cot(c+dx)}{ad} - \frac{\int \frac{\cot(c+dx)\left(bB-aC+aB\tan(c+dx)+bB\tan^2(c+dx)\right)}{a+b\tan(c+dx)}}{a}$$

$$= -\frac{(aB+bC)x}{a^2+b^2} - \frac{B\cot(c+dx)}{ad} - \frac{(bB-aC)\int\cot(c+dx)}{a^2}$$

$$= -\frac{(aB+bC)x}{a^2+b^2} - \frac{B\cot(c+dx)}{ad} - \frac{(bB-aC)\log(\sin(c+d}{a^2d}$$

Mathematica [C] time = 0.89, size = 138, normalized size = 1.34

$$\frac{\frac{2b^2(bB-aC)\log(a+b\tan(c+dx))}{a^2(a^2+b^2)}+\frac{2(aC-bB)\log(\tan(c+dx))}{a^2}+\frac{i(B+iC)\log(-\tan(c+dx)+i)}{a+ib}-\frac{(C+iB)\log(\tan(c+dx)+i)}{a-ib}-\frac{2B\cot(c+dx)}{a}}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x]),x]
```

```
[Out] ((-2*B*Cot[c + d*x])/a + (I*(B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b) + (2
*(-(b*B) + a*C)*Log[Tan[c + d*x]])/a^2 - ((I*B + C)*Log[I + Tan[c + d*x]])/
(a - I*b) + (2*b^2*(b*B - a*C)*Log[a + b*Tan[c + d*x]])/(a^2*(a^2 + b^2)))/
(2*d)
```

fricas [A] time = 0.77, size = 177, normalized size = 1.72

$$-\frac{2\,Ba^3+2\,Bab^2+2\left(Ba^3+Ca^2b\right)dx\tan\left(dx+c\right)-\left(Ca^3-Ba^2b+Cab^2-Bb^3\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan\left(dx+c\right)+\left(\right.}{2\left(a^4+a^2b^2\right)d\tan\left(dx+c\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="fricas")
```

```
[Out] -1/2*(2*B*a^3 + 2*B*a*b^2 + 2*(B*a^3 + C*a^2*b)*d*x*tan(d*x + c) - (C*a^3 -
 B*a^2*b + C*a*b^2 - B*b^3)*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1))*tan(d*
x + c) + (C*a*b^2 - B*b^3)*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a
^2)/(tan(d*x + c)^2 + 1))*tan(d*x + c))/((a^4 + a^2*b^2)*d*tan(d*x + c))
```

giac [A] time = 4.81, size = 157, normalized size = 1.52

$$-\frac{\frac{2\,(Ba+Cb)(dx+c)}{a^2+b^2}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{2\left(Cab^3-Bb^4\right)\log(|b\tan(dx+c)+a|)}{a^4b+a^2b^3}-\frac{2\,(Ca-Bb)\log(|\tan(dx+c)|)}{a^2}+\frac{2\,(Ca\tan(dx+c)-Bb\tan(}{a^2\tan(dx+c)}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="giac")
```

```
[Out] -1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) + (C*a - B*b)*log(tan(d*x + c)^2
+ 1)/(a^2 + b^2) + 2*(C*a*b^3 - B*b^4)*log(abs(b*tan(d*x + c) + a))/(a^4*b
+ a^2*b^3) - 2*(C*a - B*b)*log(abs(tan(d*x + c)))/a^2 + 2*(C*a*tan(d*x + c)
 - B*b*tan(d*x + c) + B*a)/(a^2*tan(d*x + c)))/d
```

maple [B] time = 0.76, size = 214, normalized size = 2.08

$$\frac{b^3 \ln(a+b\tan(dx+c))B}{d\,a^2\left(a^2+b^2\right)}-\frac{b^2\ln(a+b\tan(dx+c))C}{da\left(a^2+b^2\right)}-\frac{B}{da\tan(dx+c)}-\frac{\ln(\tan(dx+c))Bb}{d\,a^2}+\frac{\ln(\tan(dx+c))}{da}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x)
```

```
[Out] 1/d*b^3/a^2/(a^2+b^2)*ln(a+b*tan(d*x+c))*B-1/d*b^2/a/(a^2+b^2)*ln(a+b*tan(d
*x+c))*C-1/d*B/a/tan(d*x+c)-1/d/a^2*ln(tan(d*x+c))*B*b+1/d/a*ln(tan(d*x+c))
*C+1/2/d/(a^2+b^2)*ln(1+tan(d*x+c)^2)*B*b-1/2/d/(a^2+b^2)*ln(1+tan(d*x+c)^2
)*a*C-1/d/(a^2+b^2)*B*arctan(tan(d*x+c))*a-1/d/(a^2+b^2)*C*arctan(tan(d*x+c
))*b
```

maxima [A] time = 0.97, size = 131, normalized size = 1.27

$$-\frac{\frac{2(Ba+Cb)(dx+c)}{a^2+b^2}+\frac{2\left(Cab^2-Bb^3\right)\log(b\tan(dx+c)+a)}{a^4+a^2b^2}+\frac{(Ca-Bb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}-\frac{2(Ca-Bb)\log(\tan(dx+c))}{a^2}+\frac{2B}{a\tan(dx+c)}}{2d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(B*a + C*b)*(d*x + c)/(a^2 + b^2) + 2*(C*a*b^2 - B*b^3)*log(b*tan(d
*x + c) + a)/(a^4 + a^2*b^2) + (C*a - B*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b
^2) - 2*(C*a - B*b)*log(tan(d*x + c))/a^2 + 2*B/(a*tan(d*x + c)))/d
```

mupad [B] time = 10.34, size = 140, normalized size = 1.36

$$\frac{\ln(a+b\tan(c+d\,x))\,\left(B\,b^3-C\,a\,b^2\right)}{d\,\left(a^4+a^2\,b^2\right)}-\frac{\ln(\tan(c+d\,x))\,(B\,b-C\,a)}{a^2\,d}+\frac{\ln(\tan(c+d\,x)+1\mathrm{i})\,(B-C\,1\mathrm{i})}{2\,d\,(b+a\,1\mathrm{i})}-\frac{B\cot(c}{a\,}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
)),x)
```

```
[Out] (log(a + b*tan(c + d*x))*(B*b^3 - C*a*b^2))/(d*(a^4 + a^2*b^2)) - (log(tan(
c + d*x))*(B*b - C*a))/(a^2*d) + (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(
a*1i + b)) - (B*cot(c + d*x))/(a*d) + (log(tan(c + d*x) - 1i)*(B*1i - C))/(
2*d*(a + b*1i))
```

sympy [A] time = 12.10, size = 2064, normalized size = 20.04

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**3*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((nan, Eq(a, 0) & Eq(b, 0) & Eq(c, 0) & Eq(d, 0)), ((-B*x - B/(d*t
an(c + d*x)) - C*log(tan(c + d*x)**2 + 1)/(2*d) + C*log(tan(c + d*x))/d)/a,
 Eq(b, 0)), ((B*log(tan(c + d*x)**2 + 1)/(2*d) - B*log(tan(c + d*x))/d - B/
(2*d*tan(c + d*x)**2) - C*x - C/(d*tan(c + d*x)))/b, Eq(a, 0)), (-3*I*B*d*x
*tan(c + d*x)**2/(2*b*d*tan(c + d*x)**2 - 2*I*b*d*tan(c + d*x)) - 3*B*d*x*t
an(c + d*x)/(2*b*d*tan(c + d*x)**2 - 2*I*b*d*tan(c + d*x)) - B*log(tan(c +
d*x)**2 + 1)*tan(c + d*x)**2/(2*b*d*tan(c + d*x)**2 - 2*I*b*d*tan(c + d*x))
 + I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c + d*x)**2 - 2*I*b
*d*tan(c + d*x)) + 2*B*log(tan(c + d*x))*tan(c + d*x)**2/(2*b*d*tan(c + d*x
)**2 - 2*I*b*d*tan(c + d*x)) - 2*I*B*log(tan(c + d*x))*tan(c + d*x)/(2*b*d*
tan(c + d*x)**2 - 2*I*b*d*tan(c + d*x)) - 3*I*B*tan(c + d*x)/(2*b*d*tan(c +
 d*x)**2 - 2*I*b*d*tan(c + d*x)) - 2*B/(2*b*d*tan(c + d*x)**2 - 2*I*b*d*tan
(c + d*x)) + C*d*x*tan(c + d*x)**2/(2*b*d*tan(c + d*x)**2 - 2*I*b*d*tan(c +
 d*x)) - I*C*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x)**2 - 2*I*b*d*tan(c + d*x)
) - I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(2*b*d*tan(c + d*x)**2 - 2
*I*b*d*tan(c + d*x)) - C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c
 + d*x)**2 - 2*I*b*d*tan(c + d*x)) + 2*I*C*log(tan(c + d*x))*tan(c + d*x)**
2/(2*b*d*tan(c + d*x)**2 - 2*I*b*d*tan(c + d*x)) + 2*C*log(tan(c + d*x))*ta
n(c + d*x)/(2*b*d*tan(c + d*x)**2 - 2*I*b*d*tan(c + d*x)) + C*tan(c + d*x)/
(2*b*d*tan(c + d*x)**2 - 2*I*b*d*tan(c + d*x)), Eq(a, -I*b)), (3*I*B*d*x*ta
n(c + d*x)**2/(2*b*d*tan(c + d*x)**2 + 2*I*b*d*tan(c + d*x)) - 3*B*d*x*tan(
c + d*x)/(2*b*d*tan(c + d*x)**2 + 2*I*b*d*tan(c + d*x)) - B*log(tan(c + d*x
)**2 + 1)*tan(c + d*x)**2/(2*b*d*tan(c + d*x)**2 + 2*I*b*d*tan(c + d*x)) -
I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c + d*x)**2 + 2*I*b*d*
tan(c + d*x)) + 2*B*log(tan(c + d*x))*tan(c + d*x)**2/(2*b*d*tan(c + d*x)**
2 + 2*I*b*d*tan(c + d*x)) + 2*I*B*log(tan(c + d*x))*tan(c + d*x)/(2*b*d*tan
(c + d*x)**2 + 2*I*b*d*tan(c + d*x)) + 3*I*B*tan(c + d*x)/(2*b*d*tan(c + d*
x)**2 + 2*I*b*d*tan(c + d*x)) - 2*B/(2*b*d*tan(c + d*x)**2 + 2*I*b*d*tan(c
+ d*x)) + C*d*x*tan(c + d*x)**2/(2*b*d*tan(c + d*x)**2 + 2*I*b*d*tan(c + d*
x)) + I*C*d*x*tan(c + d*x)/(2*b*d*tan(c + d*x)**2 + 2*I*b*d*tan(c + d*x)) +
 I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(2*b*d*tan(c + d*x)**2 + 2*I*
b*d*tan(c + d*x)) - C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*b*d*tan(c +
d*x)**2 + 2*I*b*d*tan(c + d*x)) - 2*I*C*log(tan(c + d*x))*tan(c + d*x)**2/(
2*b*d*tan(c + d*x)**2 + 2*I*b*d*tan(c + d*x)) + 2*C*log(tan(c + d*x))*tan(c
 + d*x)/(2*b*d*tan(c + d*x)**2 + 2*I*b*d*tan(c + d*x)) + C*tan(c + d*x)/(2*
b*d*tan(c + d*x)**2 + 2*I*b*d*tan(c + d*x)), Eq(a, I*b)), (nan, Eq(c, -d*x)
), (x*(B*tan(c) + C*tan(c)**2)*cot(c)**3/(a + b*tan(c)), Eq(d, 0)), (-2*B*a
```

```
**3*d*x*tan(c + d*x)/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) -
 2*B*a**3/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) + B*a**2*b*l
og(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d
*tan(c + d*x)) - 2*B*a**2*b*log(tan(c + d*x))*tan(c + d*x)/(2*a**4*d*tan(c
+ d*x) + 2*a**2*b**2*d*tan(c + d*x)) - 2*B*a*b**2/(2*a**4*d*tan(c + d*x) +
2*a**2*b**2*d*tan(c + d*x)) + 2*B*b**3*log(a/b + tan(c + d*x))*tan(c + d*x)
/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) - 2*B*b**3*log(tan(c
+ d*x))*tan(c + d*x)/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) -
 C*a**3*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**4*d*tan(c + d*x) + 2*a*
*2*b**2*d*tan(c + d*x)) + 2*C*a**3*log(tan(c + d*x))*tan(c + d*x)/(2*a**4*d
*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) - 2*C*a**2*b*d*x*tan(c + d*x)/(
2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d*x)) - 2*C*a*b**2*log(a/b +
tan(c + d*x))*tan(c + d*x)/(2*a**4*d*tan(c + d*x) + 2*a**2*b**2*d*tan(c + d
*x)) + 2*C*a*b**2*log(tan(c + d*x))*tan(c + d*x)/(2*a**4*d*tan(c + d*x) + 2
*a**2*b**2*d*tan(c + d*x)), True))
```

3.31 $\int \frac{\cot^4(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx$

Optimal. Leaf size=137

$$\frac{x(bB-aC)}{a^2+b^2}+\frac{(bB-aC)\cot(c+dx)}{a^2d}-\frac{\left(a^2B+abC-b^2B\right)\log(\sin(c+dx))}{a^3d}-\frac{b^3(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx)}{a^3d\left(a^2+b^2\right)}$$

```
[Out] (B*b-C*a)*x/(a^2+b^2)+(B*b-C*a)*cot(d*x+c)/a^2/d-1/2*B*cot(d*x+c)^2/a/d-(B*
a^2-B*b^2+C*a*b)*ln(sin(d*x+c))/a^3/d-b^3*(B*b-C*a)*ln(a*cos(d*x+c)+b*sin(d
*x+c))/a^3/(a^2+b^2)/d
```

Rubi [A] time = 0.68, antiderivative size = 137, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3632, 3609, 3649, 3651, 3530, 3475}

$$-\frac{\left(a^2B+abC-b^2B\right)\log(\sin(c+dx))}{a^3d}-\frac{b^3(bB-aC)\log(a\cos(c+dx)+b\sin(c+dx))}{a^3d\left(a^2+b^2\right)}+\frac{x(bB-aC)}{a^2+b^2}+\frac{(bB-aC)\cot(c+dx)}{a^2d}$$

Antiderivative was successfully verified.

```
[In] Int[(Cot[c + d*x]^4*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
]),x]
```

```
[Out] ((b*B - a*C)*x)/(a^2 + b^2) + ((b*B - a*C)*Cot[c + d*x])/(a^2*d) - (B*Cot[c
 + d*x]^2)/(2*a*d) - ((a^2*B - b^2*B + a*b*C)*Log[Sin[c + d*x]])/(a^3*d) -
(b^3*(b*B - a*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/(a^3*(a^2 + b^2)*d)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
```

```
mp[(b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1)
)/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{\cot^4(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{a+b\tan(c+dx)}\,dx = \int \frac{\cot^3(c+dx)(B+C\tan(c+dx))}{a+b\tan(c+dx)}\,dx$$

$$= -\frac{B\cot^2(c+dx)}{2ad} - \frac{\int \frac{\cot^2(c+dx)\left(2(bB-aC)+2aB\tan(c+dx)+2bB\tan^2(\right.}{a+b\tan(c+dx)}}{2a}$$

$$= \frac{(bB-aC)\cot(c+dx)}{a^2d} - \frac{B\cot^2(c+dx)}{2ad} + \frac{\int \frac{\cot(c+dx)\left(-2\left(a^2B\right.\right.}{}}{}$$

$$= \frac{(bB-aC)x}{a^2+b^2} + \frac{(bB-aC)\cot(c+dx)}{a^2d} - \frac{B\cot^2(c+dx)}{2ad} - \frac{(b}{}$$

$$= \frac{(bB-aC)x}{a^2+b^2} + \frac{(bB-aC)\cot(c+dx)}{a^2d} - \frac{B\cot^2(c+dx)}{2ad} - \frac{(a}{}$$

Mathematica [C] time = 1.42, size = 163, normalized size = 1.19

$$\frac{\frac{2(bB-aC)\cot(c+dx)}{a^2} - \frac{2\left(a^2B+abC-b^2B\right)\log(\tan(c+dx))}{a^3} + \frac{2b^3(aC-bB)\log(a+b\tan(c+dx))}{a^3\left(a^2+b^2\right)} + \frac{(B+iC)\log(-\tan(c+dx)+i)}{a+ib} + \frac{(B-iC)\log(\tan(c+dx}{a-ib}}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Cot[c + d*x]^4*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x]),x]
```

```
[Out] ((2*(b*B - a*C)*Cot[c + d*x])/a^2 - (B*Cot[c + d*x]^2)/a + ((B + I*C)*Log[I
 - Tan[c + d*x]])/(a + I*b) - (2*(a^2*B - b^2*B + a*b*C)*Log[Tan[c + d*x]])
/a^3 + ((B - I*C)*Log[I + Tan[c + d*x]])/(a - I*b) + (2*b^3*(-(b*B) + a*C)*
Log[a + b*Tan[c + d*x]])/(a^3*(a^2 + b^2)))/(2*d)
```

fricas [A] time = 0.63, size = 234, normalized size = 1.71

$$-\frac{Ba^4+Ba^2b^2+\left(Ba^4+Ca^3b+Cab^3-Bb^4\right)\log\left(\frac{\tan(dx+c)^2}{\tan(dx+c)^2+1}\right)\tan(dx+c)^2-\left(Cab^3-Bb^4\right)\log\left(\frac{b^2\tan(dx+c)^2+2\,ab\,\mathrm{ta}}{\tan(dx+c)^2}\right.}{2\left(a^5+a\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="fricas")
```

```
[Out] -1/2*(B*a^4 + B*a^2*b^2 + (B*a^4 + C*a^3*b + C*a*b^3 - B*b^4)*log(tan(d*x +
 c)^2/(tan(d*x + c)^2 + 1))*tan(d*x + c)^2 - (C*a*b^3 - B*b^4)*log((b^2*tan
(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1))*tan(d*x + c)^
2 + (B*a^4 + B*a^2*b^2 + 2*(C*a^4 - B*a^3*b)*d*x)*tan(d*x + c)^2 + 2*(C*a^4
 - B*a^3*b + C*a^2*b^2 - B*a*b^3)*tan(d*x + c))/((a^5 + a^3*b^2)*d*tan(d*x
+ c)^2)
```

giac [A] time = 5.48, size = 214, normalized size = 1.56

$$-\frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}-\frac{2\left(Cab^4-Bb^5\right)\log(|b\tan(dx+c)+a|)}{a^5b+a^3b^3}+\frac{2\left(Ba^2+Cab-Bb^2\right)\log(|\tan(dx+c)|)}{a^3}-\frac{3\,Ba^2\tan(dx+c}{}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="giac")
```

```
[Out] -1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - (B*a + C*b)*log(tan(d*x + c)^2
+ 1)/(a^2 + b^2) - 2*(C*a*b^4 - B*b^5)*log(abs(b*tan(d*x + c) + a))/(a^5*b
+ a^3*b^3) + 2*(B*a^2 + C*a*b - B*b^2)*log(abs(tan(d*x + c)))/a^3 - (3*B*a^
2*tan(d*x + c)^2 + 3*C*a*b*tan(d*x + c)^2 - 3*B*b^2*tan(d*x + c)^2 - 2*C*a^
2*tan(d*x + c) + 2*B*a*b*tan(d*x + c) - B*a^2)/(a^3*tan(d*x + c)^2))/d
```

maple [A] time = 0.98, size = 266, normalized size = 1.94

$$-\frac{b^4\ln(a+b\tan(dx+c))\,B}{d\,a^3\left(a^2+b^2\right)}+\frac{b^3\ln(a+b\tan(dx+c))\,C}{d\,a^2\left(a^2+b^2\right)}-\frac{B}{2da\tan(dx+c)^2}+\frac{Bb}{d\,a^2\tan(dx+c)}-\frac{C}{da\tan(dx+c)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^4*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x)
```

```
[Out] -1/d*b^4/a^3/(a^2+b^2)*ln(a+b*tan(d*x+c))*B+1/d*b^3/a^2/(a^2+b^2)*ln(a+b*ta
n(d*x+c))*C-1/2/d*B/a/tan(d*x+c)^2+1/d/a^2/tan(d*x+c)*B*b-1/d/a/tan(d*x+c)*
C-1/d*B/a*ln(tan(d*x+c))+1/d/a^3*ln(tan(d*x+c))*b^2*B-1/d/a^2*ln(tan(d*x+c)
)*C*b+1/2/d/(a^2+b^2)*ln(1+tan(d*x+c)^2)*a*B+1/2/d/(a^2+b^2)*ln(1+tan(d*x+c
)^2)*C*b+1/d/(a^2+b^2)*B*arctan(tan(d*x+c))*b-1/d/(a^2+b^2)*C*arctan(tan(d*
x+c))*a
```

maxima [A] time = 0.62, size = 158, normalized size = 1.15

$$-\frac{\frac{2\,(Ca-Bb)(dx+c)}{a^2+b^2}-\frac{2\left(Cab^3-Bb^4\right)\log(b\tan(dx+c)+a)}{a^5+a^3b^2}-\frac{(Ba+Cb)\log\left(\tan(dx+c)^2+1\right)}{a^2+b^2}+\frac{2\left(Ba^2+Cab-Bb^2\right)\log(\tan(dx+c))}{a^3}+\frac{Ba+2\,(Ca-Bb)\tan}{a^2\tan(dx+}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^4*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c)),x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(C*a - B*b)*(d*x + c)/(a^2 + b^2) - 2*(C*a*b^3 - B*b^4)*log(b*tan(d
*x + c) + a)/(a^5 + a^3*b^2) - (B*a + C*b)*log(tan(d*x + c)^2 + 1)/(a^2 + b
^2) + 2*(B*a^2 + C*a*b - B*b^2)*log(tan(d*x + c))/a^3 + (B*a + 2*(C*a - B*b
)*tan(d*x + c))/(a^2*tan(d*x + c)^2))/d
```

mupad [B] time = 10.93, size = 175, normalized size = 1.28

$$-\frac{\cot\left(c+d\,x\right)^2\,\left(\frac{B}{2\,a}-\frac{\tan\left(c+d\,x\right)\,\left(B\,b-C\,a\right)}{a^2}\right)}{d}+\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-b+a\,1\mathrm{i}\right)}-\frac{\ln\left(\tan\left(c+d\,x\right)\right)\,\left(B\,a^2+C\,a\,b-B\,b^2\right)}{a^3\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((cot(c + d*x)^4*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
)),x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*1i - b)) - (cot(c + d*x)^2*(B/(
2*a) - (tan(c + d*x)*(B*b - C*a))/a^2))/d - (log(tan(c + d*x))*(B*a^2 - B*b
^2 + C*a*b))/(a^3*d) - (log(a + b*tan(c + d*x))*(B*b^4 - C*a*b^3))/(d*(a^5
+ a^3*b^2)) + (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a - b*1i))
```

sympy [A] time = 33.89, size = 2621, normalized size = 19.13

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**4*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c)),x)
```

```
[Out] Piecewise((nan, Eq(a, 0) & Eq(b, 0) & Eq(c, 0) & Eq(d, 0)), ((B*x + B/(d*ta
n(c + d*x)) - B/(3*d*tan(c + d*x)**3) + C*log(tan(c + d*x)**2 + 1)/(2*d) -
C*log(tan(c + d*x))/d - C/(2*d*tan(c + d*x)**2))/b, Eq(a, 0)), (3*B*d*x*tan
(c + d*x)**3/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) - 3*I*B*d*x
*tan(c + d*x)**2/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) - 2*I*B
*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d
*tan(c + d*x)**2) - 2*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(-2*b*d*ta
n(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) + 4*I*B*log(tan(c + d*x))*tan(c +
d*x)**3/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) + 4*B*log(tan(c
+ d*x))*tan(c + d*x)**2/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2)
+ 3*B*tan(c + d*x)**2/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) -
I*B*tan(c + d*x)/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) + B/(-2
*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) + 3*I*C*d*x*tan(c + d*x)**3
/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) + 3*C*d*x*tan(c + d*x)*
*2/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) + C*log(tan(c + d*x)*
```

```
*2 + 1)*tan(c + d*x)**3/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2)
- I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(-2*b*d*tan(c + d*x)**3 + 2*
I*b*d*tan(c + d*x)**2) - 2*C*log(tan(c + d*x))*tan(c + d*x)**3/(-2*b*d*tan(
c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) + 2*I*C*log(tan(c + d*x))*tan(c + d*
x)**2/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) + 3*I*C*tan(c + d*
x)**2/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2) + 2*C*tan(c + d*x)
/(-2*b*d*tan(c + d*x)**3 + 2*I*b*d*tan(c + d*x)**2), Eq(a, -I*b)), (3*B*d*x
*tan(c + d*x)**3/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) + 3*I*B
*d*x*tan(c + d*x)**2/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) + 2
*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(-2*b*d*tan(c + d*x)**3 - 2*I
*b*d*tan(c + d*x)**2) - 2*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(-2*b*
d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) - 4*I*B*log(tan(c + d*x))*tan(
c + d*x)**3/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) + 4*B*log(ta
n(c + d*x))*tan(c + d*x)**2/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)*
*2) + 3*B*tan(c + d*x)**2/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2
) + I*B*tan(c + d*x)/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) + B
/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) - 3*I*C*d*x*tan(c + d*x
)**3/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) + 3*C*d*x*tan(c + d
*x)**2/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) + C*log(tan(c + d
*x)**2 + 1)*tan(c + d*x)**3/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)*
*2) + I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(-2*b*d*tan(c + d*x)**3
- 2*I*b*d*tan(c + d*x)**2) - 2*C*log(tan(c + d*x))*tan(c + d*x)**3/(-2*b*d*
tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) - 2*I*C*log(tan(c + d*x))*tan(c
+ d*x)**2/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) - 3*I*C*tan(c
+ d*x)**2/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2) + 2*C*tan(c +
d*x)/(-2*b*d*tan(c + d*x)**3 - 2*I*b*d*tan(c + d*x)**2), Eq(a, I*b)), (nan,
 Eq(c, -d*x)), (x*(B*tan(c) + C*tan(c)**2)*cot(c)**4/(a + b*tan(c)), Eq(d,
0)), ((B*log(tan(c + d*x)**2 + 1)/(2*d) - B*log(tan(c + d*x))/d - B/(2*d*ta
n(c + d*x)**2) - C*x - C/(d*tan(c + d*x)))/a, Eq(b, 0)), (B*a**4*log(tan(c
+ d*x)**2 + 1)*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*ta
n(c + d*x)**2) - 2*B*a**4*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**5*d*tan(c
 + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) - B*a**4/(2*a**5*d*tan(c + d*x)
**2 + 2*a**3*b**2*d*tan(c + d*x)**2) + 2*B*a**3*b*d*x*tan(c + d*x)**2/(2*a*
*5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) + 2*B*a**3*b*tan(c +
d*x)/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) - B*a**2*b*
*2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) + 2*B*a*b**3*
tan(c + d*x)/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) - 2
*B*b**4*log(a/b + tan(c + d*x))*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 +
 2*a**3*b**2*d*tan(c + d*x)**2) + 2*B*b**4*log(tan(c + d*x))*tan(c + d*x)**
2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) - 2*C*a**4*d*x
*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2)
 - 2*C*a**4*tan(c + d*x)/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c +
d*x)**2) + C*a**3*b*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(2*a**5*d*tan(
c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) - 2*C*a**3*b*log(tan(c + d*x))
*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2)
```

```
 - 2*C*a**2*b**2*tan(c + d*x)/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan
(c + d*x)**2) + 2*C*a*b**3*log(a/b + tan(c + d*x))*tan(c + d*x)**2/(2*a**5*
d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*x)**2) - 2*C*a*b**3*log(tan(c +
 d*x))*tan(c + d*x)**2/(2*a**5*d*tan(c + d*x)**2 + 2*a**3*b**2*d*tan(c + d*
x)**2), True))
```

3.32 $\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$

Optimal. Leaf size=208

$$\frac{a(bB-aC)\tan^2(c+dx)}{bd\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{\left(-2a^2C+abB-b^2C\right)\tan(c+dx)}{b^2d\left(a^2+b^2\right)}+\frac{\left(a^2B+2abC-b^2B\right)\log(\cos(c+dx))}{d\left(a^2+b^2\right)^2}-\frac{x\left(a\right.}{}$$

```
[Out] -(2*B*a*b-C*a^2+C*b^2)*x/(a^2+b^2)^2+(B*a^2-B*b^2+2*C*a*b)*ln(cos(d*x+c))/(
a^2+b^2)^2/d+a^2*(B*a^2*b+3*B*b^3-2*C*a^3-4*C*a*b^2)*ln(a+b*tan(d*x+c))/b^3
/(a^2+b^2)^2/d-(B*a*b-2*C*a^2-C*b^2)*tan(d*x+c)/b^2/(a^2+b^2)/d+a*(B*b-C*a)
*tan(d*x+c)^2/b/(a^2+b^2)/d/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.53, antiderivative size = 208, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.175, Rules used = {3632, 3605, 3647, 3626, 3617, 31, 3475}

$$\frac{a(bB-aC)\tan^2(c+dx)}{bd\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{\left(-2a^2C+abB-b^2C\right)\tan(c+dx)}{b^2d\left(a^2+b^2\right)}+\frac{a^2\left(a^2bB-2a^3C-4ab^2C+3b^3B\right)\log(a+}{b^3d\left(a^2+b^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^2,x]
```

```
[Out] -(((2*a*b*B - a^2*C + b^2*C)*x)/(a^2 + b^2)^2) + ((a^2*B - b^2*B + 2*a*b*C)
*Log[Cos[c + d*x]])/((a^2 + b^2)^2*d) + (a^2*(a^2*b*B + 3*b^3*B - 2*a^3*C -
 4*a*b^2*C)*Log[a + b*Tan[c + d*x]])/(b^3*(a^2 + b^2)^2*d) - ((a*b*B - 2*a^
2*C - b^2*C)*Tan[c + d*x])/(b^2*(a^2 + b^2)*d) + (a*(b*B - a*C)*Tan[c + d*x
]^2)/(b*(a^2 + b^2)*d*(a + b*Tan[c + d*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3605

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
```

```
mp[((b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x
])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
```

```
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \int \frac{\tan^3(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^2}\,dx$$

$$= \frac{a(bB-aC)\tan^2(c+dx)}{b\left(a^2+b^2\right)d(a+b\tan(c+dx))} + \frac{\int \frac{\tan(c+dx)\left(-2a(bB-aC)+b\right.}{}}{}$$

$$= -\frac{\left(abB-2a^2C-b^2C\right)\tan(c+dx)}{b^2\left(a^2+b^2\right)d} + \frac{a(bB-aC)\tan^2}{b\left(a^2+b^2\right)d(a+b\,\mathrm{t}}$$

$$= -\frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2} - \frac{\left(abB-2a^2C-b^2C\right)\tan(c+a}{b^2\left(a^2+b^2\right)d}$$

$$= -\frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2} + \frac{\left(a^2B-b^2B+2abC\right)\log(\cos(}{\left(a^2+b^2\right)^2d}$$

$$= -\frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2} + \frac{\left(a^2B-b^2B+2abC\right)\log(\cos(}{\left(a^2+b^2\right)^2d}$$

Mathematica [C] time = 4.27, size = 444, normalized size = 2.13

$$\frac{2b^2C\left(a^2+b^2\right)^2\tan^2(c+dx)+2ia^2\left(2a^3C-a^2bB+4ab^2C-3b^3B\right)\tan^{-1}(\tan(c+dx))(a+b\tan(c+dx))+a\left(2\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^2,x]
```

```
[Out] (a*(2*(a + I*b)^2*(2*a*b^2*(B + I*C) + I*a^2*b*(B + (4*I)*C) - (2*I)*a^3*C
+ b^3*C)*(c + d*x) + 2*(a^2 + b^2)^2*(-(b*B) + 2*a*C)*Log[Cos[c + d*x]] + a
^2*(a^2*b*B + 3*b^3*B - 2*a^3*C - 4*a*b^2*C)*Log[(a*Cos[c + d*x] + b*Sin[c
+ d*x])^2]) + b*(2*(a^3*b^2*C*(3 - (4*I)*c - (4*I)*d*x) - b^5*C*(c + d*x) +
 I*a^4*b*B*(I + c + d*x) - (2*I)*a^5*C*(I + c + d*x) + a*b^4*(C - 2*B*(c +
```

```
d*x)) + a^2*b^3*(C*(c + d*x) + I*B*(I + 3*c + 3*d*x))) + 2*(a^2 + b^2)^2*(-
(b*B) + 2*a*C)*Log[Cos[c + d*x]] + a^2*(a^2*b*B + 3*b^3*B - 2*a^3*C - 4*a*b
^2*C)*Log[(a*Cos[c + d*x] + b*Sin[c + d*x])^2])*Tan[c + d*x] + 2*b^2*(a^2 +
 b^2)^2*C*Tan[c + d*x]^2 + (2*I)*a^2*(-(a^2*b*B) - 3*b^3*B + 2*a^3*C + 4*a*
b^2*C)*ArcTan[Tan[c + d*x]]*(a + b*Tan[c + d*x]))/(2*b^3*(a^2 + b^2)^2*d*(a
 + b*Tan[c + d*x]))
```

fricas [B] time = 0.81, size = 434, normalized size = 2.09

$$-\frac{2\,Ca^4b^2 - 2\,Ba^3b^3 - 2\left(Ca^3b^3 - 2\,Ba^2b^4 - Cab^5\right)dx - 2\left(Ca^4b^2 + 2\,Ca^2b^4 + Cb^6\right)\tan\left(dx+c\right)^2 + \left(2\,Ca^6 - Ba^5b\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="fricas")
```

```
[Out] -1/2*(2*C*a^4*b^2 - 2*B*a^3*b^3 - 2*(C*a^3*b^3 - 2*B*a^2*b^4 - C*a*b^5)*d*x
 - 2*(C*a^4*b^2 + 2*C*a^2*b^4 + C*b^6)*tan(d*x + c)^2 + (2*C*a^6 - B*a^5*b
+ 4*C*a^4*b^2 - 3*B*a^3*b^3 + (2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 - 3*B*a^
2*b^4)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(t
an(d*x + c)^2 + 1)) - (2*C*a^6 - B*a^5*b + 4*C*a^4*b^2 - 2*B*a^3*b^3 + 2*C*
a^2*b^4 - B*a*b^5 + (2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 - 2*B*a^2*b^4 + 2*
C*a*b^5 - B*b^6)*tan(d*x + c))*log(1/(tan(d*x + c)^2 + 1)) - 2*(2*C*a^5*b -
 B*a^4*b^2 + 2*C*a^3*b^3 + C*a*b^5 + (C*a^2*b^4 - 2*B*a*b^5 - C*b^6)*d*x)*t
an(d*x + c))/((a^4*b^4 + 2*a^2*b^6 + b^8)*d*tan(d*x + c) + (a^5*b^3 + 2*a^3
*b^5 + a*b^7)*d)
```

giac [A] time = 2.94, size = 290, normalized size = 1.39

$$\frac{\frac{2\left(Ca^2-2\,Bab-Cb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4} - \frac{\left(Ba^2+2\,Cab-Bb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4} - \frac{2\left(2\,Ca^5-Ba^4b+4\,Ca^3b^2-3\,Ba^2b^3\right)\log\left(|b\tan(dx+c)+a|\right)}{a^4b^3+2\,a^2b^5+b^7} + \frac{2\,C\tan(dx+c)}{b^2} + \cdots}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="giac")
```

```
[Out] 1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - (B*a^2
 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2*(2*
C*a^5 - B*a^4*b + 4*C*a^3*b^2 - 3*B*a^2*b^3)*log(abs(b*tan(d*x + c) + a))/(
a^4*b^3 + 2*a^2*b^5 + b^7) + 2*C*tan(d*x + c)/b^2 + 2*(2*C*a^5*b*tan(d*x +
c) - B*a^4*b^2*tan(d*x + c) + 4*C*a^3*b^3*tan(d*x + c) - 3*B*a^2*b^4*tan(d*
x + c) + C*a^6 + 3*C*a^4*b^2 - 2*B*a^3*b^3)/((a^4*b^3 + 2*a^2*b^5 + b^7)*(b
*tan(d*x + c) + a)))/d
```

maple [A] time = 0.25, size = 364, normalized size = 1.75

$$\frac{C\tan(dx+c)}{d\,b^2}+\frac{a^4\ln(a+b\tan(dx+c))\,B}{d\,b^2\left(a^2+b^2\right)^2}+\frac{3a^2\ln(a+b\tan(dx+c))\,B}{d\left(a^2+b^2\right)^2}-\frac{2a^5\ln(a+b\tan(dx+c))\,C}{d\,b^3\left(a^2+b^2\right)^2}-\frac{4a^3\ln(}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x)
```

```
[Out] 1/d*C/b^2*tan(d*x+c)+1/d/b^2*a^4/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*B+3/d*a^2/(
a^2+b^2)^2*ln(a+b*tan(d*x+c))*B-2/d/b^3*a^5/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*
C-4/d/b*a^3/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*C+1/d/b^2*a^3/(a^2+b^2)/(a+b*tan
(d*x+c))*B-1/d/b^3*a^4/(a^2+b^2)/(a+b*tan(d*x+c))*C-1/2/d/(a^2+b^2)^2*ln(1+
tan(d*x+c)^2)*a^2*B+1/2/d/(a^2+b^2)^2*ln(1+tan(d*x+c)^2)*b^2*B-1/d/(a^2+b^2
)^2*ln(1+tan(d*x+c)^2)*C*a*b-2/d/(a^2+b^2)^2*B*arctan(tan(d*x+c))*a*b+1/d/(
a^2+b^2)^2*C*arctan(tan(d*x+c))*a^2-1/d/(a^2+b^2)^2*C*arctan(tan(d*x+c))*b^
2
```

maxima [A] time = 0.57, size = 220, normalized size = 1.06

$$\frac{\frac{2\left(Ca^2-2\,Bab-Cb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(2\,Ca^5-Ba^4b+4\,Ca^3b^2-3\,Ba^2b^3\right)\log(b\tan(dx+c)+a)}{a^4b^3+2\,a^2b^5+b^7}-\frac{\left(Ba^2+2\,Cab-Bb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(Ca^4-\right.}{a^3b^3+ab^5+\left(a^2b^4+\right.}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="maxima")
```

```
[Out] 1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - 2*(2*C
*a^5 - B*a^4*b + 4*C*a^3*b^2 - 3*B*a^2*b^3)*log(b*tan(d*x + c) + a)/(a^4*b^
3 + 2*a^2*b^5 + b^7) - (B*a^2 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a
^4 + 2*a^2*b^2 + b^4) - 2*(C*a^4 - B*a^3*b)/(a^3*b^3 + a*b^5 + (a^2*b^4 + b
^6)*tan(d*x + c)) + 2*C*tan(d*x + c)/b^2)/d
```

mupad [B] time = 9.65, size = 210, normalized size = 1.01

$$\frac{C\tan(c+d\,x)}{b^2\,d}-\frac{\ln(a+b\tan(c+d\,x))\left(2\,C\,a^5-B\,a^4\,b+4\,C\,a^3\,b^2-3\,B\,a^2\,b^3\right)}{d\left(a^4\,b^3+2\,a^2\,b^5+b^7\right)}-\frac{\ln(\tan(c+d\,x)-\mathrm{i})\,(B+C\,1\mathrm{i})}{2\,d\left(a^2+a\,b\,2\mathrm{i}-b^2\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((tan(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^2,x)
```

```
[Out] (C*tan(c + d*x))/(b^2*d) - (log(a + b*tan(c + d*x))*(2*C*a^5 - 3*B*a^2*b^3
+ 4*C*a^3*b^2 - B*a^4*b))/(d*(b^7 + 2*a^2*b^5 + a^4*b^3)) - (log(tan(c + d*
x) - 1i)*(B + C*1i))/(2*d*(a*b*2i + a^2 - b^2)) - (log(tan(c + d*x) + 1i)*(
B*1i + C))/(2*d*(2*a*b + a^2*1i - b^2*1i)) - (a^2*(C*a^2 - B*a*b))/(b*d*(a*
b^2 + b^3*tan(c + d*x))*(a^2 + b^2))
```

sympy [A] time = 2.98, size = 4602, normalized size = 22.12

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,
x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2), Eq(a, 0) & Eq(b, 0) & Eq(d, 0)),
 ((-B*log(tan(c + d*x)**2 + 1)/(2*d) + B*tan(c + d*x)**2/(2*d) + C*x + C*ta
n(c + d*x)**3/(3*d) - C*tan(c + d*x)/d)/a**2, Eq(b, 0)), (-3*B*d*x*tan(c +
d*x)**2/(4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4*I*b**2*d) +
 6*I*B*d*x*tan(c + d*x)/(4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x)
 - 4*I*b**2*d) + 3*B*d*x/(4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x
) - 4*I*b**2*d) + 2*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(4*I*b**2*
d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4*I*b**2*d) + 4*B*log(tan(c + d
*x)**2 + 1)*tan(c + d*x)/(4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x
) - 4*I*b**2*d) - 2*I*B*log(tan(c + d*x)**2 + 1)/(4*I*b**2*d*tan(c + d*x)**
2 + 8*b**2*d*tan(c + d*x) - 4*I*b**2*d) + 5*B*tan(c + d*x)/(4*I*b**2*d*tan(
c + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4*I*b**2*d) - 4*I*B/(4*I*b**2*d*tan(c
 + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4*I*b**2*d) - 9*I*C*d*x*tan(c + d*x)**
2/(4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4*I*b**2*d) - 18*C*
d*x*tan(c + d*x)/(4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4*I*
b**2*d) + 9*I*C*d*x/(4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4
*I*b**2*d) - 4*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(4*I*b**2*d*tan(c
 + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4*I*b**2*d) + 8*I*C*log(tan(c + d*x)**
2 + 1)*tan(c + d*x)/(4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4
*I*b**2*d) + 4*C*log(tan(c + d*x)**2 + 1)/(4*I*b**2*d*tan(c + d*x)**2 + 8*b
**2*d*tan(c + d*x) - 4*I*b**2*d) + 4*I*C*tan(c + d*x)**3/(4*I*b**2*d*tan(c
+ d*x)**2 + 8*b**2*d*tan(c + d*x) - 4*I*b**2*d) + 19*I*C*tan(c + d*x)/(4*I*
b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4*I*b**2*d) + 14*C/(4*I*b*
*2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) - 4*I*b**2*d), Eq(a, -I*b)), (
-3*B*d*x*tan(c + d*x)**2/(-4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*
x) + 4*I*b**2*d) - 6*I*B*d*x*tan(c + d*x)/(-4*I*b**2*d*tan(c + d*x)**2 + 8*
b**2*d*tan(c + d*x) + 4*I*b**2*d) + 3*B*d*x/(-4*I*b**2*d*tan(c + d*x)**2 +
8*b**2*d*tan(c + d*x) + 4*I*b**2*d) - 2*I*B*log(tan(c + d*x)**2 + 1)*tan(c
+ d*x)**2/(-4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) + 4*I*b**2*d
) + 4*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(-4*I*b**2*d*tan(c + d*x)**2
+ 8*b**2*d*tan(c + d*x) + 4*I*b**2*d) + 2*I*B*log(tan(c + d*x)**2 + 1)/(-4*
```

```
I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) + 4*I*b**2*d) + 5*B*tan(c
+ d*x)/(-4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) + 4*I*b**2*d) +
 4*I*B/(-4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) + 4*I*b**2*d) +
 9*I*C*d*x*tan(c + d*x)**2/(-4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c +
d*x) + 4*I*b**2*d) - 18*C*d*x*tan(c + d*x)/(-4*I*b**2*d*tan(c + d*x)**2 + 8
*b**2*d*tan(c + d*x) + 4*I*b**2*d) - 9*I*C*d*x/(-4*I*b**2*d*tan(c + d*x)**2
 + 8*b**2*d*tan(c + d*x) + 4*I*b**2*d) - 4*C*log(tan(c + d*x)**2 + 1)*tan(c
 + d*x)**2/(-4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) + 4*I*b**2*
d) - 8*I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(-4*I*b**2*d*tan(c + d*x)*
*2 + 8*b**2*d*tan(c + d*x) + 4*I*b**2*d) + 4*C*log(tan(c + d*x)**2 + 1)/(-4
*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) + 4*I*b**2*d) - 4*I*C*tan
(c + d*x)**3/(-4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d*x) + 4*I*b**
2*d) - 19*I*C*tan(c + d*x)/(-4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c +
d*x) + 4*I*b**2*d) + 14*C/(-4*I*b**2*d*tan(c + d*x)**2 + 8*b**2*d*tan(c + d
*x) + 4*I*b**2*d), Eq(a, I*b)), (x*(B*tan(c) + C*tan(c)**2)*tan(c)**2/(a +
b*tan(c))**2, Eq(d, 0)), (2*B*a**5*b*log(a/b + tan(c + d*x))/(2*a**5*b**3*d
 + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x)
+ 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) + 2*B*a**5*b/(2*a**5*b**3*d + 2*a**4*
b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7
*d + 2*b**8*d*tan(c + d*x)) + 2*B*a**4*b**2*log(a/b + tan(c + d*x))*tan(c +
 d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*
b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) + 6*B*a**3*b**3*l
og(a/b + tan(c + d*x))/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3
*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x))
- B*a**3*b**3*log(tan(c + d*x)**2 + 1)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c
 + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*
d*tan(c + d*x)) + 2*B*a**3*b**3/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x)
 + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c
 + d*x)) - 4*B*a**2*b**4*d*x/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) +
4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c +
d*x)) + 6*B*a**2*b**4*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b**3*d +
 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) +
2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - B*a**2*b**4*log(tan(c + d*x)**2 + 1)*
tan(c + d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d +
4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 4*B*a*b*
*5*d*x*tan(c + d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b*
*5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) + B
*a*b**5*log(tan(c + d*x)**2 + 1)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x
) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(
c + d*x)) + B*b**6*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*b**3*d + 2
*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*
a*b**7*d + 2*b**8*d*tan(c + d*x)) - 4*C*a**6*log(a/b + tan(c + d*x))/(2*a**
5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c
 + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 4*C*a**6/(2*a**5*b**3*d + 2
*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*
```

```
a*b**7*d + 2*b**8*d*tan(c + d*x)) - 4*C*a**5*b*log(a/b + tan(c + d*x))*tan(
c + d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a*
*2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 8*C*a**4*b**
2*log(a/b + tan(c + d*x))/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a
**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x
)) + 2*C*a**4*b**2*tan(c + d*x)**2/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d
*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*ta
n(c + d*x)) - 6*C*a**4*b**2/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4
*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d
*x)) + 2*C*a**3*b**3*d*x/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a*
*3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)
) - 8*C*a**3*b**3*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b**3*d + 2*a
**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*
b**7*d + 2*b**8*d*tan(c + d*x)) + 2*C*a**2*b**4*d*x*tan(c + d*x)/(2*a**5*b*
*3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d
*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 2*C*a**2*b**4*log(tan(c + d*x)*
*2 + 1)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**
2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) + 4*C*a**2*b**4
*tan(c + d*x)**2/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*
d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 2*C*
a**2*b**4/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a
**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 2*C*a*b**5*
d*x/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x) + 4*a**3*b**5*d + 4*a**2*b*
*6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c + d*x)) - 2*C*a*b**5*log(ta
n(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c + d*x)
 + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*tan(c
 + d*x)) - 2*C*b**6*d*x*tan(c + d*x)/(2*a**5*b**3*d + 2*a**4*b**4*d*tan(c +
 d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**8*d*
tan(c + d*x)) + 2*C*b**6*tan(c + d*x)**2/(2*a**5*b**3*d + 2*a**4*b**4*d*tan
(c + d*x) + 4*a**3*b**5*d + 4*a**2*b**6*d*tan(c + d*x) + 2*a*b**7*d + 2*b**
8*d*tan(c + d*x)), True))
```

3.33 $\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$

Optimal. Leaf size=157

$$-\frac{a^2(bB-aC)}{b^2d\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{\left(a^2(-C)+2abB+b^2C\right)\log(\cos(c+dx))}{d\left(a^2+b^2\right)^2}-\frac{x\left(a^2B+2abC-b^2B\right)}{\left(a^2+b^2\right)^2}-\frac{a\left(a^3(-C\right.}{}$$

```
[Out] -(B*a^2-B*b^2+2*C*a*b)*x/(a^2+b^2)^2-(2*B*a*b-C*a^2+C*b^2)*ln(cos(d*x+c))/(
a^2+b^2)^2/d-a*(2*B*b^3-C*a^3-3*C*a*b^2)*ln(a+b*tan(d*x+c))/b^2/(a^2+b^2)^2
/d-a^2*(B*b-C*a)/b^2/(a^2+b^2)/d/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.31, antiderivative size = 157, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.158, Rules used = {3632, 3604, 3626, 3617, 31, 3475}

$$-\frac{a^2(bB-aC)}{b^2d\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{a\left(a^3(-C)-3ab^2C+2b^3B\right)\log(a+b\tan(c+dx))}{b^2d\left(a^2+b^2\right)^2}-\frac{\left(a^2(-C)+2abB+b^2C\right)}{d\left(a^2+b^2\right.}$$

Antiderivative was successfully verified.

```
[In] Int[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
^2,x]
```

```
[Out] -(((a^2*B - b^2*B + 2*a*b*C)*x)/(a^2 + b^2)^2) - ((2*a*b*B - a^2*C + b^2*C)
*Log[Cos[c + d*x]])/((a^2 + b^2)^2*d) - (a*(2*b^3*B - a^3*C - 3*a*b^2*C)*Lo
g[a + b*Tan[c + d*x]])/(b^2*(a^2 + b^2)^2*d) - (a^2*(b*B - a*C))/(b^2*(a^2
+ b^2)*d*(a + b*Tan[c + d*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3604

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> -Simp
[((B*c - A*d)*(b*c - a*d)^2*(c + d*Tan[e + f*x])^(n + 1))/(f*d^2*(n + 1)*(c
```

```
^2 + d^2)), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*S
imp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c +
 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)*T
an[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c
- a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \int \frac{\tan^2(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^2}\,dx$$

$$= -\frac{a^2(bB-aC)}{b^2\left(a^2+b^2\right)d(a+b\tan(c+dx))} + \frac{\int \frac{-a(bB-aC)+b(bB-aC)\text{t}}{a+}}{b}$$

$$= -\frac{\left(a^2B-b^2B+2abC\right)x}{\left(a^2+b^2\right)^2} - \frac{a^2(bB-aC)}{b^2\left(a^2+b^2\right)d(a+b\tan(c+dx}$$

$$= -\frac{\left(a^2B-b^2B+2abC\right)x}{\left(a^2+b^2\right)^2} - \frac{\left(2abB-a^2C+b^2C\right)\log(\cos(c}{\left(a^2+b^2\right)^2 d}$$

$$= -\frac{\left(a^2B-b^2B+2abC\right)x}{\left(a^2+b^2\right)^2} - \frac{\left(2abB-a^2C+b^2C\right)\log(\cos(c}{\left(a^2+b^2\right)^2 d}$$

Mathematica [C] time = 2.18, size = 324, normalized size = 2.06

$$\frac{-2ia\left(a^3C+3ab^2C-2b^3B\right)\tan^{-1}(\tan(c+dx))(a+b\tan(c+dx))+a\left(a\left(a^3C+3ab^2C-2b^3B\right)\log\left((a\cos(c+\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x])^2,x]
```

```
[Out] (a*(2*(a + I*b)^2*(-(b^2*B) + I*a^2*C + 2*a*b*C)*(c + d*x) - 2*(a^2 + b^2)^
2*C*Log[Cos[c + d*x]] + a*(-2*b^3*B + a^3*C + 3*a*b^2*C)*Log[(a*Cos[c + d*x
] + b*Sin[c + d*x])^2]) + b*(2*(a + I*b)*((-I)*b^3*B*(c + d*x) + I*a^3*C*(I
 + c + d*x) - a*b^2*((-2*I)*C*(c + d*x) + B*(I + c + d*x)) + a^2*b*(B + C*(
I + c + d*x))) - 2*(a^2 + b^2)^2*C*Log[Cos[c + d*x]] + a*(-2*b^3*B + a^3*C
+ 3*a*b^2*C)*Log[(a*Cos[c + d*x] + b*Sin[c + d*x])^2])*Tan[c + d*x] - (2*I)
*a*(-2*b^3*B + a^3*C + 3*a*b^2*C)*ArcTan[Tan[c + d*x]]*(a + b*Tan[c + d*x])
)/(2*b^2*(a^2 + b^2)^2*d*(a + b*Tan[c + d*x]))
```

fricas [B] time = 0.82, size = 311, normalized size = 1.98

$$\frac{2\,Ca^3b^2-2\,Ba^2b^3-2\left(Ba^3b^2+2\,Ca^2b^3-Bab^4\right)dx+\left(Ca^5+3\,Ca^3b^2-2\,Ba^2b^3+\left(Ca^4b+3\,Ca^2b^3-2\,Bab^4\right)\text{ta}\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="fricas")
```

```
[Out] 1/2*(2*C*a^3*b^2 - 2*B*a^2*b^3 - 2*(B*a^3*b^2 + 2*C*a^2*b^3 - B*a*b^4)*d*x
+ (C*a^5 + 3*C*a^3*b^2 - 2*B*a^2*b^3 + (C*a^4*b + 3*C*a^2*b^3 - 2*B*a*b^4)*
tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x
+ c)^2 + 1)) - (C*a^5 + 2*C*a^3*b^2 + C*a*b^4 + (C*a^4*b + 2*C*a^2*b^3 + C*
b^5)*tan(d*x + c))*log(1/(tan(d*x + c)^2 + 1)) - 2*(C*a^4*b - B*a^3*b^2 + (
B*a^2*b^3 + 2*C*a*b^4 - B*b^5)*d*x)*tan(d*x + c))/((a^4*b^3 + 2*a^2*b^5 + b
^7)*d*tan(d*x + c) + (a^5*b^2 + 2*a^3*b^4 + a*b^6)*d)
```

giac [A] time = 2.02, size = 244, normalized size = 1.55

$$-\frac{\frac{2\left(Ba^2+2\,Cab-Bb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}+\frac{\left(Ca^2-2\,Bab-Cb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(Ca^4+3\,Ca^2b^2-2\,Bab^3\right)\log(|b\tan(dx+c)+a|)}{a^4b^2+2\,a^2b^4+b^6}+\frac{2\left(Ca^4\tan(dx+c)+3\,Ca^2b\right.}{\left(a^4\right.}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="giac")
```

```
[Out] -1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + (C*a^
2 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2*(C
*a^4 + 3*C*a^2*b^2 - 2*B*a*b^3)*log(abs(b*tan(d*x + c) + a))/(a^4*b^2 + 2*a
^2*b^4 + b^6) + 2*(C*a^4*tan(d*x + c) + 3*C*a^2*b^2*tan(d*x + c) - 2*B*a*b^
3*tan(d*x + c) + B*a^4 + 2*C*a^3*b - B*a^2*b^2)/((a^4*b + 2*a^2*b^3 + b^5)*
(b*tan(d*x + c) + a)))/d
```

maple [A] time = 0.32, size = 313, normalized size = 1.99

$$-\frac{a^2B}{db\left(a^2+b^2\right)(a+b\tan(dx+c))}+\frac{a^3C}{d\,b^2\left(a^2+b^2\right)(a+b\tan(dx+c))}-\frac{2ab\ln(a+b\tan(dx+c))\,B}{d\left(a^2+b^2\right)^2}+\frac{a^4\ln(a+b\,\mathrm{t}}{d\left(a^2+\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x)
```

```
[Out] -1/d*a^2/b/(a^2+b^2)/(a+b*tan(d*x+c))*B+1/d*a^3/b^2/(a^2+b^2)/(a+b*tan(d*x+
c))*C-2/d*a/(a^2+b^2)^2*b*ln(a+b*tan(d*x+c))*B+1/d*a^4/(a^2+b^2)^2/b^2*ln(a
+b*tan(d*x+c))*C+3/d*a^2/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*C+1/d/(a^2+b^2)^2*l
n(1+tan(d*x+c)^2)*B*a*b-1/2/d/(a^2+b^2)^2*ln(1+tan(d*x+c)^2)*a^2*C+1/2/d/(a
^2+b^2)^2*ln(1+tan(d*x+c)^2)*b^2*C-1/d/(a^2+b^2)^2*B*arctan(tan(d*x+c))*a^2
```

```
+1/d/(a^2+b^2)^2*B*arctan(tan(d*x+c))*b^2-2/d/(a^2+b^2)^2*C*arctan(tan(d*x+
c))*a*b
```

maxima [A] time = 0.61, size = 197, normalized size = 1.25

$$-\frac{\frac{2\left(Ba^2+2\,Cab-Bb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(Ca^4+3\,Ca^2b^2-2\,Bab^3\right)\log(b\tan(dx+c)+a)}{a^4b^2+2\,a^2b^4+b^6}+\frac{\left(Ca^2-2\,Bab-Cb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(Ca^3-Ba^2b\right)}{a^3b^2+ab^4+\left(a^2b^3+b^5\right)\tan}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - 2*(C*
a^4 + 3*C*a^2*b^2 - 2*B*a*b^3)*log(b*tan(d*x + c) + a)/(a^4*b^2 + 2*a^2*b^4
 + b^6) + (C*a^2 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^
2 + b^4) - 2*(C*a^3 - B*a^2*b)/(a^3*b^2 + a*b^4 + (a^2*b^3 + b^5)*tan(d*x +
 c)))/d
```

mupad [B] time = 9.11, size = 165, normalized size = 1.05

$$\frac{\ln\left(\tan\left(c+d\,x\right)+1{}\mathrm{i}\right)\,\left(C+B\,1{}\mathrm{i}\right)}{2\,d\,\left(-a^2+a\,b\,2{}\mathrm{i}+b^2\right)}+\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(B+C\,1{}\mathrm{i}\right)}{2\,d\,\left(-a^2\,1{}\mathrm{i}+2\,a\,b+b^2\,1{}\mathrm{i}\right)}-\frac{a^2\,\left(B\,b-C\,a\right)}{b^2\,d\,\left(a^2+b^2\right)\,\left(a+b\,\tan\left(c+d\,x\right)\right)}+\frac{a\,\ln\left(a+\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((tan(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B*1i + C))/(2*d*(a*b*2i - a^2 + b^2)) + (log(tan(c
 + d*x) - 1i)*(B + C*1i))/(2*d*(2*a*b - a^2*1i + b^2*1i)) - (a^2*(B*b - C*a
))/(b^2*d*(a^2 + b^2)*(a + b*tan(c + d*x))) + (a*log(a + b*tan(c + d*x))*(C
*a^3 - 2*B*b^3 + 3*C*a*b^2))/(b^2*d*(a^2 + b^2)^2)
```

sympy [A] time = 2.29, size = 3497, normalized size = 22.27

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)/tan(c), Eq(a, 0) & Eq(b, 0) & Eq(
d, 0)), ((-B*x + B*tan(c + d*x)/d - C*log(tan(c + d*x)**2 + 1)/(2*d) + C*ta
n(c + d*x)**2/(2*d))/a**2, Eq(b, 0)), (-B*d*x*tan(c + d*x)**2/(-4*b**2*d*ta
```

```
n(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + 2*I*B*d*x*tan(c + d*x
)/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + B*d*x/
(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + 3*B*tan(
c + d*x)/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) -
 2*I*B/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) - 3
*I*C*d*x*tan(c + d*x)**2/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*
x) + 4*b**2*d) - 6*C*d*x*tan(c + d*x)/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2
*d*tan(c + d*x) + 4*b**2*d) + 3*I*C*d*x/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b*
*2*d*tan(c + d*x) + 4*b**2*d) - 2*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**
2/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + 4*I*C*
log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2
*d*tan(c + d*x) + 4*b**2*d) + 2*C*log(tan(c + d*x)**2 + 1)/(-4*b**2*d*tan(c
 + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + 5*I*C*tan(c + d*x)/(-4*b
**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + 4*C/(-4*b**2*
d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d), Eq(a, -I*b)), (-B*
d*x*tan(c + d*x)**2/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) +
4*b**2*d) - 2*I*B*d*x*tan(c + d*x)/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*
tan(c + d*x) + 4*b**2*d) + B*d*x/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*ta
n(c + d*x) + 4*b**2*d) + 3*B*tan(c + d*x)/(-4*b**2*d*tan(c + d*x)**2 - 8*I*
b**2*d*tan(c + d*x) + 4*b**2*d) + 2*I*B/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b*
*2*d*tan(c + d*x) + 4*b**2*d) + 3*I*C*d*x*tan(c + d*x)**2/(-4*b**2*d*tan(c
+ d*x)**2 - 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) - 6*C*d*x*tan(c + d*x)/(-4*
b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) - 3*I*C*d*x/(-
4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) - 2*C*log(ta
n(c + d*x)**2 + 1)*tan(c + d*x)**2/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*
tan(c + d*x) + 4*b**2*d) - 4*I*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(-4*
b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + 2*C*log(tan(
c + d*x)**2 + 1)/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) + 4*b
**2*d) - 5*I*C*tan(c + d*x)/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c +
 d*x) + 4*b**2*d) + 4*C/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x
) + 4*b**2*d), Eq(a, I*b)), (x*(B*tan(c) + C*tan(c)**2)*tan(c)/(a + b*tan(c
))**2, Eq(d, 0)), (-2*B*a**4*b/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x)
+ 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c
+ d*x)) - 2*B*a**3*b**2*d*x/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4
*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d
*x)) - 2*B*a**2*b**3*d*x*tan(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c
+ d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d
*tan(c + d*x)) - 4*B*a**2*b**3*log(a/b + tan(c + d*x))/(2*a**5*b**2*d + 2*a
**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*
b**6*d + 2*b**7*d*tan(c + d*x)) + 2*B*a**2*b**3*log(tan(c + d*x)**2 + 1)/(2
*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*t
an(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) - 2*B*a**2*b**3/(2*a**5*b
**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c +
d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + 2*B*a*b**4*d*x/(2*a**5*b**2*d
+ 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) +
```

```
 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) - 4*B*a*b**4*log(a/b + tan(c + d*x))*t
an(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4
*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + 2*B*a*b**
4*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(
c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7
*d*tan(c + d*x)) + 2*B*b**5*d*x*tan(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d
*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2
*b**7*d*tan(c + d*x)) + 2*C*a**5*log(a/b + tan(c + d*x))/(2*a**5*b**2*d + 2
*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*
a*b**6*d + 2*b**7*d*tan(c + d*x)) + 2*C*a**5/(2*a**5*b**2*d + 2*a**4*b**3*d
*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2
*b**7*d*tan(c + d*x)) + 2*C*a**4*b*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*
a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*ta
n(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + 6*C*a**3*b**2*log(a/b +
tan(c + d*x))/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d +
 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) - C*a**3*
b**2*log(tan(c + d*x)**2 + 1)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) +
 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c +
 d*x)) + 2*C*a**3*b**2/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3
*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x))
- 4*C*a**2*b**3*d*x/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b*
*4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + 6
*C*a**2*b**3*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b**2*d + 2*a**4*b
**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*
d + 2*b**7*d*tan(c + d*x)) - C*a**2*b**3*log(tan(c + d*x)**2 + 1)*tan(c + d
*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b*
*5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) - 4*C*a*b**4*d*x*ta
n(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*
a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x)) + C*a*b**4*l
og(tan(c + d*x)**2 + 1)/(2*a**5*b**2*d + 2*a**4*b**3*d*tan(c + d*x) + 4*a**
3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d + 2*b**7*d*tan(c + d*x))
 + C*b**5*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*b**2*d + 2*a**4*b**
3*d*tan(c + d*x) + 4*a**3*b**4*d + 4*a**2*b**5*d*tan(c + d*x) + 2*a*b**6*d
+ 2*b**7*d*tan(c + d*x)), True))
```

3.34 $\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx$

Optimal. Leaf size=115

$$\frac{a(bB-aC)}{bd\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{\left(a^2B+2abC-b^2B\right)\log(a\cos(c+dx)+b\sin(c+dx))}{d\left(a^2+b^2\right)^2}+\frac{x\left(a^2(-C)+2abB+b^2C\right.}{\left(a^2+b^2\right)^2}$$

```
[Out] (2*B*a*b-C*a^2+C*b^2)*x/(a^2+b^2)^2-(B*a^2-B*b^2+2*C*a*b)*ln(a*cos(d*x+c)+b
*sin(d*x+c))/(a^2+b^2)^2/d+a*(B*b-C*a)/b/(a^2+b^2)/d/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.15, antiderivative size = 115, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, integrand size = 32, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.094, Rules used = {3628, 3531, 3530}

$$\frac{a(bB-aC)}{bd\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{\left(a^2B+2abC-b^2B\right)\log(a\cos(c+dx)+b\sin(c+dx))}{d\left(a^2+b^2\right)^2}+\frac{x\left(a^2(-C)+2abB+b^2C\right.}{\left(a^2+b^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x])^2,x]
```

```
[Out] ((2*a*b*B - a^2*C + b^2*C)*x)/(a^2 + b^2)^2 - ((a^2*B - b^2*B + 2*a*b*C)*Lo
g[a*Cos[c + d*x] + b*Sin[c + d*x]])/((a^2 + b^2)^2*d) + (a*(b*B - a*C))/(b*
(a^2 + b^2)*d*(a + b*Tan[c + d*x]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
```

```
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^2}\,dx = \frac{a(bB-aC)}{b\left(a^2+b^2\right)d(a+b\tan(c+dx))}+\frac{\int \frac{bB-aC+(aB+bC)\tan(c+dx)}{a+b\tan(c+dx)}\,dx}{a^2+b^2}$$
$$= \frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2}+\frac{a(bB-aC)}{b\left(a^2+b^2\right)d(a+b\tan(c+dx))}-\frac{\left(a^2B-b^2B\right.}{}$$
$$= \frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2}-\frac{\left(a^2B-b^2B+2abC\right)\log(a\cos(c+dx)+b\sin(}{\left(a^2+b^2\right)^2 d}$$

Mathematica [C] time = 2.19, size = 140, normalized size = 1.22

$$\frac{\frac{2\left(\left(a^2(-B)-2abC+b^2B\right)\log(a+b\tan(c+dx))-\frac{a\left(a^2+b^2\right)(aC-bB)}{b(a+b\tan(c+dx))}\right)}{\left(a^2+b^2\right)^2}+\frac{(B+iC)\log(-\tan(c+dx)+i)}{(a+ib)^2}+\frac{(B-iC)\log(\tan(c+dx)+i)}{(a-ib)^2}}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x])^2,x]
```

```
[Out] (((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^2 + ((B - I*C)*Log[I + Tan[c +
 d*x]])/(a - I*b)^2 + (2*((-(a^2*B) + b^2*B - 2*a*b*C)*Log[a + b*Tan[c + d*
x]] - (a*(a^2 + b^2)*(-(b*B) + a*C))/(b*(a + b*Tan[c + d*x]))))/(a^2 + b^2)
^2)/(2*d)
```

fricas [A] time = 0.59, size = 221, normalized size = 1.92

$$-\frac{2\,Ca^2b-2\,Bab^2+2\left(Ca^3-2\,Ba^2b-Cab^2\right)dx+\left(Ba^3+2\,Ca^2b-Bab^2+\left(Ba^2b+2\,Cab^2-Bb^3\right)\tan\left(dx+c\right)\right)}{2\left(\left(a^4b+2\,a^2b^3+b^5\right)d\tan\left(dx+c\right)+\left(a\right.\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, algorithm="fr
icas")
```

```
[Out] -1/2*(2*C*a^2*b - 2*B*a*b^2 + 2*(C*a^3 - 2*B*a^2*b - C*a*b^2)*d*x + (B*a^3
+ 2*C*a^2*b - B*a*b^2 + (B*a^2*b + 2*C*a*b^2 - B*b^3)*tan(d*x + c))*log((b^
2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1)) - 2*(C*a
^3 - B*a^2*b - (C*a^2*b - 2*B*a*b^2 - C*b^3)*d*x)*tan(d*x + c))/((a^4*b + 2
*a^2*b^3 + b^5)*d*tan(d*x + c) + (a^5 + 2*a^3*b^2 + a*b^4)*d)
```

giac [B] time = 1.99, size = 241, normalized size = 2.10

$$-\frac{\frac{2\left(Ca^2-2\,Bab-Cb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}-\frac{\left(Ba^2+2\,Cab-Bb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}+\frac{2\left(Ba^2b+2\,Cab^2-Bb^3\right)\log(|b\tan(dx+c)+a|)}{a^4b+2\,a^2b^3+b^5}-\frac{2\left(Ba^2b^2\tan(dx+c)+2\,Cab^3\tan\right.}{\left(a^4b+2\right.}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, algorithm="gi
ac")
```

```
[Out] -1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - (B*a^
2 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) + 2*(B
*a^2*b + 2*C*a*b^2 - B*b^3)*log(abs(b*tan(d*x + c) + a))/(a^4*b + 2*a^2*b^3
 + b^5) - 2*(B*a^2*b^2*tan(d*x + c) + 2*C*a*b^3*tan(d*x + c) - B*b^4*tan(d*
x + c) - C*a^4 + 2*B*a^3*b + C*a^2*b^2)/((a^4*b + 2*a^2*b^3 + b^5)*(b*tan(d
*x + c) + a)))/d
```

maple [B] time = 0.31, size = 305, normalized size = 2.65

$$\frac{aB}{d\left(a^2+b^2\right)(a+b\tan(dx+c))}-\frac{a^2C}{d\left(a^2+b^2\right)b\,(a+b\tan(dx+c))}-\frac{a^2\ln(a+b\tan(dx+c))\,B}{d\left(a^2+b^2\right)^2}+\frac{\ln(a+b\tan(dx+}{d\left(a^2+b^2\right)^{}}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x)
```

```
[Out] 1/d*a/(a^2+b^2)/(a+b*tan(d*x+c))*B-1/d*a^2/(a^2+b^2)/b/(a+b*tan(d*x+c))*C-1
/d*a^2/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*B+1/d/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*
b^2*B-2/d/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*C*a*b+1/2/d/(a^2+b^2)^2*ln(1+tan(d
*x+c)^2)*a^2*B-1/2/d/(a^2+b^2)^2*ln(1+tan(d*x+c)^2)*b^2*B+1/d/(a^2+b^2)^2*l
n(1+tan(d*x+c)^2)*C*a*b+2/d/(a^2+b^2)^2*B*arctan(tan(d*x+c))*a*b-1/d/(a^2+b
^2)^2*C*arctan(tan(d*x+c))*a^2+1/d/(a^2+b^2)^2*C*arctan(tan(d*x+c))*b^2
```

maxima [A] time = 0.64, size = 185, normalized size = 1.61

$$-\frac{\frac{2\left(Ca^2-2\,Bab-Cb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}+\frac{2\left(Ba^2+2\,Cab-Bb^2\right)\log(b\tan(dx+c)+a)}{a^4+2\,a^2b^2+b^4}-\frac{\left(Ba^2+2\,Cab-Bb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}+\frac{2\left(Ca^2-Bab\right)}{a^3b+ab^3+\left(a^2b^2+b^4\right)\tan(dx+c)}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, algorithm="ma
xima")
```

```
[Out] -1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + 2*(B*
a^2 + 2*C*a*b - B*b^2)*log(b*tan(d*x + c) + a)/(a^4 + 2*a^2*b^2 + b^4) - (B
*a^2 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) + 2
*(C*a^2 - B*a*b)/(a^3*b + a*b^3 + (a^2*b^2 + b^4)*tan(d*x + c)))/d
```

mupad [B] time = 9.01, size = 163, normalized size = 1.42

$$\frac{a\,\left(B\,b-C\,a\right)}{b\,d\,\left(a^2+b^2\right)\,\left(a+b\,\tan\left(c+d\,x\right)\right)}+\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)}{2\,d\,\left(a^2+a\,b\,2\mathrm{i}-b^2\right)}+\frac{\ln\left(\tan\left(c+d\,x\right)+1\mathrm{i}\right)\,\left(C+B\,1\mathrm{i}\right)}{2\,d\,\left(a^2\,1\mathrm{i}+2\,a\,b-b^2\,1\mathrm{i}\right)}-\frac{\ln\left(a+b\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)/(a + b*tan(c + d*x))^2,x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B + C*1i))/(2*d*(a*b*2i + a^2 - b^2)) - (log(a + b
*tan(c + d*x))*(B/(a^2 + b^2) - (2*b*(B*b - C*a))/(a^2 + b^2)^2))/d + (log(
tan(c + d*x) + 1i)*(B*1i + C))/(2*d*(2*a*b + a^2*1i - b^2*1i)) + (a*(B*b -
C*a))/(b*d*(a^2 + b^2)*(a + b*tan(c + d*x)))
```

sympy [A] time = 1.84, size = 2995, normalized size = 26.04

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)/tan(c)**2, Eq(a, 0) & Eq(b, 0) &
Eq(d, 0)), ((B*log(tan(c + d*x)**2 + 1)/(2*d) - C*x + C*tan(c + d*x)/d)/a**
2, Eq(b, 0)), (I*B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2
*d*tan(c + d*x) - 4*b**2*d) + 2*B*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**
2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - I*B*d*x/(4*b**2*d*tan(c + d*x)**2
 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + I*B*tan(c + d*x)/(4*b**2*d*tan(c +
 d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + C*d*x*tan(c + d*x)**2/(4*b
**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 2*I*C*d*x*tan
(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) -
 C*d*x/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 3*
C*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2
*d) + 2*I*C/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d)
, Eq(a, -I*b)), (-I*B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b
**2*d*tan(c + d*x) - 4*b**2*d) + 2*B*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x
)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + I*B*d*x/(4*b**2*d*tan(c + d*x)
```

```
**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - I*B*tan(c + d*x)/(4*b**2*d*tan(
c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + C*d*x*tan(c + d*x)**2/(
4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*I*C*d*x*
tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d
) - C*d*x/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) -
 3*C*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b
**2*d) - 2*I*C/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2
*d), Eq(a, I*b)), (x*(B*tan(c) + C*tan(c)**2)/(a + b*tan(c))**2, Eq(d, 0)),
 (-2*B*a**3*b*log(a/b + tan(c + d*x))/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d
*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*ta
n(c + d*x)) + B*a**3*b*log(tan(c + d*x)**2 + 1)/(2*a**5*b*d + 2*a**4*b**2*d
*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2
*b**6*d*tan(c + d*x)) + 2*B*a**3*b/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x)
 + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c
 + d*x)) + 4*B*a**2*b**2*d*x/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a
**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x
)) - 2*B*a**2*b**2*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b*d + 2*a**
4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b*
*5*d + 2*b**6*d*tan(c + d*x)) + B*a**2*b**2*log(tan(c + d*x)**2 + 1)*tan(c
+ d*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b*
*4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) + 4*B*a*b**3*d*x*ta
n(c + d*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**
2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) + 2*B*a*b**3*lo
g(a/b + tan(c + d*x))/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**
3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) - B*
a*b**3*log(tan(c + d*x)**2 + 1)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) +
4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c +
d*x)) + 2*B*a*b**3/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d
 + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) + 2*B*b
**4*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c
+ d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d
*tan(c + d*x)) - B*b**4*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*b*d +
 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) +
2*a*b**5*d + 2*b**6*d*tan(c + d*x)) - 2*C*a**4/(2*a**5*b*d + 2*a**4*b**2*d*
tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*
b**6*d*tan(c + d*x)) - 2*C*a**3*b*d*x/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d
*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*ta
n(c + d*x)) - 2*C*a**2*b**2*d*x*tan(c + d*x)/(2*a**5*b*d + 2*a**4*b**2*d*ta
n(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b*
*6*d*tan(c + d*x)) - 4*C*a**2*b**2*log(a/b + tan(c + d*x))/(2*a**5*b*d + 2*
a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a
*b**5*d + 2*b**6*d*tan(c + d*x)) + 2*C*a**2*b**2*log(tan(c + d*x)**2 + 1)/(
2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan
(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) - 2*C*a**2*b**2/(2*a**5*b*d
 + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x)
```

```
+ 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) + 2*C*a*b**3*d*x/(2*a**5*b*d + 2*a**4
*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**
5*d + 2*b**6*d*tan(c + d*x)) - 4*C*a*b**3*log(a/b + tan(c + d*x))*tan(c + d
*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a**3*b**3*d + 4*a**2*b**4*
d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x)) + 2*C*a*b**3*log(tan(c
 + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4*a
**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d*x
)) + 2*C*b**4*d*x*tan(c + d*x)/(2*a**5*b*d + 2*a**4*b**2*d*tan(c + d*x) + 4
*a**3*b**3*d + 4*a**2*b**4*d*tan(c + d*x) + 2*a*b**5*d + 2*b**6*d*tan(c + d
*x)), True))
```

3.35 $\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$

Optimal. Leaf size=111

$$-\frac{bB-aC}{d\left(a^2+b^2\right)(a+b\tan(c+dx))}+\frac{\left(a^2(-C)+2abB+b^2C\right)\log(a\cos(c+dx)+b\sin(c+dx))}{d\left(a^2+b^2\right)^2}+\frac{x\left(a^2B+2abC-b^2\right.}{\left(a^2+b^2\right)^2}$$

```
[Out] (B*a^2-B*b^2+2*C*a*b)*x/(a^2+b^2)^2+(2*B*a*b-C*a^2+C*b^2)*ln(a*cos(d*x+c)+b
*sin(d*x+c))/(a^2+b^2)^2/d+(-B*b+C*a)/(a^2+b^2)/d/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.21, antiderivative size = 111, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3632, 3529, 3531, 3530}

$$-\frac{bB-aC}{d\left(a^2+b^2\right)(a+b\tan(c+dx))}+\frac{\left(a^2(-C)+2abB+b^2C\right)\log(a\cos(c+dx)+b\sin(c+dx))}{d\left(a^2+b^2\right)^2}+\frac{x\left(a^2B+2abC-b^2\right.}{\left(a^2+b^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
^2,x]
```

```
[Out] ((a^2*B - b^2*B + 2*a*b*C)*x)/(a^2 + b^2)^2 + ((2*a*b*B - a^2*C + b^2*C)*Lo
g[a*Cos[c + d*x] + b*Sin[c + d*x]])/((a^2 + b^2)^2*d) - (b*B - a*C)/((a^2 +
 b^2)*d*(a + b*Tan[c + d*x]))
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \int \frac{B+C\tan(c+dx)}{(a+b\tan(c+dx))^2}\,dx$$

$$= -\frac{bB-aC}{\left(a^2+b^2\right)d(a+b\tan(c+dx))} + \frac{\int \frac{aB+bC-(bB-aC)\tan(c+dx)}{a+b\tan(c+dx)}}{a^2+b^2}$$

$$= \frac{\left(a^2B-b^2B+2abC\right)x}{\left(a^2+b^2\right)^2} - \frac{bB-aC}{\left(a^2+b^2\right)d(a+b\tan(c+dx))} +$$

$$= \frac{\left(a^2B-b^2B+2abC\right)x}{\left(a^2+b^2\right)^2} + \frac{\left(2abB-a^2C+b^2C\right)\log(a\cos(c}{\left(a^2+b^2\right)^2}$$

Mathematica [C] time = 2.22, size = 190, normalized size = 1.71

$$\frac{\frac{C((-b-ia)\log(-\tan(c+dx)+i)+i(a+ib)\log(\tan(c+dx)+i)+2b\log(a+b\tan(c+dx)))}{a^2+b^2} - (bB-aC)\left(\frac{2b\left(\frac{a^2+b^2}{a+b\tan(c+dx)}-2a\log(a+b\tan(c+dx))\right)}{\left(a^2+b^2\right)^2} + \frac{i\log}{}\right.}{2bd}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x])^2,x]
```

```
[Out] ((C*(((-I)*a - b)*Log[I - Tan[c + d*x]] + I*(a + I*b)*Log[I + Tan[c + d*x]]
 + 2*b*Log[a + b*Tan[c + d*x]]))/(a^2 + b^2) - (b*B - a*C)*((I*Log[I - Tan[
c + d*x]])/(a + I*b)^2 - (I*Log[I + Tan[c + d*x]])/(a - I*b)^2 + (2*b*(-2*a
*Log[a + b*Tan[c + d*x]] + (a^2 + b^2)/(a + b*Tan[c + d*x])))/(a^2 + b^2)^2
))/(2*b*d)
```

fricas [A] time = 0.80, size = 222, normalized size = 2.00

$$\frac{2\,Cab^2 - 2\,Bb^3 + 2\left(Ba^3 + 2\,Ca^2b - Bab^2\right)dx - \left(Ca^3 - 2\,Ba^2b - Cab^2 + \left(Ca^2b - 2\,Bab^2 - Cb^3\right)\tan\left(dx+c\right)\right)\log\left(\right.}{2\left(\left(a^4b + 2\,a^2b^3 + b^5\right)d\tan\left(dx+c\right) + \left(a^5 +\right.\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="fricas")
```

```
[Out] 1/2*(2*C*a*b^2 - 2*B*b^3 + 2*(B*a^3 + 2*C*a^2*b - B*a*b^2)*d*x - (C*a^3 - 2
*B*a^2*b - C*a*b^2 + (C*a^2*b - 2*B*a*b^2 - C*b^3)*tan(d*x + c))*log((b^2*t
an(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1)) - 2*(C*a^2*
b - B*a*b^2 - (B*a^2*b + 2*C*a*b^2 - B*b^3)*d*x)*tan(d*x + c))/((a^4*b + 2*
a^2*b^3 + b^5)*d*tan(d*x + c) + (a^5 + 2*a^3*b^2 + a*b^4)*d)
```

giac [B] time = 2.61, size = 234, normalized size = 2.11

$$\frac{\frac{2\left(Ba^2+2\,Cab-Bb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4} + \frac{\left(Ca^2-2\,Bab-Cb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4} - \frac{2\left(Ca^2b-2\,Bab^2-Cb^3\right)\log\left(|b\tan(dx+c)+a|\right)}{a^4b+2\,a^2b^3+b^5} + \frac{2\left(Ca^2b\tan(dx+c)-2\,Bab^2\tan\right.}{\left(a^4+2\,a^2\right.}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="giac")
```

```
[Out] 1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + (C*a^2
 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2*(C*
a^2*b - 2*B*a*b^2 - C*b^3)*log(abs(b*tan(d*x + c) + a))/(a^4*b + 2*a^2*b^3
+ b^5) + 2*(C*a^2*b*tan(d*x + c) - 2*B*a*b^2*tan(d*x + c) - C*b^3*tan(d*x +
 c) + 2*C*a^3 - 3*B*a^2*b - B*b^3)/((a^4 + 2*a^2*b^2 + b^4)*(b*tan(d*x + c)
 + a)))/d
```

maple [B] time = 0.76, size = 301, normalized size = 2.71

$$-\frac{Bb}{d\left(a^2+b^2\right)\left(a+b\tan\left(dx+c\right)\right)} + \frac{aC}{d\left(a^2+b^2\right)\left(a+b\tan\left(dx+c\right)\right)} + \frac{2ab\ln\left(a+b\tan\left(dx+c\right)\right)B}{d\left(a^2+b^2\right)^2} - \frac{a^2\ln\left(a+b\tan\left(\right.\right.}{d\left(a^2+b^2\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x)
```

```
[Out] -1/d/(a^2+b^2)/(a+b*tan(d*x+c))*B*b+1/d/(a^2+b^2)/(a+b*tan(d*x+c))*a*C+2/d*
a/(a^2+b^2)^2*b*ln(a+b*tan(d*x+c))*B-1/d*a^2/(a^2+b^2)^2*ln(a+b*tan(d*x+c))
*C+1/d/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*b^2*C-1/d/(a^2+b^2)^2*ln(1+tan(d*x+c)
^2)*B*a*b+1/2/d/(a^2+b^2)^2*ln(1+tan(d*x+c)^2)*a^2*C-1/2/d/(a^2+b^2)^2*ln(1
+tan(d*x+c)^2)*b^2*C+1/d/(a^2+b^2)^2*B*arctan(tan(d*x+c))*a^2-1/d/(a^2+b^2)
^2*B*arctan(tan(d*x+c))*b^2+2/d/(a^2+b^2)^2*C*arctan(tan(d*x+c))*a*b
```

maxima [A] time = 0.57, size = 177, normalized size = 1.59

$$\frac{\frac{2\left(Ba^2+2\,Cab-Bb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(Ca^2-2\,Bab-Cb^2\right)\log(b\tan(dx+c)+a)}{a^4+2\,a^2b^2+b^4}+\frac{\left(Ca^2-2\,Bab-Cb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}+\frac{2\,(Ca-Bb)}{a^3+ab^2+\left(a^2b+b^3\right)\tan(dx+c)}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x, al
gorithm="maxima")
```

```
[Out] 1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - 2*(C*a
^2 - 2*B*a*b - C*b^2)*log(b*tan(d*x + c) + a)/(a^4 + 2*a^2*b^2 + b^4) + (C*
a^2 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) + 2*
(C*a - B*b)/(a^3 + a*b^2 + (a^2*b + b^3)*tan(d*x + c)))/d
```

mupad [B] time = 9.09, size = 153, normalized size = 1.38

$$\frac{\ln\left(a+b\tan\left(c+d\,x\right)\right)\,\left(-C\,a^2+2\,B\,a\,b+C\,b^2\right)}{d\,{\left(a^2+b^2\right)}^2}-\frac{B\,b-C\,a}{d\,\left(a^2+b^2\right)\,\left(a+b\tan\left(c+d\,x\right)\right)}-\frac{\ln\left(\tan\left(c+d\,x\right)+1{}\mathrm{i}\right)\,\left(C+B\right.}{2\,d\,\left(-a^2+a\,b\,2{}\mathrm{i}+b^2\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
^2,x)
```

```
[Out] (log(a + b*tan(c + d*x))*(C*b^2 - C*a^2 + 2*B*a*b))/(d*(a^2 + b^2)^2) - (B*
b - C*a)/(d*(a^2 + b^2)*(a + b*tan(c + d*x))) - (log(tan(c + d*x) + 1i)*(B*
1i + C))/(2*d*(a*b*2i - a^2 + b^2)) - (log(tan(c + d*x) - 1i)*(B + C*1i))/(
2*d*(2*a*b - a^2*1i + b^2*1i))
```

sympy [A] time = 4.99, size = 2895, normalized size = 26.08

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)*cot(c)/tan(c)**2, Eq(a, 0) & Eq(b
, 0) & Eq(d, 0)), (B*d*x*tan(c + d*x)**2/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b
**2*d*tan(c + d*x) + 4*b**2*d) - 2*I*B*d*x*tan(c + d*x)/(-4*b**2*d*tan(c +
d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) - B*d*x/(-4*b**2*d*tan(c + d*
x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + B*tan(c + d*x)/(-4*b**2*d*tan
(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) - 2*I*B/(-4*b**2*d*tan(c
 + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) - I*C*d*x*tan(c + d*x)**2/
(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) - 2*C*d*x*
tan(c + d*x)/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*
d) + I*C*d*x/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) + 4*b**2*
d) - I*C*tan(c + d*x)/(-4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x)
+ 4*b**2*d), Eq(a, -I*b)), (B*d*x*tan(c + d*x)**2/(-4*b**2*d*tan(c + d*x)**
2 - 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + 2*I*B*d*x*tan(c + d*x)/(-4*b**2*d
*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) - B*d*x/(-4*b**2*d*t
an(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + B*tan(c + d*x)/(-4*b
**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + 2*I*B/(-4*b**
2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) + I*C*d*x*tan(c +
 d*x)**2/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) + 4*b**2*d) -
 2*C*d*x*tan(c + d*x)/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x)
+ 4*b**2*d) - I*C*d*x/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x)
+ 4*b**2*d) + I*C*tan(c + d*x)/(-4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(
c + d*x) + 4*b**2*d), Eq(a, I*b)), (x*(B*tan(c) + C*tan(c)**2)*cot(c)/(a +
b*tan(c))**2, Eq(d, 0)), ((B*x + C*log(tan(c + d*x)**2 + 1)/(2*d))/a**2, Eq
(b, 0)), (2*B*a**3*d*x/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d
+ 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 2*B*a*
*2*b*d*x*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d +
 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 4*B*a**
2*b*log(a/b + tan(c + d*x))/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b*
*2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2
*B*a**2*b*log(tan(c + d*x)**2 + 1)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*
a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*
x)) - 2*B*a**2*b/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a*
*2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*B*a*b**2*d
*x/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*tan(
c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 4*B*a*b**2*log(a/b + tan(c
 + d*x))*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d +
 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*B*a*b
**2*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*
x) + 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan
(c + d*x)) - 2*B*b**3*d*x*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x)
+ 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c
+ d*x)) - 2*B*b**3/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*
a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*C*a**3*l
```

```
og(a/b + tan(c + d*x))/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d
+ 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + C*a**3
*log(tan(c + d*x)**2 + 1)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2
*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 2*C
*a**3/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*t
an(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 4*C*a**2*b*d*x/(2*a**5*
d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) +
2*a*b**4*d + 2*b**5*d*tan(c + d*x)) - 2*C*a**2*b*log(a/b + tan(c + d*x))*ta
n(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**
3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + C*a**2*b*log(tan(c
 + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b
**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) +
4*C*a*b**2*d*x*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b*
*2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 2
*C*a*b**2*log(a/b + tan(c + d*x))/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a
**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x
)) - C*a*b**2*log(tan(c + d*x)**2 + 1)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x)
+ 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c
+ d*x)) + 2*C*a*b**2/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) + 4*a**3*b**2*d +
4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c + d*x)) + 2*C*b**3
*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**5*d + 2*a**4*b*d*tan(c + d*x) +
 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d + 2*b**5*d*tan(c +
 d*x)) - C*b**3*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**5*d + 2*a**4*b*
d*tan(c + d*x) + 4*a**3*b**2*d + 4*a**2*b**3*d*tan(c + d*x) + 2*a*b**4*d +
2*b**5*d*tan(c + d*x)), True))
```

3.36 $\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$

Optimal. Leaf size=137

$$\frac{b(bB-aC)}{ad\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{x\left(a^2(-C)+2abB+b^2C\right)}{\left(a^2+b^2\right)^2}+\frac{B\log(\sin(c+dx))}{a^2d}-\frac{b\left(-2a^3C+3a^2bB+b^3B\right)\log(a}{a^2d\left(a^2+\right.}$$

```
[Out] -(2*B*a*b-C*a^2+C*b^2)*x/(a^2+b^2)^2+B*ln(sin(d*x+c))/a^2/d-b*(3*B*a^2*b+B*
b^3-2*C*a^3)*ln(a*cos(d*x+c)+b*sin(d*x+c))/a^2/(a^2+b^2)^2/d+b*(B*b-C*a)/a/
(a^2+b^2)/d/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.40, antiderivative size = 137, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3632, 3609, 3651, 3530, 3475}

$$\frac{b(bB-aC)}{ad\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{b\left(3a^2bB-2a^3C+b^3B\right)\log(a\cos(c+dx)+b\sin(c+dx))}{a^2d\left(a^2+b^2\right)^2}-\frac{x\left(a^2(-C)+2abB+\right.}{\left(a^2+b^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^2,x]
```

```
[Out] -(((2*a*b*B - a^2*C + b^2*C)*x)/(a^2 + b^2)^2) + (B*Log[Sin[c + d*x]])/(a^2
*d) - (b*(3*a^2*b*B + b^3*B - 2*a^3*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]]
)/(a^2*(a^2 + b^2)^2*d) + (b*(b*B - a*C))/(a*(a^2 + b^2)*d*(a + b*Tan[c + d
*x]))
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
```

```
mp[(b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1)
)/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx=\int \frac{\cot(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^2}\,dx$$

$$=\frac{b(bB-aC)}{a\left(a^2+b^2\right)d(a+b\tan(c+dx))}+\frac{\int \frac{\cot(c+dx)\left(\left(a^2+b^2\right)B-a(bB-\ldots\right.}{a+\ldots}}{a\ldots}$$

$$=-\frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2}+\frac{b(bB-aC)}{a\left(a^2+b^2\right)d(a+b\tan(c+dx))}\ldots$$

$$=-\frac{\left(2abB-a^2C+b^2C\right)x}{\left(a^2+b^2\right)^2}+\frac{B\log(\sin(c+dx))}{a^2d}-\frac{b\left(3a^2bB\ldots\right.}{\ldots}$$

Mathematica [C] time = 2.41, size = 159, normalized size = 1.16

$$-\frac{\frac{2b(aC-bB)}{a\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{2B\log(\tan(c+dx))}{a^2}+\frac{2b\left(-2a^3C+3a^2bB+b^3B\right)\log(a+b\tan(c+dx))}{a^2\left(a^2+b^2\right)^2}+\frac{(B+iC)\log(-\tan(c+dx)+i)}{(a+ib)^2}+\frac{(B-iC)\log(\tan(c\ldots}{(a-ib)^2}}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^2,x]
```

```
[Out] -1/2*(((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^2 - (2*B*Log[Tan[c + d*x]
])/a^2 + ((B - I*C)*Log[I + Tan[c + d*x]])/(a - I*b)^2 + (2*b*(3*a^2*b*B +
b^3*B - 2*a^3*C)*Log[a + b*Tan[c + d*x]])/(a^2*(a^2 + b^2)^2) + (2*b*(-(b*B
) + a*C))/(a*(a^2 + b^2)*(a + b*Tan[c + d*x])))/d
```

fricas [B] time = 0.92, size = 323, normalized size = 2.36

$$-\frac{2\,Ca^2b^3-2\,Bab^4-2\left(Ca^5-2\,Ba^4b-Ca^3b^2\right)dx-\left(Ba^5+2\,Ba^3b^2+Bab^4+\left(Ba^4b+2\,Ba^2b^3+Bb^5\right)\tan\left(dx+c\right)\ldots\right.}{\ldots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="fricas")
```

```
[Out] -1/2*(2*C*a^2*b^3 - 2*B*a*b^4 - 2*(C*a^5 - 2*B*a^4*b - C*a^3*b^2)*d*x - (B*
a^5 + 2*B*a^3*b^2 + B*a*b^4 + (B*a^4*b + 2*B*a^2*b^3 + B*b^5)*tan(d*x + c))
```

```
*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1)) - (2*C*a^4*b - 3*B*a^3*b^2 - B*a*
b^4 + (2*C*a^3*b^2 - 3*B*a^2*b^3 - B*b^5)*tan(d*x + c))*log((b^2*tan(d*x +
c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1)) - 2*(C*a^3*b^2 - B*a
^2*b^3 + (C*a^4*b - 2*B*a^3*b^2 - C*a^2*b^3)*d*x)*tan(d*x + c))/((a^6*b + 2
*a^4*b^3 + a^2*b^5)*d*tan(d*x + c) + (a^7 + 2*a^5*b^2 + a^3*b^4)*d)
```

giac [B] time = 5.29, size = 279, normalized size = 2.04

$$\frac{\frac{2\left(Ca^2-2\,Bab-Cb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}-\frac{\left(Ba^2+2\,Cab-Bb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}+\frac{2\left(2\,Ca^3b^2-3\,Ba^2b^3-Bb^5\right)\log(|b\tan(dx+c)+a|)}{a^6b+2\,a^4b^3+a^2b^5}+\frac{2\,B\log(|\tan(dx+c)|)}{a^2}-}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="giac")
```

```
[Out] 1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) - (B*a^2
 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) + 2*(2*
C*a^3*b^2 - 3*B*a^2*b^3 - B*b^5)*log(abs(b*tan(d*x + c) + a))/(a^6*b + 2*a^
4*b^3 + a^2*b^5) + 2*B*log(abs(tan(d*x + c)))/a^2 - 2*(2*C*a^3*b^2*tan(d*x
+ c) - 3*B*a^2*b^3*tan(d*x + c) - B*b^5*tan(d*x + c) + 3*C*a^4*b - 4*B*a^3*
b^2 + C*a^2*b^3 - 2*B*a*b^4)/((a^6 + 2*a^4*b^2 + a^2*b^4)*(b*tan(d*x + c) +
 a)))/d
```

maple [B] time = 0.84, size = 325, normalized size = 2.37

$$-\frac{3\ln\left(a+b\tan\left(dx+c\right)\right)b^2B}{d\left(a^2+b^2\right)^2}-\frac{b^4\ln\left(a+b\tan\left(dx+c\right)\right)B}{d\,a^2\left(a^2+b^2\right)^2}+\frac{2\ln\left(a+b\tan\left(dx+c\right)\right)Cab}{d\left(a^2+b^2\right)^2}+\frac{b^2B}{da\left(a^2+b^2\right)\left(a+b\tan\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x)
```

```
[Out] -3/d/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*b^2*B-1/d*b^4/a^2/(a^2+b^2)^2*ln(a+b*ta
n(d*x+c))*B+2/d/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*C*a*b+1/d*b^2/a/(a^2+b^2)/(a
+b*tan(d*x+c))*B-1/d*b/(a^2+b^2)/(a+b*tan(d*x+c))*C+1/d*B/a^2*ln(tan(d*x+c)
)-1/2/d/(a^2+b^2)^2*ln(1+tan(d*x+c)^2)*a^2*B+1/2/d/(a^2+b^2)^2*ln(1+tan(d*x
+c)^2)*b^2*B-1/d/(a^2+b^2)^2*ln(1+tan(d*x+c)^2)*C*a*b-2/d/(a^2+b^2)^2*B*arc
tan(tan(d*x+c))*a*b+1/d/(a^2+b^2)^2*C*arctan(tan(d*x+c))*a^2-1/d/(a^2+b^2)^
2*C*arctan(tan(d*x+c))*b^2
```

maxima [A] time = 1.33, size = 208, normalized size = 1.52

$$\frac{\frac{2\left(Ca^2-2\,Bab-Cb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}+\frac{2\left(2\,Ca^3b-3\,Ba^2b^2-Bb^4\right)\log(b\tan(dx+c)+a)}{a^6+2\,a^4b^2+a^2b^4}-\frac{\left(Ba^2+2\,Cab-Bb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(Cab-Bb^2\right)}{a^4+a^2b^2+\left(a^3b+ab^3\right)\tan(d}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="maxima")
```

```
[Out] 1/2*(2*(C*a^2 - 2*B*a*b - C*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + 2*(2*C
*a^3*b - 3*B*a^2*b^2 - B*b^4)*log(b*tan(d*x + c) + a)/(a^6 + 2*a^4*b^2 + a^
2*b^4) - (B*a^2 + 2*C*a*b - B*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2
 + b^4) - 2*(C*a*b - B*b^2)/(a^4 + a^2*b^2 + (a^3*b + a*b^3)*tan(d*x + c))
+ 2*B*log(tan(d*x + c))/a^2)/d
```

mupad [B] time = 10.69, size = 180, normalized size = 1.31

$$\frac{B\,\ln\left(\tan\left(c+d\,x\right)\right)}{a^2\,d}-\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(B+C\,1\mathrm{i}\right)}{2\,d\,\left(a^2+a\,b\,2\mathrm{i}-b^2\right)}-\frac{\ln\left(\tan\left(c+d\,x\right)+1\mathrm{i}\right)\,\left(C+B\,1\mathrm{i}\right)}{2\,d\,\left(a^2\,1\mathrm{i}+2\,a\,b-b^2\,1\mathrm{i}\right)}+\frac{B\,b^2-C\,a\,b}{a\,d\,\left(a^2+b^2\right)\,\left(a+b\,\tan\left(\right.\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^2,x)
```

```
[Out] (B*log(tan(c + d*x)))/(a^2*d) - (log(tan(c + d*x) - 1i)*(B + C*1i))/(2*d*(a
*b*2i + a^2 - b^2)) - (log(tan(c + d*x) + 1i)*(B*1i + C))/(2*d*(2*a*b + a^2
*1i - b^2*1i)) + (B*b^2 - C*a*b)/(a*d*(a^2 + b^2)*(a + b*tan(c + d*x))) - (
b*log(a + b*tan(c + d*x))*(B*b^3 - 2*C*a^3 + 3*B*a^2*b))/(a^2*d*(a^2 + b^2)
^2)
```

sympy [A] time = 9.51, size = 4461, normalized size = 32.56

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,
x)
```

```
[Out] Piecewise((zoo*x*(B*tan(c) + C*tan(c)**2)*cot(c)**2/tan(c)**2, Eq(a, 0) & E
q(b, 0) & Eq(d, 0)), ((-B*log(tan(c + d*x)**2 + 1)/(2*d) + B*log(tan(c + d*
x))/d + C*x)/a**2, Eq(b, 0)), ((B*log(tan(c + d*x)**2 + 1)/(2*d) - B*log(ta
n(c + d*x))/d - B/(2*d*tan(c + d*x)**2) - C*x - C/(d*tan(c + d*x)))/b**2, E
q(a, 0)), (3*I*B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d
*tan(c + d*x) - 4*b**2*d) + 6*B*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2
- 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 3*I*B*d*x/(4*b**2*d*tan(c + d*x)**2
 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*B*log(tan(c + d*x)**2 + 1)*tan(c
 + d*x)**2/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d)
- 4*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8
```

```
*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 2*B*log(tan(c + d*x)**2 + 1)/(4*b**2*d
*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 4*B*log(tan(c + d*
x))*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4
*b**2*d) + 8*I*B*log(tan(c + d*x))*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 -
 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 4*B*log(tan(c + d*x))/(4*b**2*d*tan(
c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 3*I*B*tan(c + d*x)/(4*b
**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 4*B/(4*b**2*d
*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - C*d*x*tan(c + d*x)
**2/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*I*C
*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b
**2*d) + C*d*x/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b**2
*d) - C*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) -
4*b**2*d) + 2*I*C/(4*b**2*d*tan(c + d*x)**2 - 8*I*b**2*d*tan(c + d*x) - 4*b
**2*d), Eq(a, -I*b)), (-3*I*B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2
 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 6*B*d*x*tan(c + d*x)/(4*b**2*d*tan
(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 3*I*B*d*x/(4*b**2*d*ta
n(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 2*B*log(tan(c + d*x)*
*2 + 1)*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x)
 - 4*b**2*d) + 4*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(4*b**2*d*tan(c
+ d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 2*B*log(tan(c + d*x)**2 +
 1)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 4*B*l
og(tan(c + d*x))*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan
(c + d*x) - 4*b**2*d) - 8*I*B*log(tan(c + d*x))*tan(c + d*x)/(4*b**2*d*tan(
c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) + 4*B*log(tan(c + d*x))/(
4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - 3*I*B*tan(
c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) +
4*B/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b**2*d) - C*d*x
*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c + d*x) - 4*b*
*2*d) - 2*I*C*d*x*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c
 + d*x) - 4*b**2*d) + C*d*x/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c +
d*x) - 4*b**2*d) - C*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*ta
n(c + d*x) - 4*b**2*d) - 2*I*C/(4*b**2*d*tan(c + d*x)**2 + 8*I*b**2*d*tan(c
 + d*x) - 4*b**2*d), Eq(a, I*b)), (x*(B*tan(c) + C*tan(c)**2)*cot(c)**2/(a
+ b*tan(c))**2, Eq(d, 0)), (-B*a**5*log(tan(c + d*x)**2 + 1)/(2*a**7*d + 2*
a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3
*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 2*B*a**5*log(tan(c + d*x))/(2*a**7*
d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) +
2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 4*B*a**4*b*d*x/(2*a**7*d + 2*
a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3
*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - B*a**4*b*log(tan(c + d*x)**2 + 1)*t
an(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b*
*3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 2*B*a**4*
b*log(tan(c + d*x))*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a*
*5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(
c + d*x)) - 4*B*a**3*b**2*d*x*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d
```

```
*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b
**5*d*tan(c + d*x)) - 6*B*a**3*b**2*log(a/b + tan(c + d*x))/(2*a**7*d + 2*a
**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*
b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + B*a**3*b**2*log(tan(c + d*x)**2 + 1)
/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c
+ d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 4*B*a**3*b**2*log(ta
n(c + d*x))/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b*
*3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 2*B*a**3*
b**2/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*ta
n(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 6*B*a**2*b**3*lo
g(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*
a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*ta
n(c + d*x)) + B*a**2*b**3*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**7*d +
 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a
**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 4*B*a**2*b**3*log(tan(c + d*x))*
tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b
**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 2*B*a*b*
*4*log(a/b + tan(c + d*x))/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**
2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*
x)) + 2*B*a*b**4*log(tan(c + d*x))/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*
a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*ta
n(c + d*x)) + 2*B*a*b**4/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*
d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)
) - 2*B*b**5*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*ta
n(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2
*a**2*b**5*d*tan(c + d*x)) + 2*B*b**5*log(tan(c + d*x))*tan(c + d*x)/(2*a**
7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x)
+ 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 2*C*a**5*d*x/(2*a**7*d + 2*
a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3
*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 2*C*a**4*b*d*x*tan(c + d*x)/(2*a**7
*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) +
 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 4*C*a**4*b*log(a/b + tan(c +
 d*x))/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*
tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 2*C*a**4*b*log
(tan(c + d*x)**2 + 1)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d +
 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) -
 2*C*a**4*b/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b*
*3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 2*C*a**3*
b**2*d*x/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*
d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) + 4*C*a**3*b**
2*log(a/b + tan(c + d*x))*tan(c + d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x)
+ 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*
d*tan(c + d*x)) - 2*C*a**3*b**2*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a*
*7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d*x)
 + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 2*C*a**2*b**3*d*x*tan(c +
```

```
d*x)/(2*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*ta
n(c + d*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)) - 2*C*a**2*b**3/(2
*a**7*d + 2*a**6*b*d*tan(c + d*x) + 4*a**5*b**2*d + 4*a**4*b**3*d*tan(c + d
*x) + 2*a**3*b**4*d + 2*a**2*b**5*d*tan(c + d*x)), True))
```

3.37 $\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx$

Optimal. Leaf size=192

$$-\frac{(2bB-aC)\log(\sin(c+dx))}{a^3d}-\frac{b\left(a^2B-abC+2b^2B\right)}{a^2d\left(a^2+b^2\right)(a+b\tan(c+dx))}-\frac{x\left(a^2B+2abC-b^2B\right)}{\left(a^2+b^2\right)^2}+\frac{b^2\left(-3a^3C+4a^2bB-a\right.}{}$$

```
[Out] -(B*a^2-B*b^2+2*C*a*b)*x/(a^2+b^2)^2-(2*B*b-C*a)*ln(sin(d*x+c))/a^3/d+b^2*(
4*B*a^2*b+2*B*b^3-3*C*a^3-C*a*b^2)*ln(a*cos(d*x+c)+b*sin(d*x+c))/a^3/(a^2+b
^2)^2/d-b*(B*a^2+2*B*b^2-C*a*b)/a^2/(a^2+b^2)/d/(a+b*tan(d*x+c))-B*cot(d*x+
c)/a/d/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.61, antiderivative size = 192, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3632, 3609, 3649, 3651, 3530, 3475}

$$-\frac{b\left(a^2B-abC+2b^2B\right)}{a^2d\left(a^2+b^2\right)(a+b\tan(c+dx))}+\frac{b^2\left(4a^2bB-3a^3C-ab^2C+2b^3B\right)\log(a\cos(c+dx)+b\sin(c+dx))}{a^3d\left(a^2+b^2\right)^2}-\frac{x\left(a^2B\right.}{\left(\right.}$$

Antiderivative was successfully verified.

```
[In] Int[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^2,x]
```

```
[Out] -(((a^2*B - b^2*B + 2*a*b*C)*x)/(a^2 + b^2)^2) - ((2*b*B - a*C)*Log[Sin[c +
 d*x]])/(a^3*d) + (b^2*(4*a^2*b*B + 2*b^3*B - 3*a^3*C - a*b^2*C)*Log[a*Cos[
c + d*x] + b*Sin[c + d*x]])/(a^3*(a^2 + b^2)^2*d) - (b*(a^2*B + 2*b^2*B - a
*b*C))/(a^2*(a^2 + b^2)*d*(a + b*Tan[c + d*x])) - (B*Cot[c + d*x])/(a*d*(a
+ b*Tan[c + d*x]))
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1)
)/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^2}\,dx = \int \frac{\cot^2(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^2}\,dx$$

$$= -\frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))} - \frac{\int \frac{\cot(c+dx)\left(2bB-aC+aB\tan(c+dx)+2bB\right.}{(a+b\tan(c+dx))^2}}{a}$$

$$= -\frac{b\left(a^2B+2b^2B-abC\right)}{a^2\left(a^2+b^2\right)d(a+b\tan(c+dx))} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))}$$

$$= -\frac{\left(a^2B-b^2B+2abC\right)x}{\left(a^2+b^2\right)^2} - \frac{b\left(a^2B+2b^2B-abC\right)}{a^2\left(a^2+b^2\right)d(a+b\tan(c+dx)}$$

$$= -\frac{\left(a^2B-b^2B+2abC\right)x}{\left(a^2+b^2\right)^2} - \frac{(2bB-aC)\log(\sin(c+dx))}{a^3d} + $$

Mathematica [C] time = 3.57, size = 193, normalized size = 1.01

$$\frac{\frac{2(aC-2bB)\log(\tan(c+dx))}{a^3} + \frac{2b^2(aC-bB)}{a^2\left(a^2+b^2\right)(a+b\tan(c+dx))} - \frac{2B\cot(c+dx)}{a^2} - \frac{2b^2\left(3a^3C-4a^2bB+ab^2C-2b^3B\right)\log(a+b\tan(c+dx))}{a^3\left(a^2+b^2\right)^2} + \frac{i(B+iC)\log(-\mathrm{ta}}{(a+ib)}}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^2,x]
```

```
[Out] ((-2*B*Cot[c + d*x])/a^2 + (I*(B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^2
+ (2*(-2*b*B + a*C)*Log[Tan[c + d*x]])/a^3 - ((I*B + C)*Log[I + Tan[c + d*x
]])/(a - I*b)^2 - (2*b^2*(-4*a^2*b*B - 2*b^3*B + 3*a^3*C + a*b^2*C)*Log[a +
 b*Tan[c + d*x]])/(a^3*(a^2 + b^2)^2) + (2*b^2*(-(b*B) + a*C))/(a^2*(a^2 +
b^2)*(a + b*Tan[c + d*x])))/(2*d)
```

fricas [B] time = 0.91, size = 465, normalized size = 2.42

$$-\frac{2\,Ba^6+4\,Ba^4b^2+2\,Ba^2b^4+2\left(Ca^3b^3-Ba^2b^4+\left(Ba^5b+2\,Ca^4b^2-Ba^3b^3\right)dx\right)\tan\left(dx+c\right)^2-\left(\left(Ca^5b-2\,Ba^4b^2\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="fricas")
```

```
[Out] -1/2*(2*B*a^6 + 4*B*a^4*b^2 + 2*B*a^2*b^4 + 2*(C*a^3*b^3 - B*a^2*b^4 + (B*a
^5*b + 2*C*a^4*b^2 - B*a^3*b^3)*d*x)*tan(d*x + c)^2 - ((C*a^5*b - 2*B*a^4*b
^2 + 2*C*a^3*b^3 - 4*B*a^2*b^4 + C*a*b^5 - 2*B*b^6)*tan(d*x + c)^2 + (C*a^6
 - 2*B*a^5*b + 2*C*a^4*b^2 - 4*B*a^3*b^3 + C*a^2*b^4 - 2*B*a*b^5)*tan(d*x +
 c))*log(tan(d*x + c)^2/(tan(d*x + c)^2 + 1)) + ((3*C*a^3*b^3 - 4*B*a^2*b^4
 + C*a*b^5 - 2*B*b^6)*tan(d*x + c)^2 + (3*C*a^4*b^2 - 4*B*a^3*b^3 + C*a^2*b
^4 - 2*B*a*b^5)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c)
+ a^2)/(tan(d*x + c)^2 + 1)) + 2*(B*a^5*b + 2*B*a^3*b^3 - C*a^2*b^4 + 2*B*a
*b^5 + (B*a^6 + 2*C*a^5*b - B*a^4*b^2)*d*x)*tan(d*x + c))/((a^7*b + 2*a^5*b
^3 + a^3*b^5)*d*tan(d*x + c)^2 + (a^8 + 2*a^6*b^2 + a^4*b^4)*d*tan(d*x + c)
)
```

giac [A] time = 6.71, size = 362, normalized size = 1.89

$$-\frac{\frac{2\left(Ba^2+2\,Cab-Bb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}+\frac{\left(Ca^2-2\,Bab-Cb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}+\frac{2\left(3\,Ca^3b^3-4\,Ba^2b^4+Cab^5-2\,Bb^6\right)\log(|b\tan(dx+c)+a|)}{a^7b+2\,a^5b^3+a^3b^5}+\frac{Ca^4b\tan(dx+}{}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="giac")
```

```
[Out] -1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + (C*a^
2 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) + 2*(3
*C*a^3*b^3 - 4*B*a^2*b^4 + C*a*b^5 - 2*B*b^6)*log(abs(b*tan(d*x + c) + a))/
(a^7*b + 2*a^5*b^3 + a^3*b^5) + (C*a^4*b*tan(d*x + c)^2 - 2*B*a^3*b^2*tan(d
*x + c)^2 - C*a^2*b^3*tan(d*x + c)^2 + C*a^5*tan(d*x + c) - 3*C*a^3*b^2*tan
(d*x + c) + 6*B*a^2*b^3*tan(d*x + c) - 2*C*a*b^4*tan(d*x + c) + 4*B*b^5*tan
(d*x + c) + 2*B*a^5 + 4*B*a^3*b^2 + 2*B*a*b^4)/((a^6 + 2*a^4*b^2 + a^2*b^4)
*(b*tan(d*x + c)^2 + a*tan(d*x + c))) - 2*(C*a - 2*B*b)*log(abs(tan(d*x + c
)))/a^3)/d
```

maple [B] time = 0.86, size = 399, normalized size = 2.08

$$\frac{4b^3\ln\left(a+b\tan\left(dx+c\right)\right)B}{da\left(a^2+b^2\right)^2}+\frac{2b^5\ln\left(a+b\tan\left(dx+c\right)\right)B}{d\,a^3\left(a^2+b^2\right)^2}-\frac{3\ln\left(a+b\tan\left(dx+c\right)\right)b^2C}{d\left(a^2+b^2\right)^2}-\frac{b^4\ln\left(a+b\tan\left(dx+c\right)\right.}{d\,a^2\left(a^2+b^2\right)^2}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x)
```

```
[Out] 4/d*b^3/a/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*B+2/d*b^5/a^3/(a^2+b^2)^2*ln(a+b*t
an(d*x+c))*B-3/d/(a^2+b^2)^2*ln(a+b*tan(d*x+c))*b^2*C-1/d*b^4/a^2/(a^2+b^2)
^2*ln(a+b*tan(d*x+c))*C-1/d*b^3/a^2/(a^2+b^2)/(a+b*tan(d*x+c))*B+1/d*b^2/a/
(a^2+b^2)/(a+b*tan(d*x+c))*C-1/d*B/a^2/tan(d*x+c)-2/d/a^3*ln(tan(d*x+c))*B*
b+1/d/a^2*ln(tan(d*x+c))*C+1/d/(a^2+b^2)^2*ln(1+tan(d*x+c)^2)*B*a*b-1/2/d/(
a^2+b^2)^2*ln(1+tan(d*x+c)^2)*a^2*C+1/2/d/(a^2+b^2)^2*ln(1+tan(d*x+c)^2)*b^
2*C-1/d/(a^2+b^2)^2*B*arctan(tan(d*x+c))*a^2+1/d/(a^2+b^2)^2*B*arctan(tan(d
*x+c))*b^2-2/d/(a^2+b^2)^2*C*arctan(tan(d*x+c))*a*b
```

maxima [A] time = 0.82, size = 262, normalized size = 1.36

$$-\frac{\frac{2\left(Ba^2+2\,Cab-Bb^2\right)(dx+c)}{a^4+2\,a^2b^2+b^4}+\frac{2\left(3\,Ca^3b^2-4\,Ba^2b^3+Cab^4-2\,Bb^5\right)\log(b\tan(dx+c)+a)}{a^7+2\,a^5b^2+a^3b^4}+\frac{\left(Ca^2-2\,Bab-Cb^2\right)\log\left(\tan(dx+c)^2+1\right)}{a^4+2\,a^2b^2+b^4}+\frac{2\left(Ba^3+Bab^2+(B}{\left(a^4b+a^2b^3\right)\tan(}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^2,x,
algorithm="maxima")
```

```
[Out] -1/2*(2*(B*a^2 + 2*C*a*b - B*b^2)*(d*x + c)/(a^4 + 2*a^2*b^2 + b^4) + 2*(3*
C*a^3*b^2 - 4*B*a^2*b^3 + C*a*b^4 - 2*B*b^5)*log(b*tan(d*x + c) + a)/(a^7 +
 2*a^5*b^2 + a^3*b^4) + (C*a^2 - 2*B*a*b - C*b^2)*log(tan(d*x + c)^2 + 1)/(
a^4 + 2*a^2*b^2 + b^4) + 2*(B*a^3 + B*a*b^2 + (B*a^2*b - C*a*b^2 + 2*B*b^3)
*tan(d*x + c))/((a^4*b + a^2*b^3)*tan(d*x + c)^2 + (a^5 + a^3*b^2)*tan(d*x
+ c)) - 2*(C*a - 2*B*b)*log(tan(d*x + c))/a^3)/d
```

mupad [B] time = 12.15, size = 230, normalized size = 1.20

$$\frac{b^2\,\ln\left(a+b\tan\left(c+d\,x\right)\right)\,\left(-3\,C\,a^3+4\,B\,a^2\,b-C\,a\,b^2+2\,B\,b^3\right)}{a^3\,d\,\left(a^2+b^2\right)^2}-\frac{\ln\left(\tan\left(c+d\,x\right)\right)\,\left(2\,B\,b-C\,a\right)}{a^3\,d}+\frac{\ln\left(\tan\left(c+d\,x\right)\right.}{2\,d\,\left(-a^2+\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^2,x)
```

```
[Out] (log(tan(c + d*x) + 1i)*(B*1i + C))/(2*d*(a*b*2i - a^2 + b^2)) - (log(tan(c
 + d*x))*(2*B*b - C*a))/(a^3*d) - (B/a + (tan(c + d*x)*(2*B*b^3 + B*a^2*b -
 C*a*b^2))/(a^2*(a^2 + b^2)))/(d*(a*tan(c + d*x) + b*tan(c + d*x)^2)) + (lo
g(tan(c + d*x) - 1i)*(B + C*1i))/(2*d*(2*a*b - a^2*1i + b^2*1i)) + (b^2*log
(a + b*tan(c + d*x))*(2*B*b^3 - 3*C*a^3 + 4*B*a^2*b - C*a*b^2))/(a^3*d*(a^2
 + b^2)^2)
```

sympy [A] time = 15.71, size = 8097, normalized size = 42.17

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**3*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**2,
x)
```

```
[Out] Piecewise((nan, Eq(a, 0) & Eq(b, 0) & Eq(c, 0) & Eq(d, 0)), ((B*x + B/(d*ta
n(c + d*x)) - B/(3*d*tan(c + d*x)**3) + C*log(tan(c + d*x)**2 + 1)/(2*d) -
C*log(tan(c + d*x))/d - C/(2*d*tan(c + d*x)**2))/b**2, Eq(a, 0)), (9*B*d*x*
tan(c + d*x)**3/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*
b**2*d*tan(c + d*x)) - 18*I*B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3
 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 9*B*d*x*tan(c + d*
x)/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c
+ d*x)) - 4*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(4*b**2*d*tan(c +
d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 8*B*log(tan
(c + d*x)**2 + 1)*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*ta
n(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) + 4*I*B*log(tan(c + d*x)**2 + 1)*tan
(c + d*x)/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d
*tan(c + d*x)) + 8*I*B*log(tan(c + d*x))*tan(c + d*x)**3/(4*b**2*d*tan(c +
d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) + 16*B*log(ta
n(c + d*x))*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c +
d*x)**2 - 4*b**2*d*tan(c + d*x)) - 8*I*B*log(tan(c + d*x))*tan(c + d*x)/(4*
b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)
) + 9*B*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)
**2 - 4*b**2*d*tan(c + d*x)) - 14*I*B*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**
3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 4*B/(4*b**2*d*tan
(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) + 3*I*C*
d*x*tan(c + d*x)**3/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2
- 4*b**2*d*tan(c + d*x)) + 6*C*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**
3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 3*I*C*d*x*tan(c +
 d*x)/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan
(c + d*x)) + 2*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(4*b**2*d*tan(c +
 d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 4*I*C*log(
tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d
*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 2*C*log(tan(c + d*x)**2 + 1)*ta
n(c + d*x)/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*
d*tan(c + d*x)) - 4*C*log(tan(c + d*x))*tan(c + d*x)**3/(4*b**2*d*tan(c + d
*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) + 8*I*C*log(ta
n(c + d*x))*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c +
d*x)**2 - 4*b**2*d*tan(c + d*x)) + 4*C*log(tan(c + d*x))*tan(c + d*x)/(4*b*
*2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x))
+ 3*I*C*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 - 8*I*b**2*d*tan(c + d*x)
**2 - 4*b**2*d*tan(c + d*x)) + 4*C*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**3 -
 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)), Eq(a, -I*b)), (9*B*d*
x*tan(c + d*x)**3/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 -
4*b**2*d*tan(c + d*x)) + 18*I*B*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)*
```

```
*3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 9*B*d*x*tan(c +
d*x)/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(
c + d*x)) + 4*I*B*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(4*b**2*d*tan(c
+ d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 8*B*log(t
an(c + d*x)**2 + 1)*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*
tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 4*I*B*log(tan(c + d*x)**2 + 1)*t
an(c + d*x)/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2
*d*tan(c + d*x)) - 8*I*B*log(tan(c + d*x))*tan(c + d*x)**3/(4*b**2*d*tan(c
+ d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) + 16*B*log(
tan(c + d*x))*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c
+ d*x)**2 - 4*b**2*d*tan(c + d*x)) + 8*I*B*log(tan(c + d*x))*tan(c + d*x)/(
4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*
x)) + 9*B*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c + d*
x)**2 - 4*b**2*d*tan(c + d*x)) + 14*I*B*tan(c + d*x)/(4*b**2*d*tan(c + d*x)
**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 4*B/(4*b**2*d*t
an(c + d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 3*I*
C*d*x*tan(c + d*x)**3/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c + d*x)**
2 - 4*b**2*d*tan(c + d*x)) + 6*C*d*x*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)
**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) + 3*I*C*d*x*tan(c
 + d*x)/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*t
an(c + d*x)) + 2*C*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**3/(4*b**2*d*tan(c
 + d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) + 4*I*C*lo
g(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2
*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 2*C*log(tan(c + d*x)**2 + 1)*
tan(c + d*x)/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**
2*d*tan(c + d*x)) - 4*C*log(tan(c + d*x))*tan(c + d*x)**3/(4*b**2*d*tan(c +
 d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)) - 8*I*C*log(
tan(c + d*x))*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c
+ d*x)**2 - 4*b**2*d*tan(c + d*x)) + 4*C*log(tan(c + d*x))*tan(c + d*x)/(4*
b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)
) - 3*I*C*tan(c + d*x)**2/(4*b**2*d*tan(c + d*x)**3 + 8*I*b**2*d*tan(c + d*
x)**2 - 4*b**2*d*tan(c + d*x)) + 4*C*tan(c + d*x)/(4*b**2*d*tan(c + d*x)**3
 + 8*I*b**2*d*tan(c + d*x)**2 - 4*b**2*d*tan(c + d*x)), Eq(a, I*b)), (nan,
Eq(c, -d*x)), (x*(B*tan(c) + C*tan(c)**2)*cot(c)**3/(a + b*tan(c))**2, Eq(d
, 0)), ((-B*x - B/(d*tan(c + d*x)) - C*log(tan(c + d*x)**2 + 1)/(2*d) + C*l
og(tan(c + d*x))/d)/a**2, Eq(b, 0)), (-2*B*a**6*d*x*tan(c + d*x)/(2*a**8*d*
tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*
a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*ta
n(c + d*x)**2) - 2*B*a**6/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)*
*2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b*
*4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 2*B*a**5*b*d*x*tan(c +
 d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*
d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x)
 + 2*a**3*b**5*d*tan(c + d*x)**2) + 2*B*a**5*b*log(tan(c + d*x)**2 + 1)*tan
(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2
```

```
*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x
) + 2*a**3*b**5*d*tan(c + d*x)**2) - 4*B*a**5*b*log(tan(c + d*x))*tan(c + d
*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan
(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*
a**3*b**5*d*tan(c + d*x)**2) - 2*B*a**5*b*tan(c + d*x)/(2*a**8*d*tan(c + d*
x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*
d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)
**2) + 2*B*a**4*b**2*d*x*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*t
an(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2
 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 2*B*a**4*b
**2*log(tan(c + d*x)**2 + 1)*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**
7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c +
d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 4*B
*a**4*b**2*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**
7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c +
d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 4*B
*a**4*b**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**
2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*
x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 2*B*a**3*b**3*d*x*tan(c + d*x)**2/(2*
a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*
x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b*
*5*d*tan(c + d*x)**2) + 8*B*a**3*b**3*log(a/b + tan(c + d*x))*tan(c + d*x)/
(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c +
 d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3
*b**5*d*tan(c + d*x)**2) - 8*B*a**3*b**3*log(tan(c + d*x))*tan(c + d*x)/(2*
a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*
x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b*
*5*d*tan(c + d*x)**2) - 6*B*a**3*b**3*tan(c + d*x)/(2*a**8*d*tan(c + d*x) +
 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*ta
n(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2)
 + 8*B*a**2*b**4*log(a/b + tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan(c +
d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**
3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*
x)**2) - 8*B*a**2*b**4*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan(c +
d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**
3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*
x)**2) - 2*B*a**2*b**4/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2
+ 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*
d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 4*B*a*b**5*log(a/b + tan(
c + d*x))*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2
+ 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*
d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 4*B*a*b**5*log(tan(c + d*
x))*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a*
*6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(
c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 4*B*a*b**5*tan(c + d*x)/(2*a**8
```

```
*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) +
 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d
*tan(c + d*x)**2) + 4*B*b**6*log(a/b + tan(c + d*x))*tan(c + d*x)**2/(2*a**
8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x)
+ 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*
d*tan(c + d*x)**2) - 4*B*b**6*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*t
an(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a
**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan
(c + d*x)**2) - C*a**6*log(tan(c + d*x)**2 + 1)*tan(c + d*x)/(2*a**8*d*tan(
c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5
*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c
+ d*x)**2) + 2*C*a**6*log(tan(c + d*x))*tan(c + d*x)/(2*a**8*d*tan(c + d*x)
 + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*
tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**
2) - 4*C*a**5*b*d*x*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c
+ d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*
a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - C*a**5*b*log(ta
n(c + d*x)**2 + 1)*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(
c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 +
2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 2*C*a**5*b*lo
g(tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c +
 d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a
**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 4*C*a**4*b**2*d*
x*tan(c + d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a
**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan
(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) - 6*C*a**4*b**2*log(a/b + tan(c
+ d*x))*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 +
4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*
tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + C*a**4*b**2*log(tan(c + d*x
)**2 + 1)*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2
+ 4*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*
d*tan(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 4*C*a**4*b**2*log(tan(c +
 d*x))*tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4
*a**6*b**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*t
an(c + d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 2*C*a**4*b**2*tan(c + d*x)/(
2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c +
d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*
b**5*d*tan(c + d*x)**2) - 6*C*a**3*b**3*log(a/b + tan(c + d*x))*tan(c + d*x
)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*ta
n(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2
*a**3*b**5*d*tan(c + d*x)**2) + C*a**3*b**3*log(tan(c + d*x)**2 + 1)*tan(c
+ d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2
*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x
) + 2*a**3*b**5*d*tan(c + d*x)**2) + 4*C*a**3*b**3*log(tan(c + d*x))*tan(c
+ d*x)**2/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2
```

```
*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x
) + 2*a**3*b**5*d*tan(c + d*x)**2) - 2*C*a**2*b**4*log(a/b + tan(c + d*x))*
tan(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b
**2*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c +
d*x) + 2*a**3*b**5*d*tan(c + d*x)**2) + 2*C*a**2*b**4*log(tan(c + d*x))*tan
(c + d*x)/(2*a**8*d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2
*d*tan(c + d*x) + 4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x
) + 2*a**3*b**5*d*tan(c + d*x)**2) + 2*C*a**2*b**4*tan(c + d*x)/(2*a**8*d*t
an(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a
**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan
(c + d*x)**2) - 2*C*a*b**5*log(a/b + tan(c + d*x))*tan(c + d*x)**2/(2*a**8*
d*tan(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) +
4*a**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*
tan(c + d*x)**2) + 2*C*a*b**5*log(tan(c + d*x))*tan(c + d*x)**2/(2*a**8*d*t
an(c + d*x) + 2*a**7*b*d*tan(c + d*x)**2 + 4*a**6*b**2*d*tan(c + d*x) + 4*a
**5*b**3*d*tan(c + d*x)**2 + 2*a**4*b**4*d*tan(c + d*x) + 2*a**3*b**5*d*tan
(c + d*x)**2), True))
```

3.38 $\int \frac{\tan^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$

Optimal. Leaf size=331

$$\frac{a(bB-aC)\tan^3(c+dx)}{2bd\left(a^2+b^2\right)(a+b\tan(c+dx))^2}+\frac{a\left(-3a^3C+a^2bB-7ab^2C+5b^3B\right)\tan^2(c+dx)}{2b^2d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}+\frac{\left(a^3(-C)+3a^2bB+3ab^2C-\right.}{d\left(a^2+b\right.}$$

```
[Out] (B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*x/(a^2+b^2)^3+(3*B*a^2*b-B*b^3-C*a^3+3*C*
a*b^2)*ln(cos(d*x+c))/(a^2+b^2)^3/d+a^2*(B*a^4*b+3*B*a^2*b^3+6*B*b^5-3*C*a^
5-9*C*a^3*b^2-10*C*a*b^4)*ln(a+b*tan(d*x+c))/b^4/(a^2+b^2)^3/d-(B*a^3*b+3*B
*a*b^3-3*C*a^4-6*C*a^2*b^2-C*b^4)*tan(d*x+c)/b^3/(a^2+b^2)^2/d+1/2*a*(B*b-C
*a)*tan(d*x+c)^3/b/(a^2+b^2)/d/(a+b*tan(d*x+c))^2+1/2*a*(B*a^2*b+5*B*b^3-3*
C*a^3-7*C*a*b^2)*tan(d*x+c)^2/b^2/(a^2+b^2)^2/d/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.86, antiderivative size = 331, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 8, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.200, Rules used = {3632, 3605, 3645, 3647, 3626, 3617, 31, 3475}

$$\frac{a(bB-aC)\tan^3(c+dx)}{2bd\left(a^2+b^2\right)(a+b\tan(c+dx))^2}+\frac{a\left(a^2bB-3a^3C-7ab^2C+5b^3B\right)\tan^2(c+dx)}{2b^2d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}-\frac{\left(-6a^2b^2C+a^3bB-3a^4C+3\right.}{b^3d\left(a^2+\right.}$$

Antiderivative was successfully verified.

```
[In] Int[(Tan[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^3,x]
```

```
[Out] ((a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*x)/(a^2 + b^2)^3 + ((3*a^2*b*B - b
^3*B - a^3*C + 3*a*b^2*C)*Log[Cos[c + d*x]])/((a^2 + b^2)^3*d) + (a^2*(a^4*
b*B + 3*a^2*b^3*B + 6*b^5*B - 3*a^5*C - 9*a^3*b^2*C - 10*a*b^4*C)*Log[a + b
*Tan[c + d*x]])/(b^4*(a^2 + b^2)^3*d) - ((a^3*b*B + 3*a*b^3*B - 3*a^4*C - 6
*a^2*b^2*C - b^4*C)*Tan[c + d*x])/(b^3*(a^2 + b^2)^2*d) + (a*(b*B - a*C)*Ta
n[c + d*x]^3)/(2*b*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) + (a*(a^2*b*B + 5*
b^3*B - 3*a^3*C - 7*a*b^2*C)*Tan[c + d*x]^2)/(2*b^2*(a^2 + b^2)^2*d*(a + b*
Tan[c + d*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3605

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[((b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x
])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
```

```
(f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{\tan^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \int \frac{\tan^4(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$

$$= \frac{a(bB-aC)\tan^3(c+dx)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{\int \frac{\tan^2(c+dx)\left(-3a(bB-aC\right.}{}}{}$$

$$= \frac{a(bB-aC)\tan^3(c+dx)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{a\left(a^2bB+5b^3B-3a\right.}{2b^2\left(a^2+b^2\right)^2}$$

$$= -\frac{\left(a^3bB+3ab^3B-3a^4C-6a^2b^2C-b^4C\right)\tan(c+dx)}{b^3\left(a^2+b^2\right)^2 d} +$$

$$= \frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3} - \frac{\left(a^3bB+3ab^3B-3a^4\right.}{b^3}$$

$$= \frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3} + \frac{\left(3a^2bB-b^3B-a^3C\right.}{\left(a^2\right.}$$

$$= \frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3} + \frac{\left(3a^2bB-b^3B-a^3C\right.}{\left(a^2\right.}$$

Mathematica [C]   time = 6.86, size = 1146, normalized size = 3.46

$$\frac{(aC-bB)\sec^2(c+dx)(a\cos(c+dx)+b\sin(c+dx))(B+C\tan(c+dx))a^4}{2(a-ib)^2(a+ib)^2b^2d(B\cos(c+dx)+C\sin(c+dx))(a+b\tan(c+dx))^3} + \frac{\sec^2(c+dx)(a\cos(c+dx)+b\sin(}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Tan[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^3,x]
```

```
[Out] (a^4*(-(b*B) + a*C)*Sec[c + d*x]^2*(a*Cos[c + d*x] + b*Sin[c + d*x])*(B + C
*Tan[c + d*x]))/(2*(a - I*b)^2*(a + I*b)^2*b^2*d*(B*Cos[c + d*x] + C*Sin[c
+ d*x])*(a + b*Tan[c + d*x])^3) + ((a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*
(c + d*x)*Sec[c + d*x]^2*(a*Cos[c + d*x] + b*Sin[c + d*x])^3*(B + C*Tan[c +
 d*x]))/((a - I*b)^3*(a + I*b)^3*d*(B*Cos[c + d*x] + C*Sin[c + d*x])*(a + b
*Tan[c + d*x])^3) + ((I*a^11*b^4*B + a^10*b^5*B + (5*I)*a^9*b^6*B + 5*a^8*b
```

```
^7*B + (13*I)*a^7*b^8*B + 13*a^6*b^9*B + (15*I)*a^5*b^10*B + 15*a^4*b^11*B
+ (6*I)*a^3*b^12*B + 6*a^2*b^13*B - (3*I)*a^12*b^3*C - 3*a^11*b^4*C - (15*I
)*a^10*b^5*C - 15*a^9*b^6*C - (31*I)*a^8*b^7*C - 31*a^7*b^8*C - (29*I)*a^6*
b^9*C - 29*a^5*b^10*C - (10*I)*a^4*b^11*C - 10*a^3*b^12*C)*(c + d*x)*Sec[c
+ d*x]^2*(a*Cos[c + d*x] + b*Sin[c + d*x])^3*(B + C*Tan[c + d*x]))/((a - I*
b)^6*(a + I*b)^5*b^7*d*(B*Cos[c + d*x] + C*Sin[c + d*x])*(a + b*Tan[c + d*x
])^3) - (I*(a^6*b*B + 3*a^4*b^3*B + 6*a^2*b^5*B - 3*a^7*C - 9*a^5*b^2*C - 1
0*a^3*b^4*C)*ArcTan[Tan[c + d*x]]*Sec[c + d*x]^2*(a*Cos[c + d*x] + b*Sin[c
+ d*x])^3*(B + C*Tan[c + d*x]))/(b^4*(a^2 + b^2)^3*d*(B*Cos[c + d*x] + C*Si
n[c + d*x])*(a + b*Tan[c + d*x])^3) + ((-(b*B) + 3*a*C)*Log[Cos[c + d*x]]*S
ec[c + d*x]^2*(a*Cos[c + d*x] + b*Sin[c + d*x])^3*(B + C*Tan[c + d*x]))/(b^
4*d*(B*Cos[c + d*x] + C*Sin[c + d*x])*(a + b*Tan[c + d*x])^3) + ((a^6*b*B +
 3*a^4*b^3*B + 6*a^2*b^5*B - 3*a^7*C - 9*a^5*b^2*C - 10*a^3*b^4*C)*Log[(a*C
os[c + d*x] + b*Sin[c + d*x])^2]*Sec[c + d*x]^2*(a*Cos[c + d*x] + b*Sin[c +
 d*x])^3*(B + C*Tan[c + d*x]))/(2*b^4*(a^2 + b^2)^3*d*(B*Cos[c + d*x] + C*S
in[c + d*x])*(a + b*Tan[c + d*x])^3) + (Sec[c + d*x]^2*(a*Cos[c + d*x] + b*
Sin[c + d*x])^2*(-(a^4*b*B*Sin[c + d*x]) - 4*a^2*b^3*B*Sin[c + d*x] + 2*a^5
*C*Sin[c + d*x] + 5*a^3*b^2*C*Sin[c + d*x])*(B + C*Tan[c + d*x]))/((a - I*b
)^2*(a + I*b)^2*b^3*d*(B*Cos[c + d*x] + C*Sin[c + d*x])*(a + b*Tan[c + d*x]
)^3) + (C*Sec[c + d*x]^2*(a*Cos[c + d*x] + b*Sin[c + d*x])^3*Tan[c + d*x]*(
B + C*Tan[c + d*x]))/(b^3*d*(B*Cos[c + d*x] + C*Sin[c + d*x])*(a + b*Tan[c
+ d*x])^3)
```

fricas [B] time = 0.93, size = 890, normalized size = 2.69

$$-\frac{3\,Ca^7b^2 - Ba^6b^3 + 9\,Ca^5b^4 - 7\,Ba^4b^5 - 2\left(Ca^6b^3 + 3\,Ca^4b^5 + 3\,Ca^2b^7 + Cb^9\right)\tan\left(dx + c\right)^3 - 2\left(Ba^5b^4 + 3\,Ca^4b\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="fricas")
```

```
[Out] -1/2*(3*C*a^7*b^2 - B*a^6*b^3 + 9*C*a^5*b^4 - 7*B*a^4*b^5 - 2*(C*a^6*b^3 +
3*C*a^4*b^5 + 3*C*a^2*b^7 + C*b^9)*tan(d*x + c)^3 - 2*(B*a^5*b^4 + 3*C*a^4*
b^5 - 3*B*a^3*b^6 - C*a^2*b^7)*d*x - (9*C*a^7*b^2 - 3*B*a^6*b^3 + 23*C*a^5*
b^4 - 9*B*a^4*b^5 + 12*C*a^3*b^6 + 4*C*a*b^8 + 2*(B*a^3*b^6 + 3*C*a^2*b^7 -
 3*B*a*b^8 - C*b^9)*d*x)*tan(d*x + c)^2 + (3*C*a^9 - B*a^8*b + 9*C*a^7*b^2
- 3*B*a^6*b^3 + 10*C*a^5*b^4 - 6*B*a^4*b^5 + (3*C*a^7*b^2 - B*a^6*b^3 + 9*C
*a^5*b^4 - 3*B*a^4*b^5 + 10*C*a^3*b^6 - 6*B*a^2*b^7)*tan(d*x + c)^2 + 2*(3*
C*a^8*b - B*a^7*b^2 + 9*C*a^6*b^3 - 3*B*a^5*b^4 + 10*C*a^4*b^5 - 6*B*a^3*b^
6)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d
*x + c)^2 + 1)) - (3*C*a^9 - B*a^8*b + 9*C*a^7*b^2 - 3*B*a^6*b^3 + 9*C*a^5*
b^4 - 3*B*a^4*b^5 + 3*C*a^3*b^6 - B*a^2*b^7 + (3*C*a^7*b^2 - B*a^6*b^3 + 9*
C*a^5*b^4 - 3*B*a^4*b^5 + 9*C*a^3*b^6 - 3*B*a^2*b^7 + 3*C*a*b^8 - B*b^9)*ta
```

```
n(d*x + c)^2 + 2*(3*C*a^8*b - B*a^7*b^2 + 9*C*a^6*b^3 - 3*B*a^5*b^4 + 9*C*a
^4*b^5 - 3*B*a^3*b^6 + 3*C*a^2*b^7 - B*a*b^8)*tan(d*x + c))*log(1/(tan(d*x
+ c)^2 + 1)) - 2*(3*C*a^8*b - B*a^7*b^2 + 6*C*a^6*b^3 - 3*B*a^5*b^4 - 2*C*a
^4*b^5 + 4*B*a^3*b^6 + C*a^2*b^7 + 2*(B*a^4*b^5 + 3*C*a^3*b^6 - 3*B*a^2*b^7
 - C*a*b^8)*d*x)*tan(d*x + c))/((a^6*b^6 + 3*a^4*b^8 + 3*a^2*b^10 + b^12)*d
*tan(d*x + c)^2 + 2*(a^7*b^5 + 3*a^5*b^7 + 3*a^3*b^9 + a*b^11)*d*tan(d*x +
c) + (a^8*b^4 + 3*a^6*b^6 + 3*a^4*b^8 + a^2*b^10)*d)
```

giac [A] time = 4.34, size = 505, normalized size = 1.53

$$\frac{\frac{2\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{2\left(3\,Ca^7-Ba^6b+9\,Ca^5b^2-3\,Ba^4b^3+10\,Ca^3b^4-6\,Ba^2b^5\right)\log}{a^6b^4+3\,a^4b^6+3\,a^2b^8+b^{10}}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="giac")
```

```
[Out] 1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x + c
)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(3*C*a^7 - B*a^6*b + 9*C*a
^5*b^2 - 3*B*a^4*b^3 + 10*C*a^3*b^4 - 6*B*a^2*b^5)*log(abs(b*tan(d*x + c) +
 a))/(a^6*b^4 + 3*a^4*b^6 + 3*a^2*b^8 + b^10) + 2*C*tan(d*x + c)/b^3 + (9*C
*a^7*b^2*tan(d*x + c)^2 - 3*B*a^6*b^3*tan(d*x + c)^2 + 27*C*a^5*b^4*tan(d*x
 + c)^2 - 9*B*a^4*b^5*tan(d*x + c)^2 + 30*C*a^3*b^6*tan(d*x + c)^2 - 18*B*a
^2*b^7*tan(d*x + c)^2 + 12*C*a^8*b*tan(d*x + c) - 2*B*a^7*b^2*tan(d*x + c)
+ 38*C*a^6*b^3*tan(d*x + c) - 6*B*a^5*b^4*tan(d*x + c) + 50*C*a^4*b^5*tan(d
*x + c) - 28*B*a^3*b^6*tan(d*x + c) + 4*C*a^9 + 13*C*a^7*b^2 + B*a^6*b^3 +
21*C*a^5*b^4 - 11*B*a^4*b^5)/((a^6*b^4 + 3*a^4*b^6 + 3*a^2*b^8 + b^10)*(b*t
an(d*x + c) + a)^2))/d
```

maple [A] time = 0.28, size = 619, normalized size = 1.87

$$\frac{C\tan(dx+c)}{d\,b^3}+\frac{a^6\ln(a+b\tan(dx+c))\,B}{d\,b^3\left(a^2+b^2\right)^3}+\frac{3a^4\ln(a+b\tan(dx+c))\,B}{db\left(a^2+b^2\right)^3}+\frac{6b\,a^2\ln(a+b\tan(dx+c))\,B}{d\left(a^2+b^2\right)^3}-\frac{3a^7\ln}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x)
```

```
[Out] 1/d*C/b^3*tan(d*x+c)+1/d/b^3*a^6/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*B+3/d/b*a^4
/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*B+6/d*b*a^2/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*
B-3/d/b^4*a^7/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*C-9/d/b^2*a^5/(a^2+b^2)^3*ln(a
+b*tan(d*x+c))*C-10/d*a^3/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*C-1/2/d/b^3*a^4/(a
```

```
^2+b^2)/(a+b*tan(d*x+c))^2*B+1/2/d/b^4*a^5/(a^2+b^2)/(a+b*tan(d*x+c))^2*C+2
/d/b^3*a^5/(a^2+b^2)^2/(a+b*tan(d*x+c))*B+4/d/b*a^3/(a^2+b^2)^2/(a+b*tan(d*
x+c))*B-3/d/b^4*a^6/(a^2+b^2)^2/(a+b*tan(d*x+c))*C-5/d/b^2*a^4/(a^2+b^2)^2/
(a+b*tan(d*x+c))*C-3/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*a^2*b*B+1/2/d/(a^2+
b^2)^3*ln(1+tan(d*x+c)^2)*b^3*B+1/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*C*a^3-
3/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*C*a*b^2+1/d/(a^2+b^2)^3*B*arctan(tan(d
*x+c))*a^3-3/d/(a^2+b^2)^3*B*arctan(tan(d*x+c))*a*b^2+3/d/(a^2+b^2)^3*C*arc
tan(tan(d*x+c))*a^2*b-1/d/(a^2+b^2)^3*C*arctan(tan(d*x+c))*b^3
```

maxima [A] time = 1.04, size = 389, normalized size = 1.18

$$\frac{\frac{2\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{2\left(3\,Ca^7-Ba^6b+9\,Ca^5b^2-3\,Ba^4b^3+10\,Ca^3b^4-6\,Ba^2b^5\right)\log(b\tan(dx+c)+a)}{a^6b^4+3\,a^4b^6+3\,a^2b^8+b^{10}}+\frac{\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)\log(\cdots}{a^6+3\,a^4b^2+3\,a^2b^4+\cdots}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="maxima")
```

```
[Out] 1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) - 2*(3*C*a^7 - B*a^6*b + 9*C*a^5*b^2 - 3*B*a^4*b^3 + 10*C
*a^3*b^4 - 6*B*a^2*b^5)*log(b*tan(d*x + c) + a)/(a^6*b^4 + 3*a^4*b^6 + 3*a^
2*b^8 + b^10) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x + c)^2
+ 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - (5*C*a^7 - 3*B*a^6*b + 9*C*a^5*b
^2 - 7*B*a^4*b^3 + 2*(3*C*a^6*b - 2*B*a^5*b^2 + 5*C*a^4*b^3 - 4*B*a^3*b^4)*
tan(d*x + c))/(a^6*b^4 + 2*a^4*b^6 + a^2*b^8 + (a^4*b^6 + 2*a^2*b^8 + b^10)
*tan(d*x + c)^2 + 2*(a^5*b^5 + 2*a^3*b^7 + a*b^9)*tan(d*x + c)) + 2*C*tan(d
*x + c)/b^3)/d
```

mupad [B] time = 10.43, size = 335, normalized size = 1.01

$$\frac{C\tan(c+d\,x)}{b^3\,d}+\frac{\ln(\tan(c+d\,x)-\mathrm{i})\,(-C+B\,1\mathrm{i})}{2\,d\,\left(-a^3-a^2\,b\,3\mathrm{i}+3\,a\,b^2+b^3\,1\mathrm{i}\right)}+\frac{\ln(\tan(c+d\,x)+1\mathrm{i})\,(B-C\,1\mathrm{i})}{2\,d\,\left(-a^3\,1\mathrm{i}-3\,a^2\,b+a\,b^2\,3\mathrm{i}+b^3\right)}-\frac{\frac{5\,C\,a^7-3\,B\,a^6\,b+9\,C\,a^5\,b^2-7\,B\,a^4\cdots}{2\,b\,\left(a^4+2\,a^2\,b^2+b^4\right)}\cdots}{d\,\left(a^2\,b^3+2\,a\,b^4\cdots\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((tan(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^3,x)
```

```
[Out] (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i
)) - ((5*C*a^7 - 7*B*a^4*b^3 + 9*C*a^5*b^2 - 3*B*a^6*b)/(2*b*(a^4 + b^4 + 2
*a^2*b^2)) + (tan(c + d*x)*(3*C*a^6 - 4*B*a^3*b^3 + 5*C*a^4*b^2 - 2*B*a^5*b
))/(a^4 + b^4 + 2*a^2*b^2))/(d*(a^2*b^3 + b^5*tan(c + d*x)^2 + 2*a*b^4*tan(
c + d*x))) + (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a*b^2*3i - 3*a^2*b -
```

```
 a^3*1i + b^3)) + (C*tan(c + d*x))/(b^3*d) + (a^2*log(a + b*tan(c + d*x))*(
6*B*b^5 - 3*C*a^5 + 3*B*a^2*b^3 - 9*C*a^3*b^2 + B*a^4*b - 10*C*a*b^4))/(b^4
*d*(a^2 + b^2)^3)
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)**3*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,
x)
```

```
[Out] Exception raised: AttributeError
```

3.39 $\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$

Optimal. Leaf size=250

$$\frac{a(bB-aC)\tan^2(c+dx)}{2bd\left(a^2+b^2\right)(a+b\tan(c+dx))^2}-\frac{a^2\left(a^3(-C)-3ab^2C+2b^3B\right)}{b^3d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}+\frac{\left(a^3B+3a^2bC-3ab^2B-b^3C\right)\log(\cos(c+$$

```
[Out] -(3*B*a^2*b-B*b^3-C*a^3+3*C*a*b^2)*x/(a^2+b^2)^3+(B*a^3-3*B*a*b^2+3*C*a^2*b
-C*b^3)*ln(cos(d*x+c))/(a^2+b^2)^3/d+a*(B*a^2*b^3-3*B*b^5+C*a^5+3*C*a^3*b^2
+6*C*a*b^4)*ln(a+b*tan(d*x+c))/b^3/(a^2+b^2)^3/d+1/2*a*(B*b-C*a)*tan(d*x+c)
^2/b/(a^2+b^2)/d/(a+b*tan(d*x+c))^2-a^2*(2*B*b^3-C*a^3-3*C*a*b^2)/b^3/(a^2+
b^2)^2/d/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.58, antiderivative size = 250, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.175, Rules used = {3632, 3605, 3635, 3626, 3617, 31, 3475}

$$\frac{a(bB-aC)\tan^2(c+dx)}{2bd\left(a^2+b^2\right)(a+b\tan(c+dx))^2}-\frac{a^2\left(a^3(-C)-3ab^2C+2b^3B\right)}{b^3d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}+\frac{a\left(a^2b^3B+3a^3b^2C+a^5C+6ab^4C-3b^5B\right.}{b^3d\left(a^2+b^2\right)^3}$$

Antiderivative was successfully verified.

```
[In] Int[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^3,x]
```

```
[Out] -(((3*a^2*b*B - b^3*B - a^3*C + 3*a*b^2*C)*x)/(a^2 + b^2)^3) + ((a^3*B - 3*
a*b^2*B + 3*a^2*b*C - b^3*C)*Log[Cos[c + d*x]])/((a^2 + b^2)^3*d) + (a*(a^2
*b^3*B - 3*b^5*B + a^5*C + 3*a^3*b^2*C + 6*a*b^4*C)*Log[a + b*Tan[c + d*x]]
)/(b^3*(a^2 + b^2)^3*d) + (a*(b*B - a*C)*Tan[c + d*x]^2)/(2*b*(a^2 + b^2)*d
*(a + b*Tan[c + d*x])^2) - (a^2*(2*b^3*B - a^3*C - 3*a*b^2*C))/(b^3*(a^2 +
b^2)^2*d*(a + b*Tan[c + d*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3605

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[((b*c - a*d)*(B*c - A*d)*(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x
])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dist[1/(d*(n + 1)*(c^2 + d^2)),
 Int[(a + b*Tan[e + f*x])^(m - 2)*(c + d*Tan[e + f*x])^(n + 1)*Simp[a*A*d*(
b*d*(m - 1) - a*c*(n + 1)) + (b*B*c - (A*b + a*B)*d)*(b*c*(m - 1) + a*d*(n
+ 1)) - d*((a*A - b*B)*(b*c - a*d) + (A*b + a*B)*(a*c + b*d))*(n + 1)*Tan[e
 + f*x] - b*(d*(A*b*c + a*B*c - a*A*d)*(m + n) - b*B*(c^2*(m - 1) - d^2*(n
+ 1)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 1] &&
 LtQ[n, -1] && (IntegerQ[m] || IntegersQ[2*m, 2*n])
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
```

```
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \frac{\tan^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \int \frac{\tan^3(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$

$$= \frac{a(bB-aC)\tan^2(c+dx)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{\int \frac{\tan(c+dx)\left(-2a(bB-aC)+\right.}{}}{}$$

$$= \frac{a(bB-aC)\tan^2(c+dx)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} - \frac{a^2\left(2b^3B-a^3C-\right.}{b^3\left(a^2+b^2\right)^2 d(a+b\,\mathrm{t}}$$

$$= -\frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)x}{\left(a^2+b^2\right)^3} + \frac{a(bB-aC)\tan^2(}{2b\left(a^2+b^2\right)d(a+b\,\mathrm{t}}$$

$$= -\frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)x}{\left(a^2+b^2\right)^3} + \frac{\left(a^3B-3ab^2B+3a^2b\right.}{\left(a^2\right.}$$

$$= -\frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)x}{\left(a^2+b^2\right)^3} + \frac{\left(a^3B-3ab^2B+3a^2b\right.}{\left(a^2\right.}$$

Mathematica [C] time = 4.92, size = 462, normalized size = 1.85

$$\frac{\sec^2(c+dx)(B+C\tan(c+dx))(a\cos(c+dx)+b\sin(c+dx))\left(-2C\left(a^2+b^2\right)^3\log(\cos(c+dx))(a\cos(c+dx)+\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Tan[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^3,x]
```

```
[Out] (Sec[c + d*x]^2*(a*Cos[c + d*x] + b*Sin[c + d*x])*(a^3*b^2*(a^2 + b^2)*(b*B
 - a*C) - 2*a*b*(a^2 + b^2)*(-3*b^3*B + a^3*C + 4*a*b^2*C)*Sin[c + d*x]*(a*
Cos[c + d*x] + b*Sin[c + d*x]) + 2*b^3*(-3*a^2*b*B + b^3*B + a^3*C - 3*a*b^
2*C)*(c + d*x)*(a*Cos[c + d*x] + b*Sin[c + d*x])^2 + (2*I)*a*(a^2*b^3*B - 3
```

```
*b^5*B + a^5*C + 3*a^3*b^2*C + 6*a*b^4*C)*(c + d*x)*(a*Cos[c + d*x] + b*Sin
[c + d*x])^2 - (2*I)*a*(a^2*b^3*B - 3*b^5*B + a^5*C + 3*a^3*b^2*C + 6*a*b^4
*C)*ArcTan[Tan[c + d*x]]*(a*Cos[c + d*x] + b*Sin[c + d*x])^2 - 2*(a^2 + b^2
)^3*C*Log[Cos[c + d*x]]*(a*Cos[c + d*x] + b*Sin[c + d*x])^2 + a*(a^2*b^3*B
- 3*b^5*B + a^5*C + 3*a^3*b^2*C + 6*a*b^4*C)*Log[(a*Cos[c + d*x] + b*Sin[c
+ d*x])^2]*(a*Cos[c + d*x] + b*Sin[c + d*x])^2)*(B + C*Tan[c + d*x]))/(2*b^
3*(a^2 + b^2)^3*d*(B*Cos[c + d*x] + C*Sin[c + d*x])*(a + b*Tan[c + d*x])^3)
```

fricas [B] time = 1.46, size = 666, normalized size = 2.66

$$\frac{Ca^6b^2 + Ba^5b^3 + 7\,Ca^4b^4 - 5\,Ba^3b^5 + 2\left(Ca^5b^3 - 3\,Ba^4b^4 - 3\,Ca^3b^5 + Ba^2b^6\right)dx - \left(3\,Ca^6b^2 - Ba^5b^3 + 9\,Ca^4b^4\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="fricas")
```

```
[Out] 1/2*(C*a^6*b^2 + B*a^5*b^3 + 7*C*a^4*b^4 - 5*B*a^3*b^5 + 2*(C*a^5*b^3 - 3*B
*a^4*b^4 - 3*C*a^3*b^5 + B*a^2*b^6)*d*x - (3*C*a^6*b^2 - B*a^5*b^3 + 9*C*a^
4*b^4 - 7*B*a^3*b^5 - 2*(C*a^3*b^5 - 3*B*a^2*b^6 - 3*C*a*b^7 + B*b^8)*d*x)*
tan(d*x + c)^2 + (C*a^8 + 3*C*a^6*b^2 + B*a^5*b^3 + 6*C*a^4*b^4 - 3*B*a^3*b
^5 + (C*a^6*b^2 + 3*C*a^4*b^4 + B*a^3*b^5 + 6*C*a^2*b^6 - 3*B*a*b^7)*tan(d*
x + c)^2 + 2*(C*a^7*b + 3*C*a^5*b^3 + B*a^4*b^4 + 6*C*a^3*b^5 - 3*B*a^2*b^6
)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*
x + c)^2 + 1)) - (C*a^8 + 3*C*a^6*b^2 + 3*C*a^4*b^4 + C*a^2*b^6 + (C*a^6*b^
2 + 3*C*a^4*b^4 + 3*C*a^2*b^6 + C*b^8)*tan(d*x + c)^2 + 2*(C*a^7*b + 3*C*a^
5*b^3 + 3*C*a^3*b^5 + C*a*b^7)*tan(d*x + c))*log(1/(tan(d*x + c)^2 + 1)) -
2*(C*a^7*b + 3*C*a^5*b^3 - 3*B*a^4*b^4 - 4*C*a^3*b^5 + 3*B*a^2*b^6 - 2*(C*a
^4*b^4 - 3*B*a^3*b^5 - 3*C*a^2*b^6 + B*a*b^7)*d*x)*tan(d*x + c))/((a^6*b^5
+ 3*a^4*b^7 + 3*a^2*b^9 + b^11)*d*tan(d*x + c)^2 + 2*(a^7*b^4 + 3*a^5*b^6 +
 3*a^3*b^8 + a*b^10)*d*tan(d*x + c) + (a^8*b^3 + 3*a^6*b^5 + 3*a^4*b^7 + a^
2*b^9)*d)
```

giac [A] time = 3.01, size = 458, normalized size = 1.83

$$\frac{\frac{2\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\left(Ca^6+3\,Ca^4b^2+Ba^3b^3+6\,Ca^2b^4-3\,Bab^5\right)\log(|b\tan(dx+}{a^6b^3+3\,a^4b^5+3\,a^2b^7+b^9}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="giac")
```

```
[Out] 1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) - (B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(tan(d*x + c
)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + 2*(C*a^6 + 3*C*a^4*b^2 + B*a
^3*b^3 + 6*C*a^2*b^4 - 3*B*a*b^5)*log(abs(b*tan(d*x + c) + a))/(a^6*b^3 + 3
*a^4*b^5 + 3*a^2*b^7 + b^9) - (3*C*a^6*b*tan(d*x + c)^2 + 9*C*a^4*b^3*tan(d
*x + c)^2 + 3*B*a^3*b^4*tan(d*x + c)^2 + 18*C*a^2*b^5*tan(d*x + c)^2 - 9*B*
a*b^6*tan(d*x + c)^2 + 2*C*a^7*tan(d*x + c) + 2*B*a^6*b*tan(d*x + c) + 6*C*
a^5*b^2*tan(d*x + c) + 14*B*a^4*b^3*tan(d*x + c) + 28*C*a^3*b^4*tan(d*x + c
) - 12*B*a^2*b^5*tan(d*x + c) + B*a^7 - C*a^6*b + 9*B*a^5*b^2 + 11*C*a^4*b^
3 - 4*B*a^3*b^4)/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*(b*tan(d*x + c) +
 a)^2))/d
```

maple [B] time = 0.28, size = 566, normalized size = 2.26

$$\frac{a^3\ln(a+b\tan(dx+c))B}{d\left(a^2+b^2\right)^3}-\frac{3a\,b^2\ln(a+b\tan(dx+c))B}{d\left(a^2+b^2\right)^3}+\frac{a^6\ln(a+b\tan(dx+c))C}{d\left(a^2+b^2\right)^3 b^3}+\frac{3a^4\ln(a+b\tan(dx+c)}{d\left(a^2+b^2\right)^3 b}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x)
```

```
[Out] 1/d*a^3/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*B-3/d*a/(a^2+b^2)^3*b^2*ln(a+b*tan(d
*x+c))*B+1/d*a^6/(a^2+b^2)^3/b^3*ln(a+b*tan(d*x+c))*C+3/d*a^4/(a^2+b^2)^3/b
*ln(a+b*tan(d*x+c))*C+6/d*a^2/(a^2+b^2)^3*b*ln(a+b*tan(d*x+c))*C-1/d*a^4/b^
2/(a^2+b^2)^2/(a+b*tan(d*x+c))*B-3/d*a^2/(a^2+b^2)^2/(a+b*tan(d*x+c))*B+2/d
*a^5/b^3/(a^2+b^2)^2/(a+b*tan(d*x+c))*C+4/d*a^3/b/(a^2+b^2)^2/(a+b*tan(d*x+
c))*C+1/2/d*a^3/b^2/(a^2+b^2)/(a+b*tan(d*x+c))^2*B-1/2/d*a^4/b^3/(a^2+b^2)/
(a+b*tan(d*x+c))^2*C-1/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*a^3*B+3/2/d/(a^2+
b^2)^3*ln(1+tan(d*x+c)^2)*B*a*b^2-3/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*C*a^
2*b+1/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*b^3*C-3/d/(a^2+b^2)^3*B*arctan(tan
(d*x+c))*a^2*b+1/d/(a^2+b^2)^3*B*arctan(tan(d*x+c))*b^3+1/d/(a^2+b^2)^3*C*a
rctan(tan(d*x+c))*a^3-3/d/(a^2+b^2)^3*C*arctan(tan(d*x+c))*a*b^2
```

maxima [A] time = 0.65, size = 366, normalized size = 1.46

$$\frac{\frac{2\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\left(Ca^6+3\,Ca^4b^2+Ba^3b^3+6\,Ca^2b^4-3\,Bab^5\right)\log(b\tan(dx+c)+a)}{a^6b^3+3\,a^4b^5+3\,a^2b^7+b^9}-\frac{\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)\log\left(\tan(dx+c)^2+\right.}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="maxima")
```

```
[Out] 1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) + 2*(C*a^6 + 3*C*a^4*b^2 + B*a^3*b^3 + 6*C*a^2*b^4 - 3*B*
```

```
a*b^5)*log(b*tan(d*x + c) + a)/(a^6*b^3 + 3*a^4*b^5 + 3*a^2*b^7 + b^9) - (B
*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(tan(d*x + c)^2 + 1)/(a^6 + 3*a^4*
b^2 + 3*a^2*b^4 + b^6) + (3*C*a^6 - B*a^5*b + 7*C*a^4*b^2 - 5*B*a^3*b^3 + 2
*(2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 - 3*B*a^2*b^4)*tan(d*x + c))/(a^6*b^3
 + 2*a^4*b^5 + a^2*b^7 + (a^4*b^5 + 2*a^2*b^7 + b^9)*tan(d*x + c)^2 + 2*(a^
5*b^4 + 2*a^3*b^6 + a*b^8)*tan(d*x + c)))/d
```

mupad [B] time = 9.32, size = 307, normalized size = 1.23

$$\frac{\frac{3\,C\,a^6-B\,a^5\,b+7\,C\,a^4\,b^2-5\,B\,a^3\,b^3}{2\,b^3\,\left(a^4+2\,a^2\,b^2+b^4\right)}-\frac{a^2\,\tan(c+d\,x)\,\left(-2\,C\,a^3+B\,a^2\,b-4\,C\,a\,b^2+3\,B\,b^3\right)}{b^2\,\left(a^4+2\,a^2\,b^2+b^4\right)}}{d\,\left(a^2+2\,a\,b\,\tan\left(c+d\,x\right)+b^2\,\tan\left(c+d\,x\right)^2\right)}+\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-a^3\,1\mathrm{i}+3\,a^2\,b+a\,b^2\,3\mathrm{i}-b^3\right)}+\frac{\ln\left(\tan\left(c\right.\right.}{2\,d\,\left(-a^3\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((tan(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^3,x)
```

```
[Out] ((3*C*a^6 - 5*B*a^3*b^3 + 7*C*a^4*b^2 - B*a^5*b)/(2*b^3*(a^4 + b^4 + 2*a^2*
b^2)) - (a^2*tan(c + d*x)*(3*B*b^3 - 2*C*a^3 + B*a^2*b - 4*C*a*b^2))/(b^2*(
a^4 + b^4 + 2*a^2*b^2)))/(d*(a^2 + b^2*tan(c + d*x)^2 + 2*a*b*tan(c + d*x))
) + (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*b^2*3i + 3*a^2*b - a^3*1i -
 b^3)) + (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(3*a*b^2 + a^2*b*3i - a^3
 - b^3*1i)) + (a*log(a + b*tan(c + d*x))*(C*a^5 - 3*B*b^5 + B*a^2*b^3 + 3*C
*a^3*b^2 + 6*C*a*b^4))/(b^3*d*(a^2 + b^2)^3)
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,
x)
```

```
[Out] Exception raised: AttributeError
```

3.40 $\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$

Optimal. Leaf size=189

$$-\frac{a^2(bB-aC)}{2b^2d\left(a^2+b^2\right)(a+b\tan(c+dx))^2}+\frac{a\left(a^3(-C)-3ab^2C+2b^3B\right)}{b^2d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}-\frac{\left(a^3(-C)+3a^2bB+3ab^2C-b^3B\right)\log(a}{d\left(a^2+b^2\right)^3}$$

```
[Out] -(B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*x/(a^2+b^2)^3-(3*B*a^2*b-B*b^3-C*a^3+3*C
*a*b^2)*ln(a*cos(d*x+c)+b*sin(d*x+c))/(a^2+b^2)^3/d-1/2*a^2*(B*b-C*a)/b^2/(
a^2+b^2)/d/(a+b*tan(d*x+c))^2+a*(2*B*b^3-C*a^3-3*C*a*b^2)/b^2/(a^2+b^2)^2/d
/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.43, antiderivative size = 189, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.132, Rules used = {3632, 3604, 3628, 3531, 3530}

$$-\frac{a^2(bB-aC)}{2b^2d\left(a^2+b^2\right)(a+b\tan(c+dx))^2}+\frac{a\left(a^3(-C)-3ab^2C+2b^3B\right)}{b^2d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}-\frac{\left(3a^2bB+a^3(-C)+3ab^2C-b^3B\right)\log(a}{d\left(a^2+b^2\right)^3}$$

Antiderivative was successfully verified.

```
[In] Int[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
^3,x]
```

```
[Out] -(((a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*x)/(a^2 + b^2)^3) - ((3*a^2*b*B
- b^3*B - a^3*C + 3*a*b^2*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/((a^2 +
b^2)^3*d) - (a^2*(b*B - a*C))/(2*b^2*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2)
+ (a*(2*b^3*B - a^3*C - 3*a*b^2*C))/(b^2*(a^2 + b^2)^2*d*(a + b*Tan[c + d*x
]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
```

```
Q[a*c + b*d, 0]
```

Rule 3604

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^2*((A_.) + (B_.)*tan[(e_.) + (f
_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> -Simp
[((B*c - A*d)*(b*c - a*d)^2*(c + d*Tan[e + f*x])^(n + 1))/(f*d^2*(n + 1)*(c
^2 + d^2)), x] + Dist[1/(d*(c^2 + d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*S
imp[B*(b*c - a*d)^2 + A*d*(a^2*c - b^2*c + 2*a*b*d) + d*(B*(a^2*c - b^2*c +
 2*a*b*d) + A*(2*a*b*c - a^2*d + b^2*d))*Tan[e + f*x] + b^2*B*(c^2 + d^2)*T
an[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B}, x] && NeQ[b*c
- a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -1]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \frac{\tan(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx=\int \frac{\tan^2(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$

$$=-\frac{a^2(bB-aC)}{2b^2\left(a^2+b^2\right)d(a+b\tan(c+dx))^2}+\frac{\int \frac{-a(bB-aC)+b(bB-aC)\ldots}{(a+\ldots}}{b\ldots}$$

$$=-\frac{a^2(bB-aC)}{2b^2\left(a^2+b^2\right)d(a+b\tan(c+dx))^2}+\frac{a\left(2b^3B-a^3C\ldots\right.}{b^2\left(a^2+b^2\right)^2d(a+\ldots}$$

$$=-\frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3}-\frac{a^2(bB-aC\ldots}{2b^2\left(a^2+b^2\right)d(a+b\,\mathrm{ta}\ldots}$$

$$=-\frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3}-\frac{\left(3a^2bB-b^3B-a^3C+\ldots\right.}{\ldots}$$

Mathematica [C] time = 5.45, size = 288, normalized size = 1.52

$$\frac{(bB-aC)\left(\frac{b\left(\frac{\left(a^2+b^2\right)\left(5a^2+4ab\tan(c+dx)+b^2\right)}{(a+b\tan(c+dx))^2}+\left(2b^2-6a^2\right)\log(a+b\tan(c+dx))\right)}{\left(a^2+b^2\right)^3}+\frac{i\log(-\tan(c+dx)+i)}{(a+ib)^3}-\frac{\log(\tan(c+dx)+i)}{(b+ia)^3}\right)+C\left(\frac{2b\left(\frac{a^2+b^2}{a+b\tan(c+dx\ldots}\right.}{\ldots}\right.}{2bd}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Tan[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x])^3,x]
```

```
[Out] (-((b*B + a*C)/(b*(a + b*Tan[c + d*x])^2)) - (2*C*Tan[c + d*x])/(a + b*Tan[
c + d*x])^2 + C*((I*Log[I - Tan[c + d*x]])/(a + I*b)^2 - (I*Log[I + Tan[c +
 d*x]])/(a - I*b)^2 + (2*b*(-2*a*Log[a + b*Tan[c + d*x]] + (a^2 + b^2)/(a +
 b*Tan[c + d*x])))/(a^2 + b^2)^2) + (b*B - a*C)*((I*Log[I - Tan[c + d*x]])/
(a + I*b)^3 - Log[I + Tan[c + d*x]]/(I*a + b)^3 + (b*((-6*a^2 + 2*b^2)*Log[
a + b*Tan[c + d*x]] + ((a^2 + b^2)*(5*a^2 + b^2 + 4*a*b*Tan[c + d*x]))/(a +
 b*Tan[c + d*x])^2))/(a^2 + b^2)^3))/(2*b*d)
```

fricas [B] time = 0.59, size = 478, normalized size = 2.53

$$\frac{Ca^5-3\,Ba^4b-5\,Ca^3b^2+3\,Ba^2b^3-2\left(Ba^5+3\,Ca^4b-3\,Ba^3b^2-Ca^2b^3\right)dx+\left(Ca^5+Ba^4b+7\,Ca^3b^2-5\,Ba^2b^3-\ldots\right.}{\ldots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="fricas")
```

```
[Out] 1/2*(C*a^5 - 3*B*a^4*b - 5*C*a^3*b^2 + 3*B*a^2*b^3 - 2*(B*a^5 + 3*C*a^4*b -
 3*B*a^3*b^2 - C*a^2*b^3)*d*x + (C*a^5 + B*a^4*b + 7*C*a^3*b^2 - 5*B*a^2*b^
3 - 2*(B*a^3*b^2 + 3*C*a^2*b^3 - 3*B*a*b^4 - C*b^5)*d*x)*tan(d*x + c)^2 + (
C*a^5 - 3*B*a^4*b - 3*C*a^3*b^2 + B*a^2*b^3 + (C*a^3*b^2 - 3*B*a^2*b^3 - 3*
C*a*b^4 + B*b^5)*tan(d*x + c)^2 + 2*(C*a^4*b - 3*B*a^3*b^2 - 3*C*a^2*b^3 +
B*a*b^4)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/
(tan(d*x + c)^2 + 1)) + 2*(B*a^5 + 3*C*a^4*b - 3*B*a^3*b^2 - 3*C*a^2*b^3 +
2*B*a*b^4 - 2*(B*a^4*b + 3*C*a^3*b^2 - 3*B*a^2*b^3 - C*a*b^4)*d*x)*tan(d*x
+ c))/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*d*tan(d*x + c)^2 + 2*(a^7*b
+ 3*a^5*b^3 + 3*a^3*b^5 + a*b^7)*d*tan(d*x + c) + (a^8 + 3*a^6*b^2 + 3*a^4*
b^4 + a^2*b^6)*d)
```

giac [B] time = 2.27, size = 410, normalized size = 2.17

$$-\frac{\frac{2\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{2\left(Ca^3b-3\,Ba^2b^2-3\,Cab^3+Bb^4\right)\log(|b\tan(dx+c)+a|)}{a^6b+3\,a^4b^3+3\,a^2b^5+b^7}+}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="giac")
```

```
[Out] -1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x +
c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(C*a^3*b - 3*B*a^2*b^2 -
3*C*a*b^3 + B*b^4)*log(abs(b*tan(d*x + c) + a))/(a^6*b + 3*a^4*b^3 + 3*a^2*
b^5 + b^7) + (3*C*a^3*b^4*tan(d*x + c)^2 - 9*B*a^2*b^5*tan(d*x + c)^2 - 9*C
*a*b^6*tan(d*x + c)^2 + 3*B*b^7*tan(d*x + c)^2 + 2*C*a^6*b*tan(d*x + c) + 1
4*C*a^4*b^3*tan(d*x + c) - 22*B*a^3*b^4*tan(d*x + c) - 12*C*a^2*b^5*tan(d*x
 + c) + 2*B*a*b^6*tan(d*x + c) + C*a^7 + B*a^6*b + 9*C*a^5*b^2 - 11*B*a^4*b
^3 - 4*C*a^3*b^4)/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*(b*tan(d*x + c)
+ a)^2))/d
```

maple [B] time = 0.34, size = 495, normalized size = 2.62

$$-\frac{a^2B}{2db\left(a^2+b^2\right)(a+b\tan(dx+c))^2}+\frac{a^3C}{2d\,b^2\left(a^2+b^2\right)(a+b\tan(dx+c))^2}-\frac{3b\,a^2\ln(a+b\tan(dx+c))\,B}{d\left(a^2+b^2\right)^3}+\frac{\ln(a}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x)
```

```
[Out] -1/2/d*a^2/b/(a^2+b^2)/(a+b*tan(d*x+c))^2*B+1/2/d*a^3/b^2/(a^2+b^2)/(a+b*ta
n(d*x+c))^2*C-3/d*b*a^2/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*B+1/d/(a^2+b^2)^3*ln
(a+b*tan(d*x+c))*b^3*B+1/d*a^3/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*C-3/d/(a^2+b^
2)^3*ln(a+b*tan(d*x+c))*C*a*b^2+2/d*a/(a^2+b^2)^2*b/(a+b*tan(d*x+c))*B-1/d/
b^2*a^4/(a^2+b^2)^2/(a+b*tan(d*x+c))*C-3/d*a^2/(a^2+b^2)^2/(a+b*tan(d*x+c))
*C+3/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*a^2*b*B-1/2/d/(a^2+b^2)^3*ln(1+tan(
d*x+c)^2)*b^3*B-1/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*C*a^3+3/2/d/(a^2+b^2)^
3*ln(1+tan(d*x+c)^2)*C*a*b^2-1/d/(a^2+b^2)^3*B*arctan(tan(d*x+c))*a^3+3/d/(
a^2+b^2)^3*B*arctan(tan(d*x+c))*a*b^2-3/d/(a^2+b^2)^3*C*arctan(tan(d*x+c))*
a^2*b+1/d/(a^2+b^2)^3*C*arctan(tan(d*x+c))*b^3
```

maxima [A] time = 0.87, size = 333, normalized size = 1.76

$$-\frac{\frac{2\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{2\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)\log(b\tan(dx+c)+a)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{\cdots}{a^6b^{\cdots}}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="maxima")
```

```
[Out] -1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) - 2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(b*tan(d*
x + c) + a)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*
a*b^2 + B*b^3)*log(tan(d*x + c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)
+ (C*a^5 + B*a^4*b + 5*C*a^3*b^2 - 3*B*a^2*b^3 + 2*(C*a^4*b + 3*C*a^2*b^3 -
 2*B*a*b^4)*tan(d*x + c))/(a^6*b^2 + 2*a^4*b^4 + a^2*b^6 + (a^4*b^4 + 2*a^2
*b^6 + b^8)*tan(d*x + c)^2 + 2*(a^5*b^3 + 2*a^3*b^5 + a*b^7)*tan(d*x + c)))
/d
```

mupad [B] time = 9.18, size = 280, normalized size = 1.48

$$\frac{\ln\left(a+b\tan\left(c+d\,x\right)\right)\,\left(C\,a^3-3\,B\,a^2\,b-3\,C\,a\,b^2+B\,b^3\right)}{d\,{\left(a^2+b^2\right)}^3}-\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(-C+B\,1\mathrm{i}\right)}{2\,d\,\left(-a^3-a^2\,b\,3\mathrm{i}+3\,a\,b^2+b^3\,1\mathrm{i}\right)}-\frac{\ln\left(\tan\left(c+d\,x\right)+\cdots\right.}{2\,d\,\left(-a^3\,1\mathrm{i}-3\,a^2\,b\cdots\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((tan(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(a + b*tan(c + d*x))*(B*b^3 + C*a^3 - 3*B*a^2*b - 3*C*a*b^2))/(d*(a^2 +
 b^2)^3) - (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(3*a*b^2 - a^2*b*3i - a
```

```
^3 + b^3*1i)) - (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a*b^2*3i - 3*a^2*
b - a^3*1i + b^3)) - ((a*(C*a^4 + 5*C*a^2*b^2 - 3*B*a*b^3 + B*a^3*b))/(2*b^
2*(a^4 + b^4 + 2*a^2*b^2)) + (tan(c + d*x)*(C*a^4 + 3*C*a^2*b^2 - 2*B*a*b^3
))/(b*(a^4 + b^4 + 2*a^2*b^2)))/(d*(a^2 + b^2*tan(c + d*x)^2 + 2*a*b*tan(c
+ d*x)))
```

sympy [**F(-2)**] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,x)
[Out] Exception raised: AttributeError
```

3.41 $\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx$

Optimal. Leaf size=179

$$\frac{a(bB-aC)}{2bd\left(a^2+b^2\right)(a+b\tan(c+dx))^2}+\frac{a^2B+2abC-b^2B}{d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}-\frac{\left(a^3B+3a^2bC-3ab^2B-b^3C\right)\log(a\cos(c+}{d\left(a^2+b^2\right)^3}$$

```
[Out] (3*B*a^2*b-B*b^3-C*a^3+3*C*a*b^2)*x/(a^2+b^2)^3-(B*a^3-3*B*a*b^2+3*C*a^2*b-
C*b^3)*ln(a*cos(d*x+c)+b*sin(d*x+c))/(a^2+b^2)^3/d+1/2*a*(B*b-C*a)/b/(a^2+b
^2)/d/(a+b*tan(d*x+c))^2+(B*a^2-B*b^2+2*C*a*b)/(a^2+b^2)^2/d/(a+b*tan(d*x+c
))
```

Rubi [A] time = 0.25, antiderivative size = 179, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 32, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.125, Rules used = {3628, 3529, 3531, 3530}

$$\frac{a(bB-aC)}{2bd\left(a^2+b^2\right)(a+b\tan(c+dx))^2}+\frac{a^2B+2abC-b^2B}{d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}-\frac{\left(3a^2bC+a^3B-3ab^2B-b^3C\right)\log(a\cos(c+}{d\left(a^2+b^2\right)^3}$$

Antiderivative was successfully verified.

```
[In] Int[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x])^3,x]
```

```
[Out] ((3*a^2*b*B - b^3*B - a^3*C + 3*a*b^2*C)*x)/(a^2 + b^2)^3 - ((a^3*B - 3*a*b
^2*B + 3*a^2*b*C - b^3*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/((a^2 + b^2
)^3*d) + (a*(b*B - a*C))/(2*b*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) + (a^2*
B - b^2*B + 2*a*b*C)/((a^2 + b^2)^2*d*(a + b*Tan[c + d*x]))
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$\int \frac{B\tan(c+dx)+C\tan^2(c+dx)}{(a+b\tan(c+dx))^3}\,dx = \frac{a(bB-aC)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{\int \frac{bB-aC+(aB+bC)\tan(c+dx)}{(a+b\tan(c+dx))^2}\,dx}{a^2+b^2}$$

$$= \frac{a(bB-aC)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{a^2B-b^2B+2abC}{\left(a^2+b^2\right)^2 d(a+b\tan(c+dx))} + \cdots$$

$$= \frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)x}{\left(a^2+b^2\right)^3} + \frac{a(bB-aC)}{2b\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \cdots$$

$$= \frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)x}{\left(a^2+b^2\right)^3} - \frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)\log(a\cdots}{\left(a^2+b^2\right)^3\cdots}$$

Mathematica [C] time = 4.07, size = 188, normalized size = 1.05

$$\frac{\frac{a(bB-aC)}{b\left(a^2+b^2\right)(a+b\tan(c+dx))^2} + \frac{2\left(a^2B+2abC-b^2B\right)}{\left(a^2+b^2\right)^2(a+b\tan(c+dx))} - \frac{2\left(a^3B+3a^2bC-3ab^2B-b^3C\right)\log(a+b\tan(c+dx))}{\left(a^2+b^2\right)^3} + \frac{(B+iC)\log(-\tan(c+dx)+i)}{(a+ib)^3} + \frac{(B-i\cdots}{\cdots}}{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[(B*Tan[c + d*x] + C*Tan[c + d*x]^2)/(a + b*Tan[c + d*x])^3,x]
```

```
[Out] (((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^3 + ((B - I*C)*Log[I + Tan[c +
 d*x]])/(a - I*b)^3 - (2*(a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*Log[a + b*
Tan[c + d*x]])/(a^2 + b^2)^3 + (a*(b*B - a*C))/(b*(a^2 + b^2)*(a + b*Tan[c
+ d*x])^2) + (2*(a^2*B - b^2*B + 2*a*b*C))/((a^2 + b^2)^2*(a + b*Tan[c + d*
x])))/(2*d)
```

fricas [B] time = 0.90, size = 488, normalized size = 2.73

$$-\frac{3\,Ca^4b-5\,Ba^3b^2-3\,Ca^2b^3+Bab^4+2\left(Ca^5-3\,Ba^4b-3\,Ca^3b^2+Ba^2b^3\right)dx-\left(Ca^4b-3\,Ba^3b^2-5\,Ca^2b^3+3\,B\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, algorithm="fr
icas")
```

```
[Out] -1/2*(3*C*a^4*b - 5*B*a^3*b^2 - 3*C*a^2*b^3 + B*a*b^4 + 2*(C*a^5 - 3*B*a^4*
b - 3*C*a^3*b^2 + B*a^2*b^3)*d*x - (C*a^4*b - 3*B*a^3*b^2 - 5*C*a^2*b^3 + 3
*B*a*b^4 - 2*(C*a^3*b^2 - 3*B*a^2*b^3 - 3*C*a*b^4 + B*b^5)*d*x)*tan(d*x + c
)^2 + (B*a^5 + 3*C*a^4*b - 3*B*a^3*b^2 - C*a^2*b^3 + (B*a^3*b^2 + 3*C*a^2*b
^3 - 3*B*a*b^4 - C*b^5)*tan(d*x + c)^2 + 2*(B*a^4*b + 3*C*a^3*b^2 - 3*B*a^2
*b^3 - C*a*b^4)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c)
+ a^2)/(tan(d*x + c)^2 + 1)) - 2*(C*a^5 - 2*B*a^4*b - 3*C*a^3*b^2 + 3*B*a^2
*b^3 + 2*C*a*b^4 - B*b^5 - 2*(C*a^4*b - 3*B*a^3*b^2 - 3*C*a^2*b^3 + B*a*b^4
)*d*x)*tan(d*x + c))/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*d*tan(d*x + c
)^2 + 2*(a^7*b + 3*a^5*b^3 + 3*a^3*b^5 + a*b^7)*d*tan(d*x + c) + (a^8 + 3*a
^6*b^2 + 3*a^4*b^4 + a^2*b^6)*d)
```

giac [B] time = 2.65, size = 410, normalized size = 2.29

$$-\frac{\frac{2\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\left(Ba^3b+3\,Ca^2b^2-3\,Bab^3-Cb^4\right)\log(|b\tan(dx+c)+a|)}{a^6b+3\,a^4b^3+3\,a^2b^5+b^7}-}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, algorithm="gi
ac")
```

```
[Out] -1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) - (B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(tan(d*x +
c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + 2*(B*a^3*b + 3*C*a^2*b^2 -
3*B*a*b^3 - C*b^4)*log(abs(b*tan(d*x + c) + a))/(a^6*b + 3*a^4*b^3 + 3*a^2*
b^5 + b^7) - (3*B*a^3*b^3*tan(d*x + c)^2 + 9*C*a^2*b^4*tan(d*x + c)^2 - 9*B
*a*b^5*tan(d*x + c)^2 - 3*C*b^6*tan(d*x + c)^2 + 8*B*a^4*b^2*tan(d*x + c) +
```

```
 22*C*a^3*b^3*tan(d*x + c) - 18*B*a^2*b^4*tan(d*x + c) - 2*C*a*b^5*tan(d*x
+ c) - 2*B*b^6*tan(d*x + c) - C*a^6 + 6*B*a^5*b + 11*C*a^4*b^2 - 7*B*a^3*b^
3 - B*a*b^5)/((a^6*b + 3*a^4*b^3 + 3*a^2*b^5 + b^7)*(b*tan(d*x + c) + a)^2)
)/d
```

maple [B] time = 0.29, size = 488, normalized size = 2.73

$$\frac{aB}{2d\left(a^2+b^2\right)\left(a+b\tan\left(dx+c\right)\right)^2}-\frac{a^2C}{2d\left(a^2+b^2\right)b\left(a+b\tan\left(dx+c\right)\right)^2}+\frac{a^2B}{d\left(a^2+b^2\right)^2\left(a+b\tan\left(dx+c\right)\right)}-\frac{}{d\left(a^2+\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x)
```

```
[Out] 1/2/d*a/(a^2+b^2)/(a+b*tan(d*x+c))^2*B-1/2/d*a^2/(a^2+b^2)/b/(a+b*tan(d*x+c
))^2*C+1/d*a^2/(a^2+b^2)^2/(a+b*tan(d*x+c))*B-1/d/(a^2+b^2)^2/(a+b*tan(d*x+
c))*b^2*B+2/d/(a^2+b^2)^2/(a+b*tan(d*x+c))*C*a*b-1/d*a^3/(a^2+b^2)^3*ln(a+b
*tan(d*x+c))*B+3/d*a/(a^2+b^2)^3*b^2*ln(a+b*tan(d*x+c))*B-3/d*a^2/(a^2+b^2)
^3*b*ln(a+b*tan(d*x+c))*C+1/d/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*b^3*C+1/2/d/(a
^2+b^2)^3*ln(1+tan(d*x+c)^2)*a^3*B-3/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*B*a
*b^2+3/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*C*a^2*b-1/2/d/(a^2+b^2)^3*ln(1+ta
n(d*x+c)^2)*b^3*C+3/d/(a^2+b^2)^3*B*arctan(tan(d*x+c))*a^2*b-1/d/(a^2+b^2)^
3*B*arctan(tan(d*x+c))*b^3-1/d/(a^2+b^2)^3*C*arctan(tan(d*x+c))*a^3+3/d/(a^
2+b^2)^3*C*arctan(tan(d*x+c))*a*b^2
```

maxima [A] time = 0.64, size = 330, normalized size = 1.84

$$-\frac{\frac{2\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)\log(b\tan(dx+c)+a)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{}{a^6}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, algorithm="ma
xima")
```

```
[Out] -1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) + 2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(b*tan(d*
x + c) + a)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - (B*a^3 + 3*C*a^2*b - 3*B*
a*b^2 - C*b^3)*log(tan(d*x + c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)
+ (C*a^4 - 3*B*a^3*b - 3*C*a^2*b^2 + B*a*b^3 - 2*(B*a^2*b^2 + 2*C*a*b^3 - B
*b^4)*tan(d*x + c))/(a^6*b + 2*a^4*b^3 + a^2*b^5 + (a^4*b^3 + 2*a^2*b^5 + b
^7)*tan(d*x + c)^2 + 2*(a^5*b^2 + 2*a^3*b^4 + a*b^6)*tan(d*x + c)))/d
```

mupad [B] time = 9.28, size = 282, normalized size = 1.58

$$\frac{\frac{\tan(c+d\,x)\left(B\,a^2\,b+2\,C\,a\,b^2-B\,b^3\right)}{a^4+2\,a^2\,b^2+b^4}-\frac{C\,a^4-3\,B\,a^3\,b-3\,C\,a^2\,b^2+B\,a\,b^3}{2\,b\left(a^4+2\,a^2\,b^2+b^4\right)}}{d\left(a^2+2\,a\,b\tan\left(c+d\,x\right)+b^2\tan\left(c+d\,x\right)^2\right)}-\frac{\ln\left(a+b\tan\left(c+d\,x\right)\right)\left(\frac{B\,a+3\,C\,b}{\left(a^2+b^2\right)^2}-\frac{4\,b^2\,(B\,a+C\,b)}{\left(a^2+b^2\right)^3}\right)}{d}-\frac{\ln\left(\tan\left(c-\right.\right.}{2\,d\left(-a^3\,1\mathrm{i}\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((B*tan(c + d*x) + C*tan(c + d*x)^2)/(a + b*tan(c + d*x))^3,x)
```

```
[Out] ((tan(c + d*x)*(B*a^2*b - B*b^3 + 2*C*a*b^2))/(a^4 + b^4 + 2*a^2*b^2) - (C*
a^4 - 3*C*a^2*b^2 + B*a*b^3 - 3*B*a^3*b)/(2*b*(a^4 + b^4 + 2*a^2*b^2)))/(d*
(a^2 + b^2*tan(c + d*x)^2 + 2*a*b*tan(c + d*x))) - (log(a + b*tan(c + d*x))
*((B*a + 3*C*b)/(a^2 + b^2)^2 - (4*b^2*(B*a + C*b))/(a^2 + b^2)^3))/d - (lo
g(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*b^2*3i + 3*a^2*b - a^3*1i - b^3))
- (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(3*a*b^2 + a^2*b*3i - a^3 - b^3*
1i))
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,x)
```

```
[Out] Exception raised: AttributeError
```

3.42 $\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$

Optimal. Leaf size=175

$$-\frac{bB-aC}{2d\left(a^2+b^2\right)(a+b\tan(c+dx))^2}-\frac{a^2(-C)+2abB+b^2C}{d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}+\frac{\left(a^3(-C)+3a^2bB+3ab^2C-b^3B\right)\log(a\cos}{d\left(a^2+b^2\right)^3}$$

```
[Out] (B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*x/(a^2+b^2)^3+(3*B*a^2*b-B*b^3-C*a^3+3*C*
a*b^2)*ln(a*cos(d*x+c)+b*sin(d*x+c))/(a^2+b^2)^3/d+1/2*(-B*b+C*a)/(a^2+b^2)
/d/(a+b*tan(d*x+c))^2+(-2*B*a*b+C*a^2-C*b^2)/(a^2+b^2)^2/d/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.32, antiderivative size = 175, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, integrand size = 38, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.105, Rules used = {3632, 3529, 3531, 3530}

$$-\frac{bB-aC}{2d\left(a^2+b^2\right)(a+b\tan(c+dx))^2}-\frac{a^2(-C)+2abB+b^2C}{d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}+\frac{\left(3a^2bB+a^3(-C)+3ab^2C-b^3B\right)\log(a\cos}{d\left(a^2+b^2\right)^3}$$

Antiderivative was successfully verified.

```
[In] Int[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x])
^3,x]
```

```
[Out] ((a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*x)/(a^2 + b^2)^3 + ((3*a^2*b*B - b
^3*B - a^3*C + 3*a*b^2*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/((a^2 + b^2
)^3*d) - (b*B - a*C)/(2*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) - (2*a*b*B -
a^2*C + b^2*C)/((a^2 + b^2)^2*d*(a + b*Tan[c + d*x]))
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rubi steps

$$\int \frac{\cot(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \int \frac{B+C\tan(c+dx)}{(a+b\tan(c+dx))^3}\,dx$$

$$= -\frac{bB-aC}{2\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{\int \frac{aB+bC-(bB-aC)\tan(c+dx)}{(a+b\tan(c+dx))^2}}{a^2+b^2}$$

$$= -\frac{bB-aC}{2\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} - \frac{2abB-a^2C+b^2}{\left(a^2+b^2\right)^2 d(a+b\tan(}$$

$$= \frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3} - \frac{bB-aC}{2\left(a^2+b^2\right)d(a+b\tan(c}$$

$$= \frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3} + \frac{\left(3a^2bB-b^3B-a^3C+3\right.}{}$$

Mathematica [C] time = 4.77, size = 243, normalized size = 1.39

$$-\frac{(bB-aC)\left(\frac{b\left(\frac{\left(a^2+b^2\right)\left(5a^2+4ab\tan(c+dx)+b^2\right)}{(a+b\tan(c+dx))^2}+\left(2b^2-6a^2\right)\log(a+b\tan(c+dx))\right)}{\left(a^2+b^2\right)^3}+\frac{i\log(-\tan(c+dx)+i)}{(a+ib)^3}-\frac{\log(\tan(c+dx)+i)}{(b+ia)^3}\right)+C\left(\frac{2b\left(\frac{a^2+b^2}{a+b\tan(c+}\right.}{}\right.}{2bd}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Cot[c + d*x]*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c +
 d*x])^3,x]
```

```
[Out] -1/2*(C*((I*Log[I - Tan[c + d*x]])/(a + I*b)^2 - (I*Log[I + Tan[c + d*x]])/
(a - I*b)^2 + (2*b*(-2*a*Log[a + b*Tan[c + d*x]] + (a^2 + b^2)/(a + b*Tan[c
 + d*x])))/(a^2 + b^2)^2) + (b*B - a*C)*((I*Log[I - Tan[c + d*x]])/(a + I*b
)^3 - Log[I + Tan[c + d*x]]/(I*a + b)^3 + (b*((-6*a^2 + 2*b^2)*Log[a + b*Ta
n[c + d*x]] + ((a^2 + b^2)*(5*a^2 + b^2 + 4*a*b*Tan[c + d*x]))/(a + b*Tan[c
 + d*x])^2))/(a^2 + b^2)^3))/(b*d)
```

fricas [B] time = 0.78, size = 482, normalized size = 2.75

$$\frac{5\,Ca^3b^2 - 7\,Ba^2b^3 - Cab^4 - Bb^5 + 2\left(Ba^5 + 3\,Ca^4b - 3\,Ba^3b^2 - Ca^2b^3\right)dx - \left(3\,Ca^3b^2 - 5\,Ba^2b^3 - 3\,Cab^4 + Bb^5\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="fricas")
```

```
[Out] 1/2*(5*C*a^3*b^2 - 7*B*a^2*b^3 - C*a*b^4 - B*b^5 + 2*(B*a^5 + 3*C*a^4*b - 3
*B*a^3*b^2 - C*a^2*b^3)*d*x - (3*C*a^3*b^2 - 5*B*a^2*b^3 - 3*C*a*b^4 + B*b^
5 - 2*(B*a^3*b^2 + 3*C*a^2*b^3 - 3*B*a*b^4 - C*b^5)*d*x)*tan(d*x + c)^2 - (
C*a^5 - 3*B*a^4*b - 3*C*a^3*b^2 + B*a^2*b^3 + (C*a^3*b^2 - 3*B*a^2*b^3 - 3*
C*a*b^4 + B*b^5)*tan(d*x + c)^2 + 2*(C*a^4*b - 3*B*a^3*b^2 - 3*C*a^2*b^3 +
B*a*b^4)*tan(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/
(tan(d*x + c)^2 + 1)) - 2*(2*C*a^4*b - 3*B*a^3*b^2 - 3*C*a^2*b^3 + 3*B*a*b^
4 + C*b^5 - 2*(B*a^4*b + 3*C*a^3*b^2 - 3*B*a^2*b^3 - C*a*b^4)*d*x)*tan(d*x
+ c))/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*d*tan(d*x + c)^2 + 2*(a^7*b
+ 3*a^5*b^3 + 3*a^3*b^5 + a*b^7)*d*tan(d*x + c) + (a^8 + 3*a^6*b^2 + 3*a^4*
b^4 + a^2*b^6)*d)
```

giac [B] time = 4.72, size = 409, normalized size = 2.34

$$\frac{\frac{2\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6} + \frac{\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6} - \frac{2\left(Ca^3b-3\,Ba^2b^2-3\,Cab^3+Bb^4\right)\log(|b\tan(dx+c)+a|)}{a^6b+3\,a^4b^3+3\,a^2b^5+b^7} + \cdots}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="giac")
```

```
[Out] 1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x + c
)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(C*a^3*b - 3*B*a^2*b^2 - 3
*C*a*b^3 + B*b^4)*log(abs(b*tan(d*x + c) + a))/(a^6*b + 3*a^4*b^3 + 3*a^2*b
^5 + b^7) + (3*C*a^3*b^2*tan(d*x + c)^2 - 9*B*a^2*b^3*tan(d*x + c)^2 - 9*C*
a*b^4*tan(d*x + c)^2 + 3*B*b^5*tan(d*x + c)^2 + 8*C*a^4*b*tan(d*x + c) - 22
*B*a^3*b^2*tan(d*x + c) - 18*C*a^2*b^3*tan(d*x + c) + 2*B*a*b^4*tan(d*x + c
) - 2*C*b^5*tan(d*x + c) + 6*C*a^5 - 14*B*a^4*b - 7*C*a^3*b^2 - 3*B*a^2*b^3
 - C*a*b^4 - B*b^5)/((a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)*(b*tan(d*x + c) +
a)^2))/d
```

maple [B] time = 0.74, size = 483, normalized size = 2.76

$$\frac{3b\,a^{2}\ln\left(a+b\tan\left(dx+c\right)\right)B}{d\left(a^{2}+b^{2}\right)^{3}}-\frac{\ln\left(a+b\tan\left(dx+c\right)\right)b^{3}B}{d\left(a^{2}+b^{2}\right)^{3}}-\frac{a^{3}\ln\left(a+b\tan\left(dx+c\right)\right)C}{d\left(a^{2}+b^{2}\right)^{3}}+\frac{3\ln\left(a+b\tan\left(dx+c\right)\right)C}{d\left(a^{2}+b^{2}\right)^{3}}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x)
```

```
[Out] 3/d*b*a^2/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*B-1/d/(a^2+b^2)^3*ln(a+b*tan(d*x+c
))*b^3*B-1/d*a^3/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*C+3/d/(a^2+b^2)^3*ln(a+b*ta
n(d*x+c))*C*a*b^2-1/2/d/(a^2+b^2)/(a+b*tan(d*x+c))^2*B*b+1/2/d/(a^2+b^2)/(a
+b*tan(d*x+c))^2*a*C-2/d*a/(a^2+b^2)^2*b/(a+b*tan(d*x+c))*B+1/d*a^2/(a^2+b^
2)^2/(a+b*tan(d*x+c))*C-1/d/(a^2+b^2)^2/(a+b*tan(d*x+c))*b^2*C-3/2/d/(a^2+b
^2)^3*ln(1+tan(d*x+c)^2)*a^2*b*B+1/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*b^3*B
+1/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*C*a^3-3/2/d/(a^2+b^2)^3*ln(1+tan(d*x+
c)^2)*C*a*b^2+1/d/(a^2+b^2)^3*B*arctan(tan(d*x+c))*a^3-3/d/(a^2+b^2)^3*B*ar
ctan(tan(d*x+c))*a*b^2+3/d/(a^2+b^2)^3*C*arctan(tan(d*x+c))*a^2*b-1/d/(a^2+
b^2)^3*C*arctan(tan(d*x+c))*b^3
```

maxima [A] time = 0.48, size = 321, normalized size = 1.83

$$\frac{\frac{2\left(Ba^{3}+3\,Ca^{2}b-3\,Bab^{2}-Cb^{3}\right)(dx+c)}{a^{6}+3\,a^{4}b^{2}+3\,a^{2}b^{4}+b^{6}}-\frac{2\left(Ca^{3}-3\,Ba^{2}b-3\,Cab^{2}+Bb^{3}\right)\log(b\tan(dx+c)+a)}{a^{6}+3\,a^{4}b^{2}+3\,a^{2}b^{4}+b^{6}}+\frac{\left(Ca^{3}-3\,Ba^{2}b-3\,Cab^{2}+Bb^{3}\right)\log\left(\tan(dx+c)^{2}+1\right)}{a^{6}+3\,a^{4}b^{2}+3\,a^{2}b^{4}+b^{6}}+\frac{}{a^{6}+2}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x, al
gorithm="maxima")
```

```
[Out] 1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) - 2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(b*tan(d*x
 + c) + a)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a
*b^2 + B*b^3)*log(tan(d*x + c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) +
```

```
(3*C*a^3 - 5*B*a^2*b - C*a*b^2 - B*b^3 + 2*(C*a^2*b - 2*B*a*b^2 - C*b^3)*t
an(d*x + c))/(a^6 + 2*a^4*b^2 + a^2*b^4 + (a^4*b^2 + 2*a^2*b^4 + b^6)*tan(d
*x + c)^2 + 2*(a^5*b + 2*a^3*b^3 + a*b^5)*tan(d*x + c)))/d
```

mupad [B] time = 8.94, size = 279, normalized size = 1.59

$$\frac{\ln(a+b\tan(c+d\,x))\left(\frac{3\,B\,b-C\,a}{\left(a^2+b^2\right)^2}-\frac{4\,b^2\,(B\,b-C\,a)}{\left(a^2+b^2\right)^3}\right)}{d}-\frac{\frac{-3\,C\,a^3+5\,B\,a^2\,b+C\,a\,b^2+B\,b^3}{2\left(a^4+2\,a^2\,b^2+b^4\right)}+\frac{\tan(c+d\,x)\left(-C\,a^2\,b+2\,B\,a\,b^2+C\,b^3\right)}{a^4+2\,a^2\,b^2+b^4}}{d\left(a^2+2\,a\,b\tan(c+d\,x)+b^2\tan(c+d\,x)^2\right)}+\frac{\ln(\tan(}{2\,d\left(-a^3\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((cot(c + d*x)*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x))
^3,x)
```

```
[Out] (log(a + b*tan(c + d*x))*((3*B*b - C*a)/(a^2 + b^2)^2 - (4*b^2*(B*b - C*a))
/(a^2 + b^2)^3))/d - ((B*b^3 - 3*C*a^3 + 5*B*a^2*b + C*a*b^2)/(2*(a^4 + b^4
 + 2*a^2*b^2)) + (tan(c + d*x)*(C*b^3 + 2*B*a*b^2 - C*a^2*b))/(a^4 + b^4 +
2*a^2*b^2))/(d*(a^2 + b^2*tan(c + d*x)^2 + 2*a*b*tan(c + d*x))) + (log(tan(
c + d*x) - 1i)*(B*1i - C))/(2*d*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i)) + (log
(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a*b^2*3i - 3*a^2*b - a^3*1i + b^3))
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,x)
```

```
[Out] Exception raised: AttributeError
```

3.43 $\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$

Optimal. Leaf size=215

$$\frac{B\log(\sin(c+dx))}{a^3d}+\frac{b(bB-aC)}{2ad\left(a^2+b^2\right)(a+b\tan(c+dx))^2}+\frac{b\left(-2a^3C+3a^2bB+b^3B\right)}{a^2d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}-\frac{x\left(a^3(-C)+3a^2bB+\right.}{\left(a^2+b^2\right)}$$

```
[Out] -(3*B*a^2*b-B*b^3-C*a^3+3*C*a*b^2)*x/(a^2+b^2)^3+B*ln(sin(d*x+c))/a^3/d-b*(
6*B*a^4*b+3*B*a^2*b^3+B*b^5-3*C*a^5+C*a^3*b^2)*ln(a*cos(d*x+c)+b*sin(d*x+c)
)/a^3/(a^2+b^2)^3/d+1/2*b*(B*b-C*a)/a/(a^2+b^2)/d/(a+b*tan(d*x+c))^2+b*(3*B
*a^2*b+B*b^3-2*C*a^3)/a^2/(a^2+b^2)^2/d/(a+b*tan(d*x+c))
```

Rubi [A] time = 0.68, antiderivative size = 215, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3632, 3609, 3649, 3651, 3530, 3475}

$$\frac{b(bB-aC)}{2ad\left(a^2+b^2\right)(a+b\tan(c+dx))^2}+\frac{b\left(3a^2bB-2a^3C+b^3B\right)}{a^2d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}-\frac{b\left(3a^2b^3B+a^3b^2C+6a^4bB-3a^5C+b^5B\right.}{a^3d\left(a^2+b\right.}$$

Antiderivative was successfully verified.

```
[In] Int[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^3,x]
```

```
[Out] -(((3*a^2*b*B - b^3*B - a^3*C + 3*a*b^2*C)*x)/(a^2 + b^2)^3) + (B*Log[Sin[c
 + d*x]])/(a^3*d) - (b*(6*a^4*b*B + 3*a^2*b^3*B + b^5*B - 3*a^5*C + a^3*b^2
*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/(a^3*(a^2 + b^2)^3*d) + (b*(b*B -
 a*C))/(2*a*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) + (b*(3*a^2*b*B + b^3*B -
 2*a^3*C))/(a^2*(a^2 + b^2)^2*d*(a + b*Tan[c + d*x]))
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1)
)/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{\cot^2(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \int \frac{\cot(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$

$$= \frac{b(bB-aC)}{2a\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{\int \frac{\cot(c+dx)\left(2\left(a^2+b^2\right)B-2a\right.}{}}{}$$

$$= \frac{b(bB-aC)}{2a\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} + \frac{b\left(3a^2bB+b^3B-\right.}{a^2\left(a^2+b^2\right)^2 d(a+b\,\mathrm{t}}$$

$$= -\frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)x}{\left(a^2+b^2\right)^3} + \frac{b(bB-aC}{2a\left(a^2+b^2\right)d(a+b\,\mathrm{ta}}$$

$$= -\frac{\left(3a^2bB-b^3B-a^3C+3ab^2C\right)x}{\left(a^2+b^2\right)^3} + \frac{B\log(\sin(c+dx))}{a^3d} -$$

Mathematica [C] time = 3.15, size = 223, normalized size = 1.04

$$\frac{\frac{2B\log(\tan(c+dx))}{a^3} + \frac{b(bB-aC)}{a\left(a^2+b^2\right)(a+b\tan(c+dx))^2} + \frac{2b\left(-2a^3C+3a^2bB+b^3B\right)}{a^2\left(a^2+b^2\right)^2(a+b\tan(c+dx))} - \frac{2b\left(-3a^5C+6a^4bB+a^3b^2C+3a^2b^3B+b^5B\right)\log(a+b\tan(c+dx))}{a^3\left(a^2+b^2\right)^3} - }{2d}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Cot[c + d*x]^2*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^3,x]
```

```
[Out] (-(((B + I*C)*Log[I - Tan[c + d*x]])/(a + I*b)^3) + (2*B*Log[Tan[c + d*x]])
/a^3 - ((B - I*C)*Log[I + Tan[c + d*x]])/(a - I*b)^3 - (2*b*(6*a^4*b*B + 3*
a^2*b^3*B + b^5*B - 3*a^5*C + a^3*b^2*C)*Log[a + b*Tan[c + d*x]])/(a^3*(a^2
 + b^2)^3) + (b*(b*B - a*C))/(a*(a^2 + b^2)*(a + b*Tan[c + d*x])^2) + (2*b*
(3*a^2*b*B + b^3*B - 2*a^3*C))/(a^2*(a^2 + b^2)^2*(a + b*Tan[c + d*x]))))/(2
*d)
```

fricas [B] time = 0.73, size = 683, normalized size = 3.18

$$-\frac{7\,Ca^5b^3 - 9\,Ba^4b^4 + Ca^3b^5 - 3\,Ba^2b^6 - 2\left(Ca^8 - 3\,Ba^7b - 3\,Ca^6b^2 + Ba^5b^3\right)dx - \left(5\,Ca^5b^3 - 7\,Ba^4b^4 - Ca^3b^5 - \right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="fricas")
```

```
[Out] -1/2*(7*C*a^5*b^3 - 9*B*a^4*b^4 + C*a^3*b^5 - 3*B*a^2*b^6 - 2*(C*a^8 - 3*B*
a^7*b - 3*C*a^6*b^2 + B*a^5*b^3)*d*x - (5*C*a^5*b^3 - 7*B*a^4*b^4 - C*a^3*b
^5 - B*a^2*b^6 + 2*(C*a^6*b^2 - 3*B*a^5*b^3 - 3*C*a^4*b^4 + B*a^3*b^5)*d*x)
*tan(d*x + c)^2 - (B*a^8 + 3*B*a^6*b^2 + 3*B*a^4*b^4 + B*a^2*b^6 + (B*a^6*b
^2 + 3*B*a^4*b^4 + 3*B*a^2*b^6 + B*b^8)*tan(d*x + c)^2 + 2*(B*a^7*b + 3*B*a
^5*b^3 + 3*B*a^3*b^5 + B*a*b^7)*tan(d*x + c))*log(tan(d*x + c)^2/(tan(d*x +
 c)^2 + 1)) - (3*C*a^7*b - 6*B*a^6*b^2 - C*a^5*b^3 - 3*B*a^4*b^4 - B*a^2*b^
6 + (3*C*a^5*b^3 - 6*B*a^4*b^4 - C*a^3*b^5 - 3*B*a^2*b^6 - B*b^8)*tan(d*x +
 c)^2 + 2*(3*C*a^6*b^2 - 6*B*a^5*b^3 - C*a^4*b^4 - 3*B*a^3*b^5 - B*a*b^7)*t
an(d*x + c))*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x +
 c)^2 + 1)) - 2*(3*C*a^6*b^2 - 4*B*a^5*b^3 - 3*C*a^4*b^4 + 3*B*a^3*b^5 + B*
a*b^7 + 2*(C*a^7*b - 3*B*a^6*b^2 - 3*C*a^5*b^3 + B*a^4*b^4)*d*x)*tan(d*x +
c))/((a^9*b^2 + 3*a^7*b^4 + 3*a^5*b^6 + a^3*b^8)*d*tan(d*x + c)^2 + 2*(a^10
*b + 3*a^8*b^3 + 3*a^6*b^5 + a^4*b^7)*d*tan(d*x + c) + (a^11 + 3*a^9*b^2 +
3*a^7*b^4 + a^5*b^6)*d)
```

giac [B] time = 8.67, size = 479, normalized size = 2.23

$$\frac{\frac{2\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)\log\left(\tan(dx+c)^2+1\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\left(3\,Ca^5b^2-6\,Ba^4b^3-Ca^3b^4-3\,Ba^2b^5-Bb^7\right)\log(|b\tan(dx+}{a^9b+3\,a^7b^3+3\,a^5b^5+a^3b^7}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="giac")
```

```
[Out] 1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) - (B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(tan(d*x + c
)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + 2*(3*C*a^5*b^2 - 6*B*a^4*b^3
 - C*a^3*b^4 - 3*B*a^2*b^5 - B*b^7)*log(abs(b*tan(d*x + c) + a))/(a^9*b + 3
*a^7*b^3 + 3*a^5*b^5 + a^3*b^7) + 2*B*log(abs(tan(d*x + c)))/a^3 - (9*C*a^5
*b^3*tan(d*x + c)^2 - 18*B*a^4*b^4*tan(d*x + c)^2 - 3*C*a^3*b^5*tan(d*x + c
)^2 - 9*B*a^2*b^6*tan(d*x + c)^2 - 3*B*b^8*tan(d*x + c)^2 + 22*C*a^6*b^2*ta
n(d*x + c) - 42*B*a^5*b^3*tan(d*x + c) - 2*C*a^4*b^4*tan(d*x + c) - 26*B*a^
3*b^5*tan(d*x + c) - 8*B*a*b^7*tan(d*x + c) + 14*C*a^7*b - 25*B*a^6*b^2 + 3
*C*a^5*b^3 - 19*B*a^4*b^4 + C*a^3*b^5 - 6*B*a^2*b^6)/((a^9 + 3*a^7*b^2 + 3*
a^5*b^4 + a^3*b^6)*(b*tan(d*x + c) + a)^2))/d
```

maple [B] time = 1.06, size = 540, normalized size = 2.51

$$\frac{3b^2B}{d\left(a^2+b^2\right)^2\left(a+b\tan\left(dx+c\right)\right)}+\frac{b^4B}{d\,a^2\left(a^2+b^2\right)^2\left(a+b\tan\left(dx+c\right)\right)}-\frac{2Cab}{d\left(a^2+b^2\right)^2\left(a+b\tan\left(dx+c\right)\right)}-\frac{6a\,b^2\ln\left(a\right.}{d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x)
```

```
[Out] 3/d/(a^2+b^2)^2/(a+b*tan(d*x+c))*b^2*B+1/d*b^4/a^2/(a^2+b^2)^2/(a+b*tan(d*x
+c))*B-2/d/(a^2+b^2)^2/(a+b*tan(d*x+c))*C*a*b-6/d*a/(a^2+b^2)^3*b^2*ln(a+b*
tan(d*x+c))*B-3/d*b^4/a/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*B-1/d*b^6/a^3/(a^2+b
^2)^3*ln(a+b*tan(d*x+c))*B+3/d*a^2/(a^2+b^2)^3*b*ln(a+b*tan(d*x+c))*C-1/d/(
a^2+b^2)^3*ln(a+b*tan(d*x+c))*b^3*C+1/2/d*b^2/a/(a^2+b^2)/(a+b*tan(d*x+c))^
2*B-1/2/d*b/(a^2+b^2)/(a+b*tan(d*x+c))^2*C+1/d*B/a^3*ln(tan(d*x+c))-1/2/d/(
a^2+b^2)^3*ln(1+tan(d*x+c)^2)*a^3*B+3/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*B*
a*b^2-3/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*C*a^2*b+1/2/d/(a^2+b^2)^3*ln(1+t
an(d*x+c)^2)*b^3*C-3/d/(a^2+b^2)^3*B*arctan(tan(d*x+c))*a^2*b+1/d/(a^2+b^2)
^3*B*arctan(tan(d*x+c))*b^3+1/d/(a^2+b^2)^3*C*arctan(tan(d*x+c))*a^3-3/d/(a
^2+b^2)^3*C*arctan(tan(d*x+c))*a*b^2
```

maxima [A] time = 0.93, size = 372, normalized size = 1.73

$$\frac{\frac{2\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\left(3\,Ca^5b-6\,Ba^4b^2-Ca^3b^3-3\,Ba^2b^4-Bb^6\right)\log(b\tan(dx+c)+a)}{a^9+3\,a^7b^2+3\,a^5b^4+a^3b^6}-\frac{\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)\log\left(\tan(dx+c)^2+\right.}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}}{2\,d}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^2*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="maxima")
```

```
[Out] 1/2*(2*(C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2 +
 3*a^2*b^4 + b^6) + 2*(3*C*a^5*b - 6*B*a^4*b^2 - C*a^3*b^3 - 3*B*a^2*b^4 -
B*b^6)*log(b*tan(d*x + c) + a)/(a^9 + 3*a^7*b^2 + 3*a^5*b^4 + a^3*b^6) - (B
*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*log(tan(d*x + c)^2 + 1)/(a^6 + 3*a^4*
b^2 + 3*a^2*b^4 + b^6) - (5*C*a^4*b - 7*B*a^3*b^2 + C*a^2*b^3 - 3*B*a*b^4 +
 2*(2*C*a^3*b^2 - 3*B*a^2*b^3 - B*b^5)*tan(d*x + c))/(a^8 + 2*a^6*b^2 + a^4
*b^4 + (a^6*b^2 + 2*a^4*b^4 + a^2*b^6)*tan(d*x + c)^2 + 2*(a^7*b + 2*a^5*b^
3 + a^3*b^5)*tan(d*x + c)) + 2*B*log(tan(d*x + c))/a^3)/d
```

mupad [B] time = 10.98, size = 315, normalized size = 1.47

$$\frac{\frac{-5\,C\,a^3\,b+7\,B\,a^2\,b^2-C\,a\,b^3+3\,B\,b^4}{2\,a\left(a^4+2\,a^2\,b^2+b^4\right)}+\frac{\tan(c+d\,x)\left(-2\,C\,a^3\,b^2+3\,B\,a^2\,b^3+B\,b^5\right)}{a^2\left(a^4+2\,a^2\,b^2+b^4\right)}}{d\,\left(a^2+2\,a\,b\tan\left(c+d\,x\right)+b^2\tan\left(c+d\,x\right)^2\right)}+\frac{B\,\ln\left(\tan\left(c+d\,x\right)\right)}{a^3\,d}+\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(-C+B\,1\right.}{2\,d\,\left(-a^3\,1\mathrm{i}+3\,a^2\,b+a\,b^2\,3\mathrm{i}-b\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((cot(c + d*x)^2*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^3,x)
```

```
[Out] ((3*B*b^4 + 7*B*a^2*b^2 - C*a*b^3 - 5*C*a^3*b)/(2*a*(a^4 + b^4 + 2*a^2*b^2)
) + (tan(c + d*x)*(B*b^5 + 3*B*a^2*b^3 - 2*C*a^3*b^2))/(a^2*(a^4 + b^4 + 2*
a^2*b^2)))/(d*(a^2 + b^2*tan(c + d*x)^2 + 2*a*b*tan(c + d*x))) + (B*log(tan
(c + d*x)))/(a^3*d) + (log(tan(c + d*x) - 1i)*(B*1i - C))/(2*d*(a*b^2*3i +
3*a^2*b - a^3*1i - b^3)) + (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(3*a*b^
2 + a^2*b*3i - a^3 - b^3*1i)) - (b*log(a + b*tan(c + d*x))*(B*b^5 - 3*C*a^5
 + 3*B*a^2*b^3 + C*a^3*b^2 + 6*B*a^4*b))/(a^3*d*(a^2 + b^2)^3)
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**2*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,
x)
```

```
[Out] Exception raised: AttributeError
```

3.44 $\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx$

Optimal. Leaf size=287

$$-\frac{(3bB-aC)\log(\sin(c+dx))}{a^4d}-\frac{b\left(2a^2B-abC+3b^2B\right)}{2a^2d\left(a^2+b^2\right)(a+b\tan(c+dx))^2}-\frac{x\left(a^3B+3a^2bC-3ab^2B-b^3C\right)}{\left(a^2+b^2\right)^3}-\frac{b\left(a^4B-3a\right.}{a^3d\left(a^2\right.}$$

```
[Out] -(B*a^3-3*B*a*b^2+3*C*a^2*b-C*b^3)*x/(a^2+b^2)^3-(3*B*b-C*a)*ln(sin(d*x+c))
/a^4/d+b^2*(10*B*a^4*b+9*B*a^2*b^3+3*B*b^5-6*C*a^5-3*C*a^3*b^2-C*a*b^4)*ln(
a*cos(d*x+c)+b*sin(d*x+c))/a^4/(a^2+b^2)^3/d-1/2*b*(2*B*a^2+3*B*b^2-C*a*b)/
a^2/(a^2+b^2)/d/(a+b*tan(d*x+c))^2-B*cot(d*x+c)/a/d/(a+b*tan(d*x+c))^2-b*(B
*a^4+6*B*a^2*b^2+3*B*b^4-3*C*a^3*b-C*a*b^3)/a^3/(a^2+b^2)^2/d/(a+b*tan(d*x+
c))
```

Rubi [A] time = 0.94, antiderivative size = 287, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, integrand size = 40, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.150, Rules used = {3632, 3609, 3649, 3651, 3530, 3475}

$$-\frac{b\left(6a^2b^2B-3a^3bC+a^4B-ab^3C+3b^4B\right)}{a^3d\left(a^2+b^2\right)^2(a+b\tan(c+dx))}-\frac{b\left(2a^2B-abC+3b^2B\right)}{2a^2d\left(a^2+b^2\right)(a+b\tan(c+dx))^2}+\frac{b^2\left(9a^2b^3B-3a^3b^2C+10a^4bB\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c + d*x
])^3,x]
```

```
[Out] -(((a^3*B - 3*a*b^2*B + 3*a^2*b*C - b^3*C)*x)/(a^2 + b^2)^3) - ((3*b*B - a*
C)*Log[Sin[c + d*x]])/(a^4*d) + (b^2*(10*a^4*b*B + 9*a^2*b^3*B + 3*b^5*B -
6*a^5*C - 3*a^3*b^2*C - a*b^4*C)*Log[a*Cos[c + d*x] + b*Sin[c + d*x]])/(a^4
*(a^2 + b^2)^3*d) - (b*(2*a^2*B + 3*b^2*B - a*b*C))/(2*a^2*(a^2 + b^2)*d*(a
 + b*Tan[c + d*x])^2) - (B*Cot[c + d*x])/(a*d*(a + b*Tan[c + d*x])^2) - (b*
(a^4*B + 6*a^2*b^2*B + 3*b^4*B - 3*a^3*b*C - a*b^3*C))/(a^3*(a^2 + b^2)^2*d
*(a + b*Tan[c + d*x]))
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
```

```
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3609

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Si
mp[(b*(A*b - a*B)*(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1)
)/(f*(m + 1)*(b*c - a*d)*(a^2 + b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^
2 + b^2)), Int[(a + b*Tan[e + f*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[b*B
*(b*c*(m + 1) + a*d*(n + 1)) + A*(a*(b*c - a*d)*(m + 1) - b^2*d*(m + n + 2)
) - (A*b - a*B)*(b*c - a*d)*(m + 1)*Tan[e + f*x] - b*d*(A*b - a*B)*(m + n +
 2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, n}, x] &&
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &
& (IntegerQ[m] || IntegersQ[2*m, 2*n]) &&  !(ILtQ[n, -1] && ( !IntegerQ[m]
|| (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3632

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_
.) + (f_.)*(x_)]^2), x_Symbol] :> Dist[1/b^2, Int[(a + b*Tan[e + f*x])^(m +
 1)*(c + d*Tan[e + f*x])^n*(b*B - a*C + b*C*Tan[e + f*x]), x], x] /; FreeQ[
{a, b, c, d, e, f, A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && EqQ[A*b^2 - a
*b*B + a^2*C, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
```

```
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{\cot^3(c+dx)\left(B\tan(c+dx)+C\tan^2(c+dx)\right)}{(a+b\tan(c+dx))^3}\,dx = \int \frac{\cot^2(c+dx)(B+C\tan(c+dx))}{(a+b\tan(c+dx))^3}\,dx$$

$$= -\frac{B\cot(c+dx)}{ad(a+b\tan(c+dx))^2} - \frac{\int \frac{\cot(c+dx)\left(3bB-aC+aB\tan(c+dx)+3bI\right.}{(a+b\tan(c+dx))^3}}{a}$$

$$= -\frac{b\left(2a^2B+3b^2B-abC\right)}{2a^2\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx}$$

$$= -\frac{b\left(2a^2B+3b^2B-abC\right)}{2a^2\left(a^2+b^2\right)d(a+b\tan(c+dx))^2} - \frac{B\cot(c+dx)}{ad(a+b\tan(c+dx}$$

$$= -\frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3} - \frac{b\left(2a^2B+3b^2B\right.}{2a^2\left(a^2+b^2\right)d(a+b\,t}$$

$$= -\frac{\left(a^3B-3ab^2B+3a^2bC-b^3C\right)x}{\left(a^2+b^2\right)^3} - \frac{(3bB-aC)\log(\sin(c}{a^4d}$$

Mathematica [C] time = 6.42, size = 288, normalized size = 1.00

$$-\frac{(3bB-aC)\log(\tan(c+dx))}{a^4d}-\frac{B\cot(c+dx)}{a^3d}-\frac{b^2(bB-aC)}{2a^2d\left(a^2+b^2\right)(a+b\tan(c+dx))^2}-\frac{b^2\left(-3a^3C+4a^2bB-ab^2C+\right.}{a^3d\left(a^2+b^2\right)^2(a+b\tan(c+}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Cot[c + d*x]^3*(B*Tan[c + d*x] + C*Tan[c + d*x]^2))/(a + b*Tan[c
 + d*x])^3,x]
```

```
[Out] -((B*Cot[c + d*x])/(a^3*d)) + ((B + I*C)*Log[I - Tan[c + d*x]])/(2*(I*a - b
)^3*d) - ((3*b*B - a*C)*Log[Tan[c + d*x]])/(a^4*d) - ((I*B + C)*Log[I + Tan
```

```
[c + d*x]])/(2*(a - I*b)^3*d) + (b^2*(10*a^4*b*B + 9*a^2*b^3*B + 3*b^5*B -
6*a^5*C - 3*a^3*b^2*C - a*b^4*C)*Log[a + b*Tan[c + d*x]])/(a^4*(a^2 + b^2)^
3*d) - (b^2*(b*B - a*C))/(2*a^2*(a^2 + b^2)*d*(a + b*Tan[c + d*x])^2) - (b^
2*(4*a^2*b*B + 2*b^3*B - 3*a^3*C - a*b^2*C))/(a^3*(a^2 + b^2)^2*d*(a + b*Ta
n[c + d*x]))
```

fricas [B] time = 0.89, size = 917, normalized size = 3.20

$$-\frac{2\,Ba^{9} + 6\,Ba^{7}b^{2} + 6\,Ba^{5}b^{4} + 2\,Ba^{3}b^{6} + \left(7\,Ca^{5}b^{4} - 9\,Ba^{4}b^{5} + Ca^{3}b^{6} - 3\,Ba^{2}b^{7} + 2\left(Ba^{7}b^{2} + 3\,Ca^{6}b^{3} - 3\,Ba^{5}b^{4}\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="fricas")
```

```
[Out] -1/2*(2*B*a^9 + 6*B*a^7*b^2 + 6*B*a^5*b^4 + 2*B*a^3*b^6 + (7*C*a^5*b^4 - 9*
B*a^4*b^5 + C*a^3*b^6 - 3*B*a^2*b^7 + 2*(B*a^7*b^2 + 3*C*a^6*b^3 - 3*B*a^5*
b^4 - C*a^4*b^5)*d*x)*tan(d*x + c)^3 + 2*(B*a^7*b^2 + 4*C*a^6*b^3 - 2*B*a^5
*b^4 - 3*C*a^4*b^5 + 6*B*a^3*b^6 - C*a^2*b^7 + 3*B*a*b^8 + 2*(B*a^8*b + 3*C
*a^7*b^2 - 3*B*a^6*b^3 - C*a^5*b^4)*d*x)*tan(d*x + c)^2 - ((C*a^7*b^2 - 3*B
*a^6*b^3 + 3*C*a^5*b^4 - 9*B*a^4*b^5 + 3*C*a^3*b^6 - 9*B*a^2*b^7 + C*a*b^8
- 3*B*b^9)*tan(d*x + c)^3 + 2*(C*a^8*b - 3*B*a^7*b^2 + 3*C*a^6*b^3 - 9*B*a^
5*b^4 + 3*C*a^4*b^5 - 9*B*a^3*b^6 + C*a^2*b^7 - 3*B*a*b^8)*tan(d*x + c)^2 +
 (C*a^9 - 3*B*a^8*b + 3*C*a^7*b^2 - 9*B*a^6*b^3 + 3*C*a^5*b^4 - 9*B*a^4*b^5
 + C*a^3*b^6 - 3*B*a^2*b^7)*tan(d*x + c))*log(tan(d*x + c)^2/(tan(d*x + c)^
2 + 1)) + ((6*C*a^5*b^4 - 10*B*a^4*b^5 + 3*C*a^3*b^6 - 9*B*a^2*b^7 + C*a*b^
8 - 3*B*b^9)*tan(d*x + c)^3 + 2*(6*C*a^6*b^3 - 10*B*a^5*b^4 + 3*C*a^4*b^5 -
 9*B*a^3*b^6 + C*a^2*b^7 - 3*B*a*b^8)*tan(d*x + c)^2 + (6*C*a^7*b^2 - 10*B*
a^6*b^3 + 3*C*a^5*b^4 - 9*B*a^4*b^5 + C*a^3*b^6 - 3*B*a^2*b^7)*tan(d*x + c)
)*log((b^2*tan(d*x + c)^2 + 2*a*b*tan(d*x + c) + a^2)/(tan(d*x + c)^2 + 1))
 + (4*B*a^8*b + 12*B*a^6*b^3 - 9*C*a^5*b^4 + 23*B*a^4*b^5 - 3*C*a^3*b^6 + 9
*B*a^2*b^7 + 2*(B*a^9 + 3*C*a^8*b - 3*B*a^7*b^2 - C*a^6*b^3)*d*x)*tan(d*x +
 c))/((a^10*b^2 + 3*a^8*b^4 + 3*a^6*b^6 + a^4*b^8)*d*tan(d*x + c)^3 + 2*(a^
11*b + 3*a^9*b^3 + 3*a^7*b^5 + a^5*b^7)*d*tan(d*x + c)^2 + (a^12 + 3*a^10*b
^2 + 3*a^8*b^4 + a^6*b^6)*d*tan(d*x + c))
```

giac [A] time = 9.82, size = 560, normalized size = 1.95

$$-\frac{\frac{2\left(Ba^{3}+3\,Ca^{2}b-3\,Bab^{2}-Cb^{3}\right)(dx+c)}{a^{6}+3\,a^{4}b^{2}+3\,a^{2}b^{4}+b^{6}} + \frac{\left(Ca^{3}-3\,Ba^{2}b-3\,Cab^{2}+Bb^{3}\right)\log\left(\tan(dx+c)^{2}+1\right)}{a^{6}+3\,a^{4}b^{2}+3\,a^{2}b^{4}+b^{6}} + \frac{2\left(6\,Ca^{5}b^{3}-10\,Ba^{4}b^{4}+3\,Ca^{3}b^{5}-9\,Ba^{2}b^{6}+Cab^{7}-3\,Bb^{8}\right)\mathrm{l}}{a^{10}b+3\,a^{8}b^{3}+3\,a^{6}b^{5}+a^{4}b^{7}}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="giac")
```

```
[Out] -1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x +
c)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + 2*(6*C*a^5*b^3 - 10*B*a^4*b
^4 + 3*C*a^3*b^5 - 9*B*a^2*b^6 + C*a*b^7 - 3*B*b^8)*log(abs(b*tan(d*x + c)
+ a))/(a^10*b + 3*a^8*b^3 + 3*a^6*b^5 + a^4*b^7) - (18*C*a^5*b^4*tan(d*x +
c)^2 - 30*B*a^4*b^5*tan(d*x + c)^2 + 9*C*a^3*b^6*tan(d*x + c)^2 - 27*B*a^2*
b^7*tan(d*x + c)^2 + 3*C*a*b^8*tan(d*x + c)^2 - 9*B*b^9*tan(d*x + c)^2 + 42
*C*a^6*b^3*tan(d*x + c) - 68*B*a^5*b^4*tan(d*x + c) + 26*C*a^4*b^5*tan(d*x
+ c) - 66*B*a^3*b^6*tan(d*x + c) + 8*C*a^2*b^7*tan(d*x + c) - 22*B*a*b^8*ta
n(d*x + c) + 25*C*a^7*b^2 - 39*B*a^6*b^3 + 19*C*a^5*b^4 - 41*B*a^4*b^5 + 6*
C*a^3*b^6 - 14*B*a^2*b^7)/((a^10 + 3*a^8*b^2 + 3*a^6*b^4 + a^4*b^6)*(b*tan(
d*x + c) + a)^2) - 2*(C*a - 3*B*b)*log(abs(tan(d*x + c)))/a^4 + 2*(C*a*tan(
d*x + c) - 3*B*b*tan(d*x + c) + B*a)/(a^4*tan(d*x + c)))/d
```

maple [B] time = 0.93, size = 651, normalized size = 2.27

$$-\frac{4b^3B}{da\left(a^2+b^2\right)^2\left(a+b\tan\left(dx+c\right)\right)}-\frac{2b^5B}{d\,a^3\left(a^2+b^2\right)^2\left(a+b\tan\left(dx+c\right)\right)}+\frac{3b^2C}{d\left(a^2+b^2\right)^2\left(a+b\tan\left(dx+c\right)\right)}+\frac{}{d\,a^2\left(a^2\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x)
```

```
[Out] -4/d*b^3/a/(a^2+b^2)^2/(a+b*tan(d*x+c))*B-2/d*b^5/a^3/(a^2+b^2)^2/(a+b*tan(
d*x+c))*B+3/d/(a^2+b^2)^2/(a+b*tan(d*x+c))*b^2*C+1/d*b^4/a^2/(a^2+b^2)^2/(a
+b*tan(d*x+c))*C+10/d/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*b^3*B+9/d*b^5/a^2/(a^2
+b^2)^3*ln(a+b*tan(d*x+c))*B+3/d*b^7/a^4/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*B-6
/d/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*C*a*b^2-3/d*b^4/a/(a^2+b^2)^3*ln(a+b*tan(
d*x+c))*C-1/d*b^6/a^3/(a^2+b^2)^3*ln(a+b*tan(d*x+c))*C-1/2/d*b^3/a^2/(a^2+b
^2)/(a+b*tan(d*x+c))^2*B+1/2/d*b^2/a/(a^2+b^2)/(a+b*tan(d*x+c))^2*C-1/d*B/a
^3/tan(d*x+c)-3/d/a^4*ln(tan(d*x+c))*B*b+1/d/a^3*ln(tan(d*x+c))*C+3/2/d/(a^
2+b^2)^3*ln(1+tan(d*x+c)^2)*a^2*b*B-1/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*b^
3*B-1/2/d/(a^2+b^2)^3*ln(1+tan(d*x+c)^2)*C*a^3+3/2/d/(a^2+b^2)^3*ln(1+tan(d
*x+c)^2)*C*a*b^2-1/d/(a^2+b^2)^3*B*arctan(tan(d*x+c))*a^3+3/d/(a^2+b^2)^3*B
*arctan(tan(d*x+c))*a*b^2-3/d/(a^2+b^2)^3*C*arctan(tan(d*x+c))*a^2*b+1/d/(a
^2+b^2)^3*C*arctan(tan(d*x+c))*b^3
```

maxima [A] time = 0.72, size = 454, normalized size = 1.58

$$-\frac{\frac{2\left(Ba^3+3\,Ca^2b-3\,Bab^2-Cb^3\right)(dx+c)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{2\left(6\,Ca^5b^2-10\,Ba^4b^3+3\,Ca^3b^4-9\,Ba^2b^5+Cab^6-3\,Bb^7\right)\log(b\tan(dx+c)+a)}{a^{10}+3\,a^8b^2+3\,a^6b^4+a^4b^6}+\frac{\left(Ca^3-3\,Ba^2b-3\,Cab^2+Bb^3\right)\log}{a^6+3\,a^4b^2+3\,a^2b^4}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)^3*(B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^3,x,
algorithm="maxima")
```

```
[Out] -1/2*(2*(B*a^3 + 3*C*a^2*b - 3*B*a*b^2 - C*b^3)*(d*x + c)/(a^6 + 3*a^4*b^2
+ 3*a^2*b^4 + b^6) + 2*(6*C*a^5*b^2 - 10*B*a^4*b^3 + 3*C*a^3*b^4 - 9*B*a^2*
b^5 + C*a*b^6 - 3*B*b^7)*log(b*tan(d*x + c) + a)/(a^10 + 3*a^8*b^2 + 3*a^6*
b^4 + a^4*b^6) + (C*a^3 - 3*B*a^2*b - 3*C*a*b^2 + B*b^3)*log(tan(d*x + c)^2
 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + (2*B*a^6 + 4*B*a^4*b^2 + 2*B*a^
2*b^4 + 2*(B*a^4*b^2 - 3*C*a^3*b^3 + 6*B*a^2*b^4 - C*a*b^5 + 3*B*b^6)*tan(d
*x + c)^2 + (4*B*a^5*b - 7*C*a^4*b^2 + 17*B*a^3*b^3 - 3*C*a^2*b^4 + 9*B*a*b
^5)*tan(d*x + c))/((a^7*b^2 + 2*a^5*b^4 + a^3*b^6)*tan(d*x + c)^3 + 2*(a^8*
b + 2*a^6*b^3 + a^4*b^5)*tan(d*x + c)^2 + (a^9 + 2*a^7*b^2 + a^5*b^4)*tan(d
*x + c)) - 2*(C*a - 3*B*b)*log(tan(d*x + c))/a^4)/d
```

mupad [B] time = 13.99, size = 380, normalized size = 1.32

$$\frac{b^2\,\ln\left(a+b\,\tan\left(c+d\,x\right)\right)\,\left(-6\,C\,a^5+10\,B\,a^4\,b-3\,C\,a^3\,b^2+9\,B\,a^2\,b^3-C\,a\,b^4+3\,B\,b^5\right)}{a^4\,d\,{\left(a^2+b^2\right)}^3}-\frac{\ln\left(\tan\left(c+d\,x\right)-\mathrm{i}\right)\,\left(-\right.}{2\,d\,\left(-a^3-a^2\,b\,3\mathrm{i}+3\,a\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((cot(c + d*x)^3*(B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c + d*x
))^3,x)
```

```
[Out] (b^2*log(a + b*tan(c + d*x))*(3*B*b^5 - 6*C*a^5 + 9*B*a^2*b^3 - 3*C*a^3*b^2
 + 10*B*a^4*b - C*a*b^4))/(a^4*d*(a^2 + b^2)^3) - (log(tan(c + d*x) - 1i)*(
B*1i - C))/(2*d*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i)) - (log(tan(c + d*x))*(
3*B*b - C*a))/(a^4*d) - (log(tan(c + d*x) + 1i)*(B - C*1i))/(2*d*(a*b^2*3i
- 3*a^2*b - a^3*1i + b^3)) - (B/a + (tan(c + d*x)^2*(3*B*b^6 + 6*B*a^2*b^4
+ B*a^4*b^2 - 3*C*a^3*b^3 - C*a*b^5))/(a^3*(a^4 + b^4 + 2*a^2*b^2)) + (tan(
c + d*x)*(9*B*b^5 + 17*B*a^2*b^3 - 7*C*a^3*b^2 + 4*B*a^4*b - 3*C*a*b^4))/(2
*a^2*(a^4 + b^4 + 2*a^2*b^2)))/(d*(a^2*tan(c + d*x) + b^2*tan(c + d*x)^3 +
2*a*b*tan(c + d*x)^2))
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(cot(d*x+c)**3*(B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**3,
x)
```

```
[Out] Exception raised: AttributeError
```

3.45 $\int \tan^2(c+dx)(b\tan(c+dx))^n \left(A + B\tan(c+dx) + C\tan\right.$

Optimal. Leaf size=132

$$\frac{(A-C)(b\tan(c+dx))^{n+3}\,{}_2F_1\left(1,\frac{n+3}{2};\frac{n+5}{2};-\tan^2(c+dx)\right)}{b^3d(n+3)}+\frac{B(b\tan(c+dx))^{n+4}\,{}_2F_1\left(1,\frac{n+4}{2};\frac{n+6}{2};-\tan^2(c+dx)\right)}{b^4d(n+4)}$$

```
[Out] C*(b*tan(d*x+c))^(3+n)/b^3/d/(3+n)+(A-C)*hypergeom([1, 3/2+1/2*n],[5/2+1/2*
n],-tan(d*x+c)^2)*(b*tan(d*x+c))^(3+n)/b^3/d/(3+n)+B*hypergeom([1, 2+1/2*n]
,[3+1/2*n],-tan(d*x+c)^2)*(b*tan(d*x+c))^(4+n)/b^4/d/(4+n)
```

Rubi [A] time = 0.16, antiderivative size = 132, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, integrand size = 39, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.128, Rules used = {16, 3630, 3538, 3476, 364}

$$\frac{(A-C)(b\tan(c+dx))^{n+3}\,{}_2F_1\left(1,\frac{n+3}{2};\frac{n+5}{2};-\tan^2(c+dx)\right)}{b^3d(n+3)}+\frac{B(b\tan(c+dx))^{n+4}\,{}_2F_1\left(1,\frac{n+4}{2};\frac{n+6}{2};-\tan^2(c+dx)\right)}{b^4d(n+4)}$$

Antiderivative was successfully verified.

```
[In] Int[Tan[c + d*x]^2*(b*Tan[c + d*x])^n*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (C*(b*Tan[c + d*x])^(3 + n))/(b^3*d*(3 + n)) + ((A - C)*Hypergeometric2F1[1
, (3 + n)/2, (5 + n)/2, -Tan[c + d*x]^2]*(b*Tan[c + d*x])^(3 + n))/(b^3*d*(
3 + n)) + (B*Hypergeometric2F1[1, (4 + n)/2, (6 + n)/2, -Tan[c + d*x]^2]*(b
*Tan[c + d*x])^(4 + n))/(b^4*d*(4 + n))
```

Rule 16

```
Int[(u_.)*(v_)^(m_.)*((b_)*(v_))^(n_), x_Symbol] :> Dist[1/b^m, Int[u*(b*v)
^(m + n), x], x] /; FreeQ[{b, n}, x] && IntegerQ[m]
```

Rule 364

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[(a^
p*(c*x)^(m + 1)*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1, -((b*x^n)/a
)])/(c*(m + 1)), x] /; FreeQ[{a, b, c, m, n, p}, x] &&  !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 3476

```
Int[((b_.)*tan[(c_.) + (d_.)*(x_)])^(n_), x_Symbol] :> Dist[b/d, Subst[Int[
x^n/(b^2 + x^2), x], x, b*Tan[c + d*x]], x] /; FreeQ[{b, c, d, n}, x] &&  !
```

```
IntegerQ[n]
```

Rule 3538

```
Int[((b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) + (f_.)*(x
_)]), x_Symbol] :> Dist[c, Int[(b*Tan[e + f*x])^m, x], x] + Dist[d/b, Int[(
b*Tan[e + f*x])^(m + 1), x], x] /; FreeQ[{b, c, d, e, f, m}, x] && NeQ[c^2
+ d^2, 0] &&  !IntegerQ[2*m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int \tan^2(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)dx = \frac{\int (b\tan(c+dx))^{2+n}\left(A+B\tan(c+\right.}{b^2}$$

$$= \frac{C(b\tan(c+dx))^{3+n}}{b^3d(3+n)} + \frac{\int (b\tan(c+}{}$$

$$= \frac{C(b\tan(c+dx))^{3+n}}{b^3d(3+n)} + \frac{B\int (b\tan(c-}{b^3}$$

$$= \frac{C(b\tan(c+dx))^{3+n}}{b^3d(3+n)} + \frac{B\,\mathrm{Subst}\left(\int \frac{}{b}\right.}{}$$

$$= \frac{C(b\tan(c+dx))^{3+n}}{b^3d(3+n)} + \frac{(A-C)\,{}_2F_1\left(\right.}{}$$

Mathematica [A] time = 0.41, size = 110, normalized size = 0.83

$$\frac{\tan^3(c+dx)(b\tan(c+dx))^n\left((n+4)(A-C)\,{}_2F_1\left(1,\frac{n+3}{2};\frac{n+5}{2};-\tan^2(c+dx)\right)+B(n+3)\tan(c+dx)\,{}_2F_1\left(1,\frac{n+4}{2}\right.\right.}{d(n+3)(n+4)}$$

Antiderivative was successfully verified.

```
[In] Integrate[Tan[c + d*x]^2*(b*Tan[c + d*x])^n*(A + B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (Tan[c + d*x]^3*(b*Tan[c + d*x])^n*(C*(4 + n) + (A - C)*(4 + n)*Hypergeomet
ric2F1[1, (3 + n)/2, (5 + n)/2, -Tan[c + d*x]^2] + B*(3 + n)*Hypergeometric
2F1[1, (4 + n)/2, (6 + n)/2, -Tan[c + d*x]^2]*Tan[c + d*x]))/(d*(3 + n)*(4
+ n))
```

fricas [F] time = 1.12, size = 0, normalized size = 0.00

$$\text{integral}\left(\left(C\tan\left(dx+c\right)^{4}+B\tan\left(dx+c\right)^{3}+A\tan\left(dx+c\right)^{2}\right)\left(b\tan\left(dx+c\right)\right)^{n},x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] integral((C*tan(d*x + c)^4 + B*tan(d*x + c)^3 + A*tan(d*x + c)^2)*(b*tan(d*
x + c))^n, x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\left(C\tan\left(dx+c\right)^{2}+B\tan\left(dx+c\right)+A\right)\left(b\tan\left(dx+c\right)\right)^{n}\tan\left(dx+c\right)^{2}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*(b*tan(d*x + c))^n*tan(d*
x + c)^2, x)
```

maple [F] time = 1.11, size = 0, normalized size = 0.00

$$\int\left(\tan^{2}\left(dx+c\right)\right)\left(b\tan\left(dx+c\right)\right)^{n}\left(A+B\tan\left(dx+c\right)+C\left(\tan^{2}\left(dx+c\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)^2*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] int(tan(d*x+c)^2*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\left(C\tan\left(dx+c\right)^{2}+B\tan\left(dx+c\right)+A\right)\left(b\tan\left(dx+c\right)\right)^{n}\tan\left(dx+c\right)^{2}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^2*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*(b*tan(d*x + c))^n*tan(d*
x + c)^2, x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.01

$$\int \tan(c+d\,x)^2\,(b\tan(c+d\,x))^n\,\left(C\tan(c+d\,x)^2+B\tan(c+d\,x)+A\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(c + d*x)^2*(b*tan(c + d*x))^n*(A + B*tan(c + d*x) + C*tan(c + d*x)^
2),x)
```

```
[Out] int(tan(c + d*x)^2*(b*tan(c + d*x))^n*(A + B*tan(c + d*x) + C*tan(c + d*x)^
2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int (b\tan(c+dx))^n\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\tan^2(c+dx)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)**2*(b*tan(d*x+c))**n*(A+B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Integral((b*tan(c + d*x))**n*(A + B*tan(c + d*x) + C*tan(c + d*x)**2)*tan(c
 + d*x)**2, x)
```

3.46 $\int \tan^m(c+dx)(b\tan(c+dx))^n \left(A + B\tan(c + dx) + C \tan\right.$

Optimal. Leaf size=154

$$\frac{(A-C)\tan^{m+1}(c+dx)(b\tan(c+dx))^n\,{}_2F_1\left(1,\frac{1}{2}(m+n+1);\frac{1}{2}(m+n+3);-\tan^2(c+dx)\right)}{d(m+n+1)}+\frac{B\tan^{m+2}(c+dx)(b\,\mathrm{t}}{}$$

```
[Out] C*tan(d*x+c)^(1+m)*(b*tan(d*x+c))^n/d/(1+m+n)+(A-C)*hypergeom([1, 1/2+1/2*m
+1/2*n],[3/2+1/2*m+1/2*n],-tan(d*x+c)^2)*tan(d*x+c)^(1+m)*(b*tan(d*x+c))^n/
d/(1+m+n)+B*hypergeom([1, 1+1/2*m+1/2*n],[2+1/2*m+1/2*n],-tan(d*x+c)^2)*tan
(d*x+c)^(2+m)*(b*tan(d*x+c))^n/d/(2+m+n)
```

Rubi [A] time = 0.14, antiderivative size = 154, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, integrand size = 39, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.128, Rules used = {20, 3630, 3538, 3476, 364}

$$\frac{(A-C)\tan^{m+1}(c+dx)(b\tan(c+dx))^n\,{}_2F_1\left(1,\frac{1}{2}(m+n+1);\frac{1}{2}(m+n+3);-\tan^2(c+dx)\right)}{d(m+n+1)}+\frac{B\tan^{m+2}(c+dx)(b\,\mathrm{t}}{}$$

Antiderivative was successfully verified.

```
[In] Int[Tan[c + d*x]^m*(b*Tan[c + d*x])^n*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^
2),x]
```

```
[Out] (C*Tan[c + d*x]^(1 + m)*(b*Tan[c + d*x])^n)/(d*(1 + m + n)) + ((A - C)*Hype
rgeometric2F1[1, (1 + m + n)/2, (3 + m + n)/2, -Tan[c + d*x]^2]*Tan[c + d*x
]^(1 + m)*(b*Tan[c + d*x])^n)/(d*(1 + m + n)) + (B*Hypergeometric2F1[1, (2
+ m + n)/2, (4 + m + n)/2, -Tan[c + d*x]^2]*Tan[c + d*x]^(2 + m)*(b*Tan[c +
 d*x])^n)/(d*(2 + m + n))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] &&  !IntegerQ[m] &&  !IntegerQ[n] &&  !
IntegerQ[m + n]
```

Rule 364

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[(a^
p*(c*x)^(m + 1)*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1, -((b*x^n)/a
)])/(c*(m + 1)), x] /; FreeQ[{a, b, c, m, n, p}, x] &&  !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 3476

```
Int[((b_.)*tan[(c_.) + (d_.)*(x_)])^(n_), x_Symbol] :> Dist[b/d, Subst[Int[
x^n/(b^2 + x^2), x], x, b*Tan[c + d*x]], x] /; FreeQ[{b, c, d, n}, x] &&  !
IntegerQ[n]
```

Rule 3538

```
Int[((b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) + (f_.)*(x
_)]), x_Symbol] :> Dist[c, Int[(b*Tan[e + f*x])^m, x], x] + Dist[d/b, Int[(
b*Tan[e + f*x])^(m + 1), x], x] /; FreeQ[{b, c, d, e, f, m}, x] && NeQ[c^2
+ d^2, 0] &&  !IntegerQ[2*m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int \tan^m(c+dx)(b\tan(c+dx))^n \left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \left(\tan^{-n}(c+dx)(b\tan(c+dx))^n\right)\int \text{t}$$

$$= \frac{C\tan^{1+m}(c+dx)(b\tan(c+dx))^n}{d(1+m+n)} +$$

$$= \frac{C\tan^{1+m}(c+dx)(b\tan(c+dx))^n}{d(1+m+n)} +$$

$$= \frac{C\tan^{1+m}(c+dx)(b\tan(c+dx))^n}{d(1+m+n)} +$$

$$= \frac{C\tan^{1+m}(c+dx)(b\tan(c+dx))^n}{d(1+m+n)} +$$

Mathematica [A] time = 0.39, size = 115, normalized size = 0.75

$$\frac{\tan^{m+1}(c+dx)(b\tan(c+dx))^n\left(\frac{(A-C)\,{}_2F_1\left(1,\frac{1}{2}(m+n+1);\frac{1}{2}(m+n+3);-\tan^2(c+dx)\right)}{m+n+1}+\frac{B\tan(c+dx)\,{}_2F_1\left(1,\frac{1}{2}(m+n+2);\frac{1}{2}(m+n+4);-\tan^2(c\right.}{m+n+2}\right.}{d}$$

Antiderivative was successfully verified.

```
[In] Integrate[Tan[c + d*x]^m*(b*Tan[c + d*x])^n*(A + B*Tan[c + d*x] + C*Tan[c +
 d*x]^2),x]
```

```
[Out] (Tan[c + d*x]^(1 + m)*(b*Tan[c + d*x])^n*(C/(1 + m + n) + ((A - C)*Hypergeo
metric2F1[1, (1 + m + n)/2, (3 + m + n)/2, -Tan[c + d*x]^2])/(1 + m + n) +
(B*Hypergeometric2F1[1, (2 + m + n)/2, (4 + m + n)/2, -Tan[c + d*x]^2]*Tan[
c + d*x])/(2 + m + n)))/d
```

fricas [F] time = 0.76, size = 0, normalized size = 0.00

$$\text{integral}\left(\left(C\tan\left(dx+c\right)^{2}+B\tan\left(dx+c\right)+A\right)\left(b\tan\left(dx+c\right)\right)^{n}\tan\left(dx+c\right)^{m},x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="fricas")
```

```
[Out] integral((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*(b*tan(d*x + c))^n*tan(d*x
 + c)^m, x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\left(C\tan\left(dx+c\right)^{2}+B\tan\left(dx+c\right)+A\right)\left(b\tan\left(dx+c\right)\right)^{n}\tan\left(dx+c\right)^{m}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="giac")
```

```
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*(b*tan(d*x + c))^n*tan(d*
x + c)^m, x)
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int\left(\tan^{m}\left(dx+c\right)\right)\left(b\tan\left(dx+c\right)\right)^{n}\left(A+B\tan\left(dx+c\right)+C\left(\tan^{2}\left(dx+c\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)^m*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] int(tan(d*x+c)^m*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\left(C\tan\left(dx+c\right)^{2}+B\tan\left(dx+c\right)+A\right)\left(b\tan\left(dx+c\right)\right)^{n}\tan\left(dx+c\right)^{m}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^n*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x,
algorithm="maxima")
```

```
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*(b*tan(d*x + c))^n*tan(d*
x + c)^m, x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.01

$$\int \tan(c+d\,x)^m\,(b\tan(c+d\,x))^n\,\left(C\tan(c+d\,x)^2+B\tan(c+d\,x)+A\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(c + d*x)^m*(b*tan(c + d*x))^n*(A + B*tan(c + d*x) + C*tan(c + d*x)^
2),x)
```

```
[Out] int(tan(c + d*x)^m*(b*tan(c + d*x))^n*(A + B*tan(c + d*x) + C*tan(c + d*x)^
2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int (b\tan(c+dx))^n\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\tan^m(c+dx)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)**m*(b*tan(d*x+c))**n*(A+B*tan(d*x+c)+C*tan(d*x+c)**2),
x)
```

```
[Out] Integral((b*tan(c + d*x))**n*(A + B*tan(c + d*x) + C*tan(c + d*x)**2)*tan(c
 + d*x)**m, x)
```

3.47 $\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}\left(A+B\tan(c+dx)+C\tan\right.$

Optimal. Leaf size=170

$$\frac{2(A-C)\sqrt{b\tan(c+dx)}\,\tan^{m+1}(c+dx)\,{}_2F_1\left(1,\frac{1}{4}(2m+3);\frac{1}{4}(2m+7);-\tan^2(c+dx)\right)}{d(2m+3)}+\frac{2B\sqrt{b\tan(c+dx)}\,\tan^{m+}}{}$$

```
[Out] 2*C*(b*tan(d*x+c))^(1/2)*tan(d*x+c)^(1+m)/d/(3+2*m)+2*(A-C)*hypergeom([1, 3
/4+1/2*m],[7/4+1/2*m],-tan(d*x+c)^2)*(b*tan(d*x+c))^(1/2)*tan(d*x+c)^(1+m)/
d/(3+2*m)+2*B*hypergeom([1, 5/4+1/2*m],[9/4+1/2*m],-tan(d*x+c)^2)*(b*tan(d*
x+c))^(1/2)*tan(d*x+c)^(2+m)/d/(5+2*m)
```

Rubi [A] time = 0.14, antiderivative size = 170, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, integrand size = 41, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.122, Rules used = {20, 3630, 3538, 3476, 364}

$$\frac{2(A-C)\sqrt{b\tan(c+dx)}\,\tan^{m+1}(c+dx)\,{}_2F_1\left(1,\frac{1}{4}(2m+3);\frac{1}{4}(2m+7);-\tan^2(c+dx)\right)}{d(2m+3)}+\frac{2B\sqrt{b\tan(c+dx)}\,\tan^{m+}}{}$$

Antiderivative was successfully verified.

```
[In] Int[Tan[c + d*x]^m*Sqrt[b*Tan[c + d*x]]*(A + B*Tan[c + d*x] + C*Tan[c + d*x
]^2),x]
```

```
[Out] (2*C*Tan[c + d*x]^(1 + m)*Sqrt[b*Tan[c + d*x]])/(d*(3 + 2*m)) + (2*(A - C)*
Hypergeometric2F1[1, (3 + 2*m)/4, (7 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c + d*x
]^(1 + m)*Sqrt[b*Tan[c + d*x]])/(d*(3 + 2*m)) + (2*B*Hypergeometric2F1[1, (
5 + 2*m)/4, (9 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c + d*x]^(2 + m)*Sqrt[b*Tan[c
 + d*x]])/(d*(5 + 2*m))
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 364

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[(a^
p*(c*x)^(m + 1)*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1, -((b*x^n)/a
)])/(c*(m + 1)), x] /; FreeQ[{a, b, c, m, n, p}, x] && !IGtQ[p, 0] && (ILt
Q[p, 0] || GtQ[a, 0])
```

Rule 3476

```
Int[((b_.)*tan[(c_.) + (d_.)*(x_)])^(n_), x_Symbol] :> Dist[b/d, Subst[Int[
x^n/(b^2 + x^2), x], x, b*Tan[c + d*x]], x] /; FreeQ[{b, c, d, n}, x] &&  !
IntegerQ[n]
```

Rule 3538

```
Int[((b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) + (f_.)*(x
_)]), x_Symbol] :> Dist[c, Int[(b*Tan[e + f*x])^m, x], x] + Dist[d/b, Int[(
b*Tan[e + f*x])^(m + 1), x], x] /; FreeQ[{b, c, d, e, f, m}, x] && NeQ[c^2
+ d^2, 0] &&  !IntegerQ[2*m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int \tan^m(c+dx)\sqrt{b\tan(c+dx)}\,\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\,dx = \frac{\sqrt{b\tan(c+dx)}\,\int \tan^{\frac{1}{2}+m}(c+dx)\left(A\right.}{\sqrt{\tan}}$$

$$= \frac{2C\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(3+2m)} +$$

$$= \frac{2C\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(3+2m)} +$$

$$= \frac{2C\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(3+2m)} +$$

$$= \frac{2C\tan^{1+m}(c+dx)\sqrt{b\tan(c+dx)}}{d(3+2m)} +$$

Mathematica [A] time = 0.53, size = 133, normalized size = 0.78

$$\frac{2\sqrt{b\tan(c+dx)}\,\tan^{m+1}(c+dx)\left((2m+5)(A-C)\,{}_2F_1\left(1,\frac{1}{4}(2m+3);\frac{1}{4}(2m+7);-\tan^2(c+dx)\right)+B(2m+3)\tan\right.}{d(2m+3)(2m+5)}$$

Antiderivative was successfully verified.

```
[In] Integrate[Tan[c + d*x]^m*Sqrt[b*Tan[c + d*x]]*(A + B*Tan[c + d*x] + C*Tan[c
 + d*x]^2),x]
```

```
[Out] (2*Tan[c + d*x]^(1 + m)*Sqrt[b*Tan[c + d*x]]*(C*(5 + 2*m) + (A - C)*(5 + 2*
m)*Hypergeometric2F1[1, (3 + 2*m)/4, (7 + 2*m)/4, -Tan[c + d*x]^2] + B*(3 +
 2*m)*Hypergeometric2F1[1, (5 + 2*m)/4, (9 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c
 + d*x]))/(d*(3 + 2*m)*(5 + 2*m))
```

fricas [F] time = 0.55, size = 0, normalized size = 0.00

$$\mathrm{integral}\left(\left(C\tan\left(dx+c\right)^2+B\tan\left(dx+c\right)+A\right)\sqrt{b\tan\left(dx+c\right)}\,\tan\left(dx+c\right)^m,x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)
,x, algorithm="fricas")
```

```
[Out] integral((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*sqrt(b*tan(d*x + c))*tan(d
*x + c)^m, x)
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)
,x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F] time = 1.71, size = 0, normalized size = 0.00

$$\int\left(\tan^m\left(dx+c\right)\right)\sqrt{b\tan\left(dx+c\right)}\,\left(A+B\tan\left(dx+c\right)+C\left(\tan^2\left(dx+c\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)^m*(b*tan(d*x+c))^(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

```
[Out] int(tan(d*x+c)^m*(b*tan(d*x+c))^(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)^2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(b*tan(d*x+c))^(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)
,x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.01

$$\int \tan\left(c+d\,x\right)^m\,\sqrt{b\,\tan\left(c+d\,x\right)}\,\left(C\,{\tan\left(c+d\,x\right)}^2+B\,\tan\left(c+d\,x\right)+A\right) \,d x$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(c + d*x)^m*(b*tan(c + d*x))^(1/2)*(A + B*tan(c + d*x) + C*tan(c + d
*x)^2),x)
```

```
[Out] int(tan(c + d*x)^m*(b*tan(c + d*x))^(1/2)*(A + B*tan(c + d*x) + C*tan(c + d
*x)^2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \sqrt{b \tan{\left(c + dx \right)}} \left(A + B \tan{\left(c + dx \right)} + C \tan^{2}{\left(c + dx \right)}\right) \tan^{m}{\left(c + dx \right)}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)**m*(b*tan(d*x+c))**(1/2)*(A+B*tan(d*x+c)+C*tan(d*x+c)*
*2),x)
```

```
[Out] Integral(sqrt(b*tan(c + d*x))*(A + B*tan(c + d*x) + C*tan(c + d*x)**2)*tan(
c + d*x)**m, x)
```

3.48 $\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx$

Optimal. Leaf size=170

$$\frac{2(A-C)\tan^{m+1}(c+dx)\,{}_2F_1\left(1,\frac{1}{4}(2m+1);\frac{1}{4}(2m+5);-\tan^2(c+dx)\right)}{d(2m+1)\sqrt{b\tan(c+dx)}}+\frac{2B\tan^{m+2}(c+dx)\,{}_2F_1\left(1,\frac{1}{4}(2m+3);\frac{1}{4}(2\right.}{d(2m+3)\sqrt{b\tan(c+}}$$

```
[Out] 2*C*tan(d*x+c)^(1+m)/d/(1+2*m)/(b*tan(d*x+c))^(1/2)+2*(A-C)*hypergeom([1, 1
/4+1/2*m],[5/4+1/2*m],-tan(d*x+c)^2)*tan(d*x+c)^(1+m)/d/(1+2*m)/(b*tan(d*x+
c))^(1/2)+2*B*hypergeom([1, 3/4+1/2*m],[7/4+1/2*m],-tan(d*x+c)^2)*tan(d*x+c
)^(2+m)/d/(3+2*m)/(b*tan(d*x+c))^(1/2)
```

Rubi [A] time = 0.14, antiderivative size = 170, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, integrand size = 41, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.122, Rules used = {20, 3630, 3538, 3476, 364}

$$\frac{2(A-C)\tan^{m+1}(c+dx)\,{}_2F_1\left(1,\frac{1}{4}(2m+1);\frac{1}{4}(2m+5);-\tan^2(c+dx)\right)}{d(2m+1)\sqrt{b\tan(c+dx)}}+\frac{2B\tan^{m+2}(c+dx)\,{}_2F_1\left(1,\frac{1}{4}(2m+3);\frac{1}{4}(2\right.}{d(2m+3)\sqrt{b\tan(c+}}$$

Antiderivative was successfully verified.

```
[In] Int[(Tan[c + d*x]^m*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^2))/Sqrt[b*Tan[c +
 d*x]],x]
```

```
[Out] (2*C*Tan[c + d*x]^(1 + m))/(d*(1 + 2*m)*Sqrt[b*Tan[c + d*x]]) + (2*(A - C)*
Hypergeometric2F1[1, (1 + 2*m)/4, (5 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c + d*x
]^(1 + m))/(d*(1 + 2*m)*Sqrt[b*Tan[c + d*x]]) + (2*B*Hypergeometric2F1[1, (
3 + 2*m)/4, (7 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c + d*x]^(2 + m))/(d*(3 + 2*m
)*Sqrt[b*Tan[c + d*x]])
```

Rule 20

```
Int[(u_.)*((a_.)*(v_))^(m_)*((b_.)*(v_))^(n_), x_Symbol] :> Dist[(b^IntPart
[n]*(b*v)^FracPart[n])/(a^IntPart[n]*(a*v)^FracPart[n]), Int[u*(a*v)^(m + n
), x], x] /; FreeQ[{a, b, m, n}, x] && !IntegerQ[m] && !IntegerQ[n] && !
IntegerQ[m + n]
```

Rule 364

```
Int[((c_.)*(x_))^(m_.)*((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[(a^
p*(c*x)^(m + 1)*Hypergeometric2F1[-p, (m + 1)/n, (m + 1)/n + 1, -((b*x^n)/a
)])/(c*(m + 1)), x] /; FreeQ[{a, b, c, m, n, p}, x] && !IGtQ[p, 0] && (ILt
```

```
Q[p, 0] || GtQ[a, 0])
```

Rule 3476

```
Int[((b_.)*tan[(c_.) + (d_.)*(x_)])^(n_), x_Symbol] :> Dist[b/d, Subst[Int[
x^n/(b^2 + x^2), x], x, b*Tan[c + d*x]], x] /; FreeQ[{b, c, d, n}, x] &&  !
IntegerQ[n]
```

Rule 3538

```
Int[((b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) + (f_.)*(x
_)]), x_Symbol] :> Dist[c, Int[(b*Tan[e + f*x])^m, x], x] + Dist[d/b, Int[(
b*Tan[e + f*x])^(m + 1), x], x] /; FreeQ[{b, c, d, e, f, m}, x] && NeQ[c^2
+ d^2, 0] &&  !IntegerQ[2*m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{b\tan(c+dx)}}\,dx = \frac{\sqrt{\tan(c+dx)}\int \tan^{-\frac{1}{2}+m}(c+dx)\left(A+B\tan(c+d\right.}{\sqrt{b\tan(c+dx)}}$$

$$= \frac{2C\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}} + \frac{\sqrt{\tan(c+dx)}\int \tan^{-\frac{1}{2}}}{}$$

$$= \frac{2C\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}} + \frac{\left(B\sqrt{\tan(c+dx)}\right)\int \tan}{\sqrt{b\tan(c+}}$$

$$= \frac{2C\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}} + \frac{\left(B\sqrt{\tan(c+dx)}\right)\mathrm{Subs}}{d\sqrt{b}}$$

$$= \frac{2C\tan^{1+m}(c+dx)}{d(1+2m)\sqrt{b\tan(c+dx)}} + \frac{2(A-C)\,{}_2F_1\left(1,\frac{1}{4}(1+\right.}{}$$

Mathematica [A] time = 0.54, size = 133, normalized size = 0.78

$$\frac{2\tan^{m+1}(c+dx)\left((2m+3)(A-C)\,{}_2F_1\left(1,\frac{1}{4}(2m+1);\frac{1}{4}(2m+5);-\tan^2(c+dx)\right)+B(2m+1)\tan(c+dx)\,{}_2F_1\left(1,\right.\right.}{d(2m+1)(2m+3)\sqrt{b\tan(c+dx)}}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Tan[c + d*x]^m*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^2))/Sqrt[b*T
an[c + d*x]],x]
```

```
[Out] (2*Tan[c + d*x]^(1 + m)*(C*(3 + 2*m) + (A - C)*(3 + 2*m)*Hypergeometric2F1[
1, (1 + 2*m)/4, (5 + 2*m)/4, -Tan[c + d*x]^2] + B*(1 + 2*m)*Hypergeometric2
F1[1, (3 + 2*m)/4, (7 + 2*m)/4, -Tan[c + d*x]^2]*Tan[c + d*x]))/(d*(1 + 2*m
)*(3 + 2*m)*Sqrt[b*Tan[c + d*x]])
```

fricas [F] time = 0.66, size = 0, normalized size = 0.00

$$\operatorname{integral}\left(\frac{\left(C\tan\left(dx+c\right)^2+B\tan\left(dx+c\right)+A\right)\sqrt{b\tan\left(dx+c\right)}\,\tan\left(dx+c\right)^m}{b\tan\left(dx+c\right)},x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(b*tan(d*x+c))^(1/2)
,x, algorithm="fricas")
```

```
[Out] integral((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*sqrt(b*tan(d*x + c))*tan(d
*x + c)^m/(b*tan(d*x + c)), x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\frac{\left(C\tan\left(dx+c\right)^2+B\tan\left(dx+c\right)+A\right)\tan\left(dx+c\right)^m}{\sqrt{b\tan\left(dx+c\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(b*tan(d*x+c))^(1/2)
,x, algorithm="giac")
```

```
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*tan(d*x + c)^m/sqrt(b*tan
(d*x + c)), x)
```

maple [F] time = 1.76, size = 0, normalized size = 0.00

$$\int\frac{\left(\tan^m\left(dx+c\right)\right)\left(A+B\tan\left(dx+c\right)+C\left(\tan^2\left(dx+c\right)\right)\right)}{\sqrt{b\tan\left(dx+c\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(b*tan(d*x+c))^(1/2),x)
```

```
[Out] int(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(b*tan(d*x+c))^(1/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(b*tan(d*x+c))^(1/2)
,x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.01

$$\int \frac{\tan\left(c+d\,x\right)^m\,\left(C\,\tan\left(c+d\,x\right)^2+B\,\tan\left(c+d\,x\right)+A\right)}{\sqrt{b\,\tan\left(c+d\,x\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((tan(c + d*x)^m*(A + B*tan(c + d*x) + C*tan(c + d*x)^2))/(b*tan(c + d*x
))^(1/2),x)
```

```
[Out] int((tan(c + d*x)^m*(A + B*tan(c + d*x) + C*tan(c + d*x)^2))/(b*tan(c + d*x
))^(1/2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(A + B \tan{\left(c + dx \right)} + C \tan^{2}{\left(c + dx \right)}\right) \tan^{m}{\left(c + dx \right)}}{\sqrt{b \tan{\left(c + dx \right)}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)**m*(A+B*tan(d*x+c)+C*tan(d*x+c)**2)/(b*tan(d*x+c))**(1
/2),x)
```

```
[Out] Integral((A + B*tan(c + d*x) + C*tan(c + d*x)**2)*tan(c + d*x)**m/sqrt(b*ta
n(c + d*x)), x)
```

3.49 $\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx$

Optimal. Leaf size=328

$$-\frac{\left(\sqrt{-b^2}\,(A-C)+bB\right)\tan^m(c+dx)\sqrt{a+b\tan(c+dx)}\,\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}F_1\left(\frac{1}{2};1,-m;\frac{3}{2};\frac{a+b\tan(c+dx)}{a-\sqrt{-b^2}},\frac{b\tan(c+dx)}{a}+1\right.}{bd\left(a-\sqrt{-b^2}\right)}$$

```
[Out] 2*C*hypergeom([1/2, -m],[3/2],1+b*tan(d*x+c)/a)*(a+b*tan(d*x+c))^(1/2)*tan(
d*x+c)^m/b/d/((-b*tan(d*x+c)/a)^m)-AppellF1(1/2,1,-m,3/2,(a+b*tan(d*x+c))/(
a+(-b^2)^(1/2)),1+b*tan(d*x+c)/a)*(b*B-(A-C)*(-b^2)^(1/2))*(a+b*tan(d*x+c))
^(1/2)*tan(d*x+c)^m/b/d/(a+(-b^2)^(1/2))/((-b*tan(d*x+c)/a)^m)-AppellF1(1/2
,1,-m,3/2,(a+b*tan(d*x+c))/(a-(-b^2)^(1/2)),1+b*tan(d*x+c)/a)*(b*B+(A-C)*(-
b^2)^(1/2))*(a+b*tan(d*x+c))^(1/2)*tan(d*x+c)^m/b/d/(a-(-b^2)^(1/2))/((-b*t
an(d*x+c)/a)^m)
```

Rubi [A] time = 1.56, antiderivative size = 328, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 7, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3655, 6720, 1692, 246, 245, 430, 429}

$$-\frac{\left(\sqrt{-b^2}\,(A-C)+bB\right)\tan^m(c+dx)\sqrt{a+b\tan(c+dx)}\,\left(-\frac{b\tan(c+dx)}{a}\right)^{-m}F_1\left(\frac{1}{2};1,-m;\frac{3}{2};\frac{a+b\tan(c+dx)}{a-\sqrt{-b^2}},\frac{b\tan(c+dx)}{a}+1\right.}{bd\left(a-\sqrt{-b^2}\right)}$$

Antiderivative was successfully verified.

```
[In] Int[(Tan[c + d*x]^m*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^2))/Sqrt[a + b*Tan
[c + d*x]],x]
```

```
[Out] -(((b*B + Sqrt[-b^2]*(A - C))*AppellF1[1/2, 1, -m, 3/2, (a + b*Tan[c + d*x]
)/(a - Sqrt[-b^2]), 1 + (b*Tan[c + d*x])/a]*Tan[c + d*x]^m*Sqrt[a + b*Tan[c
 + d*x]])/(b*(a - Sqrt[-b^2])*d*(-((b*Tan[c + d*x])/a))^m)) - ((b*B - Sqrt[
-b^2]*(A - C))*AppellF1[1/2, 1, -m, 3/2, (a + b*Tan[c + d*x])/(a + Sqrt[-b^
2]), 1 + (b*Tan[c + d*x])/a]*Tan[c + d*x]^m*Sqrt[a + b*Tan[c + d*x]])/(b*(a
 + Sqrt[-b^2])*d*(-((b*Tan[c + d*x])/a))^m) + (2*C*Hypergeometric2F1[1/2, -
m, 3/2, 1 + (b*Tan[c + d*x])/a]*Tan[c + d*x]^m*Sqrt[a + b*Tan[c + d*x]])/(b
*d*(-((b*Tan[c + d*x])/a))^m)
```

Rule 245

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Simp[a^p*x*Hypergeometric2F
1[-p, 1/n, 1/n + 1, -((b*x^n)/a)], x] /; FreeQ[{a, b, n, p}, x] && !IGtQ[p
, 0] && !IntegerQ[1/n] && !ILtQ[Simplify[1/n + p], 0] && (IntegerQ[p] ||
```

```
GtQ[a, 0])
```

Rule 246

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_), x_Symbol] :> Dist[(a^IntPart[p]*(a + b*x
^n)^FracPart[p])/(1 + (b*x^n)/a)^FracPart[p], Int[(1 + (b*x^n)/a)^p, x], x]
 /; FreeQ[{a, b, n, p}, x] &&  !IGtQ[p, 0] &&  !IntegerQ[1/n] &&  !ILtQ[Sim
plify[1/n + p], 0] &&  !(IntegerQ[p] || GtQ[a, 0])
```

Rule 429

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_)*((c_) + (d_.)*(x_)^(n_))^(q_), x_Symbol]
:> Simp[a^p*c^q*x*AppellF1[1/n, -p, -q, 1 + 1/n, -((b*x^n)/a), -((d*x^n)/c)
], x] /; FreeQ[{a, b, c, d, n, p, q}, x] && NeQ[b*c - a*d, 0] && NeQ[n, -1]
 && (IntegerQ[p] || GtQ[a, 0]) && (IntegerQ[q] || GtQ[c, 0])
```

Rule 430

```
Int[((a_) + (b_.)*(x_)^(n_))^(p_)*((c_) + (d_.)*(x_)^(n_))^(q_), x_Symbol]
:> Dist[(a^IntPart[p]*(a + b*x^n)^FracPart[p])/(1 + (b*x^n)/a)^FracPart[p],
 Int[(1 + (b*x^n)/a)^p*(c + d*x^n)^q, x], x] /; FreeQ[{a, b, c, d, n, p, q}
, x] && NeQ[b*c - a*d, 0] && NeQ[n, -1] &&  !(IntegerQ[p] || GtQ[a, 0])
```

Rule 1692

```
Int[(Px_)*((d_) + (e_.)*(x_)^2)^(q_.)*((a_) + (b_.)*(x_)^2 + (c_.)*(x_)^4)^
(p_.), x_Symbol] :> Int[ExpandIntegrand[Px*(d + e*x^2)^q*(a + b*x^2 + c*x^4
)^p, x], x] /; FreeQ[{a, b, c, d, e, q}, x] && PolyQ[Px, x^2] && NeQ[b^2 -
4*a*c, 0] && NeQ[c*d^2 - b*d*e + a*e^2, 0] && IntegerQ[p]
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6720

```
Int[(u_.)*((a_.)*(v_)^(m_.))^(p_), x_Symbol] :> Dist[(a^IntPart[p]*(a*v^m)^
FracPart[p])/v^(m*FracPart[p]), Int[u*v^(m*p), x], x] /; FreeQ[{a, m, p}, x
] &&  !IntegerQ[p] &&  !FreeQ[v, x] &&  !(EqQ[a, 1] && EqQ[m, 1]) &&  !(EqQ
```

```
[v, x] && EqQ[m, 1])
```

Rubi steps

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx = \frac{\text{Subst}\left(\int \frac{x^m\left(A+Bx+Cx^2\right)}{\sqrt{a+bx}\left(1+x^2\right)}\,dx, x, \tan(c+dx)\right)}{d}$$

$$= \frac{2\,\text{Subst}\left(\int \frac{\left(\frac{-a+x^2}{b}\right)^m\left(Ab^2+\left(a-x^2\right)\left(-bB+C\left(a-x^2\right)\right)\right)}{a^2+b^2-2ax^2+x^4}\,dx, x, \sqrt{a}\right.}{bd}$$

$$= \frac{\left(2\tan^m(c+dx)(b\tan(c+dx))^{-m}\right)\text{Subst}\left(\int \frac{\left(-a+x^2\right)^m(}{}\right.}{bd}$$

$$= \frac{\left(2\tan^m(c+dx)(b\tan(c+dx))^{-m}\right)\text{Subst}\left(\int\left(C\left(-a+\right.\right.\right.}{}$$

$$= \frac{\left(2\tan^m(c+dx)(b\tan(c+dx))^{-m}\right)\text{Subst}\left(\int \frac{\left(-a+x^2\right)^m(}{a^2-}\right.}{bd}$$

$$= \frac{\left(2\tan^m(c+dx)(b\tan(c+dx))^{-m}\right)\text{Subst}\left(\int\left(\frac{\left(bB-\sqrt{-b}\right.}{-2a}\right.\right.}{}$$

$$= \frac{2C\,{}_2F_1\left(\frac{1}{2},-m;\frac{3}{2};\frac{a+b\tan(c+dx)}{a}\right)\tan^m(c+dx)\left(-\frac{b\tan(c+}{a}\right.}{bd}$$

$$= \frac{2C\,{}_2F_1\left(\frac{1}{2},-m;\frac{3}{2};\frac{a+b\tan(c+dx)}{a}\right)\tan^m(c+dx)\left(-\frac{b\tan(c+}{a}\right.}{bd}$$

$$-\frac{\left(bB+\sqrt{-b^2}\,(A-C)\right)F_1\left(\frac{1}{2};1,-m;\frac{3}{2};\frac{a+b\tan(c+dx)}{a-\sqrt{-b^2}},\right.}{b\left(\right.}$$

Mathematica [F] time = 28.56, size = 0, normalized size = 0.00

$$\int \frac{\tan^m(c+dx)\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)}{\sqrt{a+b\tan(c+dx)}}\,dx$$

Verification is Not applicable to the result.

```
[In] Integrate[(Tan[c + d*x]^m*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^2))/Sqrt[a +
 b*Tan[c + d*x]],x]
```

```
[Out] Integrate[(Tan[c + d*x]^m*(A + B*Tan[c + d*x] + C*Tan[c + d*x]^2))/Sqrt[a +
 b*Tan[c + d*x]], x]
```

fricas [F] time = 0.82, size = 0, normalized size = 0.00

$$\text{integral}\left(\frac{\left(C\tan\left(dx+c\right)^{2}+B\tan\left(dx+c\right)+A\right)\tan\left(dx+c\right)^{m}}{\sqrt{b\tan\left(dx+c\right)+a}},x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^(1/
2),x, algorithm="fricas")
```

```
[Out] integral((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*tan(d*x + c)^m/sqrt(b*tan(
d*x + c) + a), x)
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^(1/
2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F] time = 2.04, size = 0, normalized size = 0.00

$$\int\frac{\left(\tan^{m}\left(dx+c\right)\right)\left(A+B\tan\left(dx+c\right)+C\left(\tan^{2}\left(dx+c\right)\right)\right)}{\sqrt{a+b\tan\left(dx+c\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^(1/2),x)
```

```
[Out] int(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^(1/2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\frac{\left(C\tan\left(dx+c\right)^{2}+B\tan\left(dx+c\right)+A\right)\tan\left(dx+c\right)^{m}}{\sqrt{b\tan\left(dx+c\right)+a}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)^m*(A+B*tan(d*x+c)+C*tan(d*x+c)^2)/(a+b*tan(d*x+c))^(1/
2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(d*x + c)^2 + B*tan(d*x + c) + A)*tan(d*x + c)^m/sqrt(b*tan
(d*x + c) + a), x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{\tan(c+d\,x)^m\,\left(C\tan(c+d\,x)^2+B\tan(c+d\,x)+A\right)}{\sqrt{a+b\tan(c+d\,x)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((tan(c + d*x)^m*(A + B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c +
 d*x))^(1/2),x)
```

```
[Out] int((tan(c + d*x)^m*(A + B*tan(c + d*x) + C*tan(c + d*x)^2))/(a + b*tan(c +
 d*x))^(1/2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(A+B\tan(c+dx)+C\tan^2(c+dx)\right)\tan^m(c+dx)}{\sqrt{a+b\tan(c+dx)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate(tan(d*x+c)**m*(A+B*tan(d*x+c)+C*tan(d*x+c)**2)/(a+b*tan(d*x+c))**
(1/2),x)
```

```
[Out] Integral((A + B*tan(c + d*x) + C*tan(c + d*x)**2)*tan(c + d*x)**m/sqrt(a +
b*tan(c + d*x)), x)
```

3.50 $\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))\left(A+B\tan(e+fx)\right.$

Optimal. Leaf size=353

$$\frac{b\tan(e+fx)\left(a^2(d(A-C)+Bc)+2ab(Ac-Bd-cC)-b^2(d(A-C)+Bc)\right)}{f}-\frac{\log(\cos(e+fx))\left(a^3(d(A-C)+\right.}{}$$

```
[Out] (a^3*(A*c-B*d-C*c)-3*a*b^2*(A*c-B*d-C*c)-3*a^2*b*(B*c+(A-C)*d)+b^3*(B*c+(A-
C)*d))*x-(3*a^2*b*(A*c-B*d-C*c)-b^3*(A*c-B*d-C*c)+a^3*(B*c+(A-C)*d)-3*a*b^2
*(B*c+(A-C)*d))*ln(cos(f*x+e))/f+b*(2*a*b*(A*c-B*d-C*c)+a^2*(B*c+(A-C)*d)-b
^2*(B*c+(A-C)*d))*tan(f*x+e)/f+1/2*(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b*c)*(a
+b*tan(f*x+e))^2/f+1/3*(B*c+(A-C)*d)*(a+b*tan(f*x+e))^3/f-1/20*(a*C*d-5*b*(
B*d+C*c))*(a+b*tan(f*x+e))^4/b^2/f+1/5*C*d*tan(f*x+e)*(a+b*tan(f*x+e))^4/b/
f
```

Rubi [A] time = 0.79, antiderivative size = 353, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 5, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3637, 3630, 3528, 3525, 3475}

$$\frac{b\tan(e+fx)\left(a^2(d(A-C)+Bc)+2ab(Ac-Bd-cC)-b^2(d(A-C)+Bc)\right)}{f}-\frac{\log(\cos(e+fx))\left(3a^2b(Ac-Bd\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan
[e + f*x]^2),x]
```

```
[Out] (a^3*(A*c - c*C - B*d) - 3*a*b^2*(A*c - c*C - B*d) - 3*a^2*b*(B*c + (A - C)
*d) + b^3*(B*c + (A - C)*d))*x - ((3*a^2*b*(A*c - c*C - B*d) - b^3*(A*c - c
*C - B*d) + a^3*(B*c + (A - C)*d) - 3*a*b^2*(B*c + (A - C)*d))*Log[Cos[e +
f*x]])/f + (b*(2*a*b*(A*c - c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c +
 (A - C)*d))*Tan[e + f*x])/f + ((A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*
C*d)*(a + b*Tan[e + f*x])^2)/(2*f) + ((B*c + (A - C)*d)*(a + b*Tan[e + f*x]
)^3)/(3*f) - ((a*C*d - 5*b*(c*C + B*d))*(a + b*Tan[e + f*x])^4)/(20*b^2*f)
+ (C*d*Tan[e + f*x]*(a + b*Tan[e + f*x])^4)/(5*b*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
```

```
f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx)) \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx = \frac{Cd\tan(e+fx)(a+b\tan(}{5bf}$$

$$= -\frac{(aCd-5b(cC+Bd))(a-}{20b^2f}$$

$$= \frac{(Bc+(A-C)d)(a+b\tan}{3f}$$

$$= \frac{(Abc+aBc-bcC+aAd}{}$$

$$= \left(a^3(Ac-cC-Bd)-3ab^2\right.$$

$$= \left(a^3(Ac-cC-Bd)-3ab^2\right.$$

Mathematica [C] time = 6.38, size = 300, normalized size = 0.85

$$\frac{Cd\tan(e+fx)(a+b\tan(e+fx))^4}{5bf} - \frac{\frac{(aCd-5b(Bd+cC))(a+b\tan(e+fx))^4}{4bf} - \frac{5\left(3(-aAd-aBc+aCd+Abc-bBd-bcC)\left(6ab^2\tan(e+fx)+(-b-\right.\right.}{}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] (C*d*Tan[e + f*x]*(a + b*Tan[e + f*x])^4)/(5*b*f) - (((a*C*d - 5*b*(c*C + B
*d))*(a + b*Tan[e + f*x])^4)/(4*b*f) - (5*(3*(A*b*c - a*B*c - b*c*C - a*A*d
 - b*B*d + a*C*d)*((I*a - b)^3*Log[I - Tan[e + f*x]] - (I*a + b)^3*Log[I +
Tan[e + f*x]] + 6*a*b^2*Tan[e + f*x] + b^3*Tan[e + f*x]^2) - (B*c + (A - C)
*d)*((3*I)*(a + I*b)^4*Log[I - Tan[e + f*x]] - (3*I)*(a - I*b)^4*Log[I + Ta
n[e + f*x]] - 6*b^2*(6*a^2 - b^2)*Tan[e + f*x] - 12*a*b^3*Tan[e + f*x]^2 -
2*b^4*Tan[e + f*x]^3)))/(6*f))/(5*b)
```

fricas [A] time = 1.04, size = 415, normalized size = 1.18

$$\frac{12\,Cb^3d\tan\left(fx+e\right)^5+15\left(Cb^3c+\left(3\,Cab^2+Bb^3\right)d\right)\tan\left(fx+e\right)^4+20\left(\left(3\,Cab^2+Bb^3\right)c+\left(3\,Ca^2b+3\,Bab^2-\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="fricas")
```

```
[Out] 1/60*(12*C*b^3*d*tan(f*x + e)^5 + 15*(C*b^3*c + (3*C*a*b^2 + B*b^3)*d)*tan(
f*x + e)^4 + 20*((3*C*a*b^2 + B*b^3)*c + (3*C*a^2*b + 3*B*a*b^2 + (A - C)*b
^3)*d)*tan(f*x + e)^3 + 60*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*
b^3)*c - (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d)*f*x + 30*((
3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c + (C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b
^2 - B*b^3)*d)*tan(f*x + e)^2 - 30*((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 -
(A - C)*b^3)*c + ((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*d)*log
(1/(tan(f*x + e)^2 + 1)) + 60*((C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3
)*c + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d)*tan(f*x + e))/
f
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.03, size = 994, normalized size = 2.82

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] 1/2/f*ln(1+tan(f*x+e)^2)*A*a^3*d-1/2/f*ln(1+tan(f*x+e)^2)*A*b^3*c+1/4/f*B*t
an(f*x+e)^4*b^3*d-1/2/f*B*tan(f*x+e)^2*b^3*d+1/2/f*C*tan(f*x+e)^2*a^3*d-1/3
/f*C*tan(f*x+e)^3*b^3*d+1/2/f*ln(1+tan(f*x+e)^2)*B*a^3*c+1/2/f*ln(1+tan(f*x
+e)^2)*B*b^3*d-1/2/f*ln(1+tan(f*x+e)^2)*a^3*C*d+1/3/f*A*tan(f*x+e)^3*b^3*d+
1/3/f*B*tan(f*x+e)^3*b^3*c-3/2/f*ln(1+tan(f*x+e)^2)*A*a*b^2*d+1/f*B*a^3*d*t
an(f*x+e)+1/f*B*tan(f*x+e)^3*a*b^2*d-3/2/f*ln(1+tan(f*x+e)^2)*C*a^2*b*c+3/f
*C*arctan(tan(f*x+e))*a*b^2*c-3/2/f*ln(1+tan(f*x+e)^2)*B*a^2*b*d-3/2/f*ln(1
+tan(f*x+e)^2)*B*a*b^2*c-3/f*A*arctan(tan(f*x+e))*a^2*b*d+1/f*C*b^3*d*tan(f
*x+e)-1/2/f*C*tan(f*x+e)^2*b^3*c+1/2/f*A*tan(f*x+e)^2*b^3*c+1/4/f*C*tan(f*x
+e)^4*b^3*c-1/f*C*arctan(tan(f*x+e))*b^3*d-1/f*B*arctan(tan(f*x+e))*a^3*d+1
/f*B*arctan(tan(f*x+e))*b^3*c-1/f*C*arctan(tan(f*x+e))*a^3*c-1/f*B*b^3*c*ta
n(f*x+e)+1/f*C*a^3*c*tan(f*x+e)-1/f*A*b^3*d*tan(f*x+e)+1/2/f*ln(1+tan(f*x+e
)^2)*C*b^3*c+1/f*A*arctan(tan(f*x+e))*a^3*c+1/f*A*arctan(tan(f*x+e))*b^3*d+
1/5/f*C*b^3*d*tan(f*x+e)^5-3/f*A*arctan(tan(f*x+e))*a*b^2*c-3/f*B*arctan(ta
```

```
n(f*x+e))*a^2*b*c+3/2/f*A*tan(f*x+e)^2*a*b^2*d+3/f*B*arctan(tan(f*x+e))*a*b
^2*d+1/f*C*tan(f*x+e)^3*a^2*b*d+3/f*C*arctan(tan(f*x+e))*a^2*b*d-3/f*C*a^2*
b*d*tan(f*x+e)+1/f*C*tan(f*x+e)^3*a*b^2*c+3/2/f*B*tan(f*x+e)^2*a*b^2*c+3/f*
B*a^2*b*c*tan(f*x+e)+3/2/f*B*tan(f*x+e)^2*a^2*b*d+3/2/f*ln(1+tan(f*x+e)^2)*
A*a^2*b*c-3/f*C*a*b^2*c*tan(f*x+e)+3/f*A*a^2*b*d*tan(f*x+e)+3/4/f*C*tan(f*x
+e)^4*a*b^2*d+3/f*A*a*b^2*c*tan(f*x+e)-3/f*B*a*b^2*d*tan(f*x+e)-3/2/f*C*tan
(f*x+e)^2*a*b^2*d+3/2/f*ln(1+tan(f*x+e)^2)*C*a*b^2*d+3/2/f*C*tan(f*x+e)^2*a
^2*b*c
```

maxima [A] time = 0.54, size = 416, normalized size = 1.18

$$\frac{12\,Cb^3d\tan\left(fx+e\right)^5+15\left(Cb^3c+\left(3\,Cab^2+Bb^3\right)d\right)\tan\left(fx+e\right)^4+20\left(\left(3\,Cab^2+Bb^3\right)c+\left(3\,Ca^2b+3\,Bab^2\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="maxima")
```

```
[Out] 1/60*(12*C*b^3*d*tan(f*x + e)^5 + 15*(C*b^3*c + (3*C*a*b^2 + B*b^3)*d)*tan(
f*x + e)^4 + 20*((3*C*a*b^2 + B*b^3)*c + (3*C*a^2*b + 3*B*a*b^2 + (A - C)*b
^3)*d)*tan(f*x + e)^3 + 30*((3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c + (C*a^
3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*d)*tan(f*x + e)^2 + 60*(((A - C)*a
^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c - (B*a^3 + 3*(A - C)*a^2*b - 3*
B*a*b^2 - (A - C)*b^3)*d)*(f*x + e) + 30*((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*
b^2 - (A - C)*b^3)*c + ((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*
d)*log(tan(f*x + e)^2 + 1) + 60*((C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b
^3)*c + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d)*tan(f*x + e)
)/f
```

mupad [B] time = 9.00, size = 477, normalized size = 1.35

$$x\left(A\,a^3\,c+A\,b^3\,d-B\,a^3\,d+B\,b^3\,c-C\,a^3\,c-C\,b^3\,d-3\,A\,a\,b^2\,c-3\,A\,a^2\,b\,d-3\,B\,a^2\,b\,c+3\,B\,a\,b^2\,d+3\,C\,a\,b^2\,c\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^3*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2),x)
```

```
[Out] x*(A*a^3*c + A*b^3*d - B*a^3*d + B*b^3*c - C*a^3*c - C*b^3*d - 3*A*a*b^2*c
- 3*A*a^2*b*d - 3*B*a^2*b*c + 3*B*a*b^2*d + 3*C*a*b^2*c + 3*C*a^2*b*d) + (t
an(e + f*x)^4*((B*b^3*d)/4 + (C*b^3*c)/4 + (3*C*a*b^2*d)/4))/f + (tan(e + f
*x)^3*((A*b^3*d)/3 + (B*b^3*c)/3 - (C*b^3*d)/3 + B*a*b^2*d + C*a*b^2*c + C*
a^2*b*d))/f + (tan(e + f*x)^2*((A*b^3*c)/2 - (B*b^3*d)/2 + (C*a^3*d)/2 - (C
```

```
*b^3*c)/2 + (3*A*a*b^2*d)/2 + (3*B*a*b^2*c)/2 + (3*B*a^2*b*d)/2 + (3*C*a^2*
b*c)/2 - (3*C*a*b^2*d)/2))/f + (log(tan(e + f*x)^2 + 1)*((A*a^3*d)/2 - (A*b
^3*c)/2 + (B*a^3*c)/2 + (B*b^3*d)/2 - (C*a^3*d)/2 + (C*b^3*c)/2 + (3*A*a^2*
b*c)/2 - (3*A*a*b^2*d)/2 - (3*B*a*b^2*c)/2 - (3*B*a^2*b*d)/2 - (3*C*a^2*b*c
)/2 + (3*C*a*b^2*d)/2))/f + (tan(e + f*x)*(B*a^3*d - A*b^3*d - B*b^3*c + C*
a^3*c + C*b^3*d + 3*A*a*b^2*c + 3*A*a^2*b*d + 3*B*a^2*b*c - 3*B*a*b^2*d - 3
*C*a*b^2*c - 3*C*a^2*b*d))/f + (C*b^3*d*tan(e + f*x)^5)/(5*f)
```

sympy [A] time = 1.65, size = 1001, normalized size = 2.84

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**3*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)
**2),x)
```

```
[Out] Piecewise((A*a**3*c*x + A*a**3*d*log(tan(e + f*x)**2 + 1)/(2*f) + 3*A*a**2*
b*c*log(tan(e + f*x)**2 + 1)/(2*f) - 3*A*a**2*b*d*x + 3*A*a**2*b*d*tan(e +
f*x)/f - 3*A*a*b**2*c*x + 3*A*a*b**2*c*tan(e + f*x)/f - 3*A*a*b**2*d*log(ta
n(e + f*x)**2 + 1)/(2*f) + 3*A*a*b**2*d*tan(e + f*x)**2/(2*f) - A*b**3*c*lo
g(tan(e + f*x)**2 + 1)/(2*f) + A*b**3*c*tan(e + f*x)**2/(2*f) + A*b**3*d*x
+ A*b**3*d*tan(e + f*x)**3/(3*f) - A*b**3*d*tan(e + f*x)/f + B*a**3*c*log(t
an(e + f*x)**2 + 1)/(2*f) - B*a**3*d*x + B*a**3*d*tan(e + f*x)/f - 3*B*a**2
*b*c*x + 3*B*a**2*b*c*tan(e + f*x)/f - 3*B*a**2*b*d*log(tan(e + f*x)**2 + 1
)/(2*f) + 3*B*a**2*b*d*tan(e + f*x)**2/(2*f) - 3*B*a*b**2*c*log(tan(e + f*x
)**2 + 1)/(2*f) + 3*B*a*b**2*c*tan(e + f*x)**2/(2*f) + 3*B*a*b**2*d*x + B*a
*b**2*d*tan(e + f*x)**3/f - 3*B*a*b**2*d*tan(e + f*x)/f + B*b**3*c*x + B*b*
*3*c*tan(e + f*x)**3/(3*f) - B*b**3*c*tan(e + f*x)/f + B*b**3*d*log(tan(e +
 f*x)**2 + 1)/(2*f) + B*b**3*d*tan(e + f*x)**4/(4*f) - B*b**3*d*tan(e + f*x
)**2/(2*f) - C*a**3*c*x + C*a**3*c*tan(e + f*x)/f - C*a**3*d*log(tan(e + f*
x)**2 + 1)/(2*f) + C*a**3*d*tan(e + f*x)**2/(2*f) - 3*C*a**2*b*c*log(tan(e
+ f*x)**2 + 1)/(2*f) + 3*C*a**2*b*c*tan(e + f*x)**2/(2*f) + 3*C*a**2*b*d*x
+ C*a**2*b*d*tan(e + f*x)**3/f - 3*C*a**2*b*d*tan(e + f*x)/f + 3*C*a*b**2*c
*x + C*a*b**2*c*tan(e + f*x)**3/f - 3*C*a*b**2*c*tan(e + f*x)/f + 3*C*a*b**
2*d*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*a*b**2*d*tan(e + f*x)**4/(4*f) - 3
*C*a*b**2*d*tan(e + f*x)**2/(2*f) + C*b**3*c*log(tan(e + f*x)**2 + 1)/(2*f)
 + C*b**3*c*tan(e + f*x)**4/(4*f) - C*b**3*c*tan(e + f*x)**2/(2*f) - C*b**3
*d*x + C*b**3*d*tan(e + f*x)**5/(5*f) - C*b**3*d*tan(e + f*x)**3/(3*f) + C*
b**3*d*tan(e + f*x)/f, Ne(f, 0)), (x*(a + b*tan(e))**3*(c + d*tan(e))*(A +
B*tan(e) + C*tan(e)**2), True))
```

3.51 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+fx)\right.$

Optimal. Leaf size=248

$$-\frac{\log(\cos(e+fx))\left(a^2(d(A-C)+Bc)+2ab(Ac-Bd-cC)-b^2(d(A-C)+Bc)\right)}{f}+x\left(a^2(Ac-Bd-cC)-2ab(\right.$$

```
[Out] (a^2*(A*c-B*d-C*c)-b^2*(A*c-B*d-C*c)-2*a*b*(B*c+(A-C)*d))*x-(2*a*b*(A*c-B*d
-C*c)+a^2*(B*c+(A-C)*d)-b^2*(B*c+(A-C)*d))*ln(cos(f*x+e))/f+b*(A*a*d+A*b*c+
B*a*c-B*b*d-C*a*d-C*b*c)*tan(f*x+e)/f+1/2*(B*c+(A-C)*d)*(a+b*tan(f*x+e))^2/
f-1/12*(a*C*d-4*b*(B*d+C*c))*(a+b*tan(f*x+e))^3/b^2/f+1/4*C*d*tan(f*x+e)*(a
+b*tan(f*x+e))^3/b/f
```

Rubi [A] time = 0.45, antiderivative size = 248, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3637, 3630, 3528, 3525, 3475}

$$-\frac{\log(\cos(e+fx))\left(a^2(d(A-C)+Bc)+2ab(Ac-Bd-cC)-b^2(d(A-C)+Bc)\right)}{f}+x\left(a^2(Ac-Bd-cC)-2ab(\right.$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan
[e + f*x]^2),x]
```

```
[Out] (a^2*(A*c - c*C - B*d) - b^2*(A*c - c*C - B*d) - 2*a*b*(B*c + (A - C)*d))*x
 - ((2*a*b*(A*c - c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A - C)*d
))*Log[Cos[e + f*x]])/f + (b*(A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d
)*Tan[e + f*x])/f + ((B*c + (A - C)*d)*(a + b*Tan[e + f*x])^2)/(2*f) - ((a*
C*d - 4*b*(c*C + B*d))*(a + b*Tan[e + f*x])^3)/(12*b^2*f) + (C*d*Tan[e + f*
x]*(a + b*Tan[e + f*x])^3)/(4*b*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\frac{Cd\tan(e+fx)(a+b\tan(e}{4bf}$$

$$=-\frac{(aCd-4b(cC+Bd))(a+}{12b^2f}$$

$$=\frac{(Bc+(A-C)d)(a+b\tan(}{2f}$$

$$=\left(a^2(Ac-cC-Bd)-b^2(Ac\right.$$

$$=\left(a^2(Ac-cC-Bd)-b^2(Ac\right.$$

Mathematica [C] time = 3.43, size = 243, normalized size = 0.98

$$\frac{-6(-aAd-aBc+aCd+Abc-bBd-bcC)\left(-2b^2\tan(e+fx)+i\left((a+ib)^2\log(-\tan(e+fx)+i)-(a-ib)^2\log\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] (((-(a*C*d) + 4*b*(c*C + B*d))*(a + b*Tan[e + f*x])^3)/b + 3*C*d*Tan[e + f*
x]*(a + b*Tan[e + f*x])^3 - 6*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*
d)*(I*((a + I*b)^2*Log[I - Tan[e + f*x]] - (a - I*b)^2*Log[I + Tan[e + f*x]
]) - 2*b^2*Tan[e + f*x]) + 6*(B*c + (A - C)*d)*((I*a - b)^3*Log[I - Tan[e +
 f*x]] - (I*a + b)^3*Log[I + Tan[e + f*x]] + 6*a*b^2*Tan[e + f*x] + b^3*Tan
[e + f*x]^2))/(12*b*f)
```

fricas [A] time = 0.57, size = 273, normalized size = 1.10

$$\frac{3\,Cb^2d\tan\left(fx+e\right)^4+4\left(Cb^2c+\left(2\,Cab+Bb^2\right)d\right)\tan\left(fx+e\right)^3+12\left(\left((A-C)a^2-2\,Bab-(A-C)b^2\right)c-\left(Ba^2\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="fricas")
```

```
[Out] 1/12*(3*C*b^2*d*tan(f*x + e)^4 + 4*(C*b^2*c + (2*C*a*b + B*b^2)*d)*tan(f*x
+ e)^3 + 12*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c - (B*a^2 + 2*(A - C)*a
*b - B*b^2)*d)*f*x + 6*((2*C*a*b + B*b^2)*c + (C*a^2 + 2*B*a*b + (A - C)*b^
2)*d)*tan(f*x + e)^2 - 6*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c + ((A - C)*a^2
- 2*B*a*b - (A - C)*b^2)*d)*log(1/(tan(f*x + e)^2 + 1)) + 12*((C*a^2 + 2*B*
a*b + (A - C)*b^2)*c + (B*a^2 + 2*(A - C)*a*b - B*b^2)*d)*tan(f*x + e))/f
```

giac [B] time = 23.12, size = 6502, normalized size = 26.22

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="giac")
```

```
[Out] 1/12*(12*A*a^2*c*f*x*tan(f*x)^4*tan(e)^4 - 12*C*a^2*c*f*x*tan(f*x)^4*tan(e)
^4 - 24*B*a*b*c*f*x*tan(f*x)^4*tan(e)^4 - 12*A*b^2*c*f*x*tan(f*x)^4*tan(e)^
```

```
4 + 12*C*b^2*c*f*x*tan(f*x)^4*tan(e)^4 - 12*B*a^2*d*f*x*tan(f*x)^4*tan(e)^4
 - 24*A*a*b*d*f*x*tan(f*x)^4*tan(e)^4 + 24*C*a*b*d*f*x*tan(f*x)^4*tan(e)^4
+ 12*B*b^2*d*f*x*tan(f*x)^4*tan(e)^4 - 6*B*a^2*c*log(4*(tan(f*x)^4*tan(e)^2
 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(
e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 12*A*a*b*c*log(4*(tan(f*x)^4*
tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f
*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 12*C*a*b*c*log(4*(tan
(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 -
 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 6*B*b^2*c*log
(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 6*A*a^
2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2
+ tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 +
 6*C*a^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*ta
n(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan
(e)^4 + 12*B*a*b*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*
x)^4*tan(e)^4 + 6*A*b^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e)
+ tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))
*tan(f*x)^4*tan(e)^4 - 6*C*b^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*
tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^
2 + 1))*tan(f*x)^4*tan(e)^4 - 48*A*a^2*c*f*x*tan(f*x)^3*tan(e)^3 + 48*C*a^2
*c*f*x*tan(f*x)^3*tan(e)^3 + 96*B*a*b*c*f*x*tan(f*x)^3*tan(e)^3 + 48*A*b^2*
c*f*x*tan(f*x)^3*tan(e)^3 - 48*C*b^2*c*f*x*tan(f*x)^3*tan(e)^3 + 48*B*a^2*d
*f*x*tan(f*x)^3*tan(e)^3 + 96*A*a*b*d*f*x*tan(f*x)^3*tan(e)^3 - 96*C*a*b*d*
f*x*tan(f*x)^3*tan(e)^3 - 48*B*b^2*d*f*x*tan(f*x)^3*tan(e)^3 + 12*C*a*b*c*t
an(f*x)^4*tan(e)^4 + 6*B*b^2*c*tan(f*x)^4*tan(e)^4 + 6*C*a^2*d*tan(f*x)^4*t
an(e)^4 + 12*B*a*b*d*tan(f*x)^4*tan(e)^4 + 6*A*b^2*d*tan(f*x)^4*tan(e)^4 -
9*C*b^2*d*tan(f*x)^4*tan(e)^4 + 24*B*a^2*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*t
an(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1
)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 48*A*a*b*c*log(4*(tan(f*x)^4*tan(e)
^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 48*C*a*b*c*log(4*(tan(f*x)^
4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan
(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*B*b^2*c*log(4*(t
an(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2
 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 24*A*a^2*d*
log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + ta
n(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*
C*a^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e
)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)
^3 - 48*B*a*b*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)
^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^
3*tan(e)^3 - 24*A*b^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) +
tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*t
```

an(f*x)^3*tan(e)^3 + 24*C*b^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*t
an(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2
 + 1))*tan(f*x)^3*tan(e)^3 - 12*C*a^2*c*tan(f*x)^4*tan(e)^3 - 24*B*a*b*c*ta
n(f*x)^4*tan(e)^3 - 12*A*b^2*c*tan(f*x)^4*tan(e)^3 + 12*C*b^2*c*tan(f*x)^4*
tan(e)^3 - 12*B*a^2*d*tan(f*x)^4*tan(e)^3 - 24*A*a*b*d*tan(f*x)^4*tan(e)^3
+ 24*C*a*b*d*tan(f*x)^4*tan(e)^3 + 12*B*b^2*d*tan(f*x)^4*tan(e)^3 - 12*C*a^
2*c*tan(f*x)^3*tan(e)^4 - 24*B*a*b*c*tan(f*x)^3*tan(e)^4 - 12*A*b^2*c*tan(f
*x)^3*tan(e)^4 + 12*C*b^2*c*tan(f*x)^3*tan(e)^4 - 12*B*a^2*d*tan(f*x)^3*tan
(e)^4 - 24*A*a*b*d*tan(f*x)^3*tan(e)^4 + 24*C*a*b*d*tan(f*x)^3*tan(e)^4 + 1
2*B*b^2*d*tan(f*x)^3*tan(e)^4 + 72*A*a^2*c*f*x*tan(f*x)^2*tan(e)^2 - 72*C*a
^2*c*f*x*tan(f*x)^2*tan(e)^2 - 144*B*a*b*c*f*x*tan(f*x)^2*tan(e)^2 - 72*A*b
^2*c*f*x*tan(f*x)^2*tan(e)^2 + 72*C*b^2*c*f*x*tan(f*x)^2*tan(e)^2 - 72*B*a^
2*d*f*x*tan(f*x)^2*tan(e)^2 - 144*A*a*b*d*f*x*tan(f*x)^2*tan(e)^2 + 144*C*a
*b*d*f*x*tan(f*x)^2*tan(e)^2 + 72*B*b^2*d*f*x*tan(f*x)^2*tan(e)^2 + 12*C*a*
b*c*tan(f*x)^4*tan(e)^2 + 6*B*b^2*c*tan(f*x)^4*tan(e)^2 + 6*C*a^2*d*tan(f*x
)^4*tan(e)^2 + 12*B*a*b*d*tan(f*x)^4*tan(e)^2 + 6*A*b^2*d*tan(f*x)^4*tan(e)
^2 - 6*C*b^2*d*tan(f*x)^4*tan(e)^2 - 24*C*a*b*c*tan(f*x)^3*tan(e)^3 - 12*B*
b^2*c*tan(f*x)^3*tan(e)^3 - 12*C*a^2*d*tan(f*x)^3*tan(e)^3 - 24*B*a*b*d*tan
(f*x)^3*tan(e)^3 - 12*A*b^2*d*tan(f*x)^3*tan(e)^3 + 24*C*b^2*d*tan(f*x)^3*t
an(e)^3 + 12*C*a*b*c*tan(f*x)^2*tan(e)^4 + 6*B*b^2*c*tan(f*x)^2*tan(e)^4 +
6*C*a^2*d*tan(f*x)^2*tan(e)^4 + 12*B*a*b*d*tan(f*x)^2*tan(e)^4 + 6*A*b^2*d*
tan(f*x)^2*tan(e)^4 - 6*C*b^2*d*tan(f*x)^2*tan(e)^4 - 4*C*b^2*c*tan(f*x)^4*
tan(e) - 8*C*a*b*d*tan(f*x)^4*tan(e) - 4*B*b^2*d*tan(f*x)^4*tan(e) - 36*B*a
^2*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2
- 72*A*a*b*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*t
an(e)^2 + 72*C*a*b*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(
f*x)^2*tan(e)^2 + 36*B*b^2*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(
e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 +
1))*tan(f*x)^2*tan(e)^2 - 36*A*a^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x
)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*C*a^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2
*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) +
 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 72*B*a*b*d*log(4*(tan(f*x)^4*tan(
e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*
tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*A*b^2*d*log(4*(tan(f*x
)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*t
an(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 36*C*b^2*d*log(4*
(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*C*a^2*
c*tan(f*x)^3*tan(e)^2 + 72*B*a*b*c*tan(f*x)^3*tan(e)^2 + 36*A*b^2*c*tan(f*x
)^3*tan(e)^2 - 48*C*b^2*c*tan(f*x)^3*tan(e)^2 + 36*B*a^2*d*tan(f*x)^3*tan(e
)^2 + 72*A*a*b*d*tan(f*x)^3*tan(e)^2 - 96*C*a*b*d*tan(f*x)^3*tan(e)^2 - 48*

B*b^2*d*tan(f*x)^3*tan(e)^2 + 36*C*a^2*c*tan(f*x)^2*tan(e)^3 + 72*B*a*b*c*t
an(f*x)^2*tan(e)^3 + 36*A*b^2*c*tan(f*x)^2*tan(e)^3 - 48*C*b^2*c*tan(f*x)^2
*tan(e)^3 + 36*B*a^2*d*tan(f*x)^2*tan(e)^3 + 72*A*a*b*d*tan(f*x)^2*tan(e)^3
 - 96*C*a*b*d*tan(f*x)^2*tan(e)^3 - 48*B*b^2*d*tan(f*x)^2*tan(e)^3 - 4*C*b^
2*c*tan(f*x)*tan(e)^4 - 8*C*a*b*d*tan(f*x)*tan(e)^4 - 4*B*b^2*d*tan(f*x)*ta
n(e)^4 + 3*C*b^2*d*tan(f*x)^4 - 48*A*a^2*c*f*x*tan(f*x)*tan(e) + 48*C*a^2*c
*f*x*tan(f*x)*tan(e) + 96*B*a*b*c*f*x*tan(f*x)*tan(e) + 48*A*b^2*c*f*x*tan(
f*x)*tan(e) - 48*C*b^2*c*f*x*tan(f*x)*tan(e) + 48*B*a^2*d*f*x*tan(f*x)*tan(
e) + 96*A*a*b*d*f*x*tan(f*x)*tan(e) - 96*C*a*b*d*f*x*tan(f*x)*tan(e) - 48*B
*b^2*d*f*x*tan(f*x)*tan(e) - 24*C*a*b*c*tan(f*x)^3*tan(e) - 12*B*b^2*c*tan(
f*x)^3*tan(e) - 12*C*a^2*d*tan(f*x)^3*tan(e) - 24*B*a*b*d*tan(f*x)^3*tan(e)
 - 12*A*b^2*d*tan(f*x)^3*tan(e) + 24*C*b^2*d*tan(f*x)^3*tan(e) + 24*C*a*b*c
*tan(f*x)^2*tan(e)^2 + 12*B*b^2*c*tan(f*x)^2*tan(e)^2 + 12*C*a^2*d*tan(f*x)
^2*tan(e)^2 + 24*B*a*b*d*tan(f*x)^2*tan(e)^2 + 12*A*b^2*d*tan(f*x)^2*tan(e)
^2 - 12*C*b^2*d*tan(f*x)^2*tan(e)^2 - 24*C*a*b*c*tan(f*x)*tan(e)^3 - 12*B*b
^2*c*tan(f*x)*tan(e)^3 - 12*C*a^2*d*tan(f*x)*tan(e)^3 - 24*B*a*b*d*tan(f*x)
*tan(e)^3 - 12*A*b^2*d*tan(f*x)*tan(e)^3 + 24*C*b^2*d*tan(f*x)*tan(e)^3 + 3
*C*b^2*d*tan(e)^4 + 4*C*b^2*c*tan(f*x)^3 + 8*C*a*b*d*tan(f*x)^3 + 4*B*b^2*d
*tan(f*x)^3 + 24*B*a^2*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) +
 tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*
tan(f*x)*tan(e) + 48*A*a*b*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(
e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 +
1))*tan(f*x)*tan(e) - 48*C*a*b*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*
tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^
2 + 1))*tan(f*x)*tan(e) - 24*B*b^2*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x
)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(e)^2 + 1))*tan(f*x)*tan(e) + 24*A*a^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan
(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(e)^2 + 1))*tan(f*x)*tan(e) - 24*C*a^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2
*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) +
 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 48*B*a*b*d*log(4*(tan(f*x)^4*tan(e)^2
 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(
e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 24*A*b^2*d*log(4*(tan(f*x)^4*tan(
e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*
tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) + 24*C*b^2*d*log(4*(tan(f*x)^4*
tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f
*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 36*C*a^2*c*tan(f*x)^2*tan
(e) - 72*B*a*b*c*tan(f*x)^2*tan(e) - 36*A*b^2*c*tan(f*x)^2*tan(e) + 48*C*b^
2*c*tan(f*x)^2*tan(e) - 36*B*a^2*d*tan(f*x)^2*tan(e) - 72*A*a*b*d*tan(f*x)^
2*tan(e) + 96*C*a*b*d*tan(f*x)^2*tan(e) + 48*B*b^2*d*tan(f*x)^2*tan(e) - 36
*C*a^2*c*tan(f*x)*tan(e)^2 - 72*B*a*b*c*tan(f*x)*tan(e)^2 - 36*A*b^2*c*tan(
f*x)*tan(e)^2 + 48*C*b^2*c*tan(f*x)*tan(e)^2 - 36*B*a^2*d*tan(f*x)*tan(e)^2
 - 72*A*a*b*d*tan(f*x)*tan(e)^2 + 96*C*a*b*d*tan(f*x)*tan(e)^2 + 48*B*b^2*d
*tan(f*x)*tan(e)^2 + 4*C*b^2*c*tan(e)^3 + 8*C*a*b*d*tan(e)^3 + 4*B*b^2*d*ta
n(e)^3 + 12*A*a^2*c*f*x - 12*C*a^2*c*f*x - 24*B*a*b*c*f*x - 12*A*b^2*c*f*x

```
+ 12*C*b^2*c*f*x - 12*B*a^2*d*f*x - 24*A*a*b*d*f*x + 24*C*a*b*d*f*x + 12*B*
b^2*d*f*x + 12*C*a*b*c*tan(f*x)^2 + 6*B*b^2*c*tan(f*x)^2 + 6*C*a^2*d*tan(f*
x)^2 + 12*B*a*b*d*tan(f*x)^2 + 6*A*b^2*d*tan(f*x)^2 - 6*C*b^2*d*tan(f*x)^2
- 24*C*a*b*c*tan(f*x)*tan(e) - 12*B*b^2*c*tan(f*x)*tan(e) - 12*C*a^2*d*tan(
f*x)*tan(e) - 24*B*a*b*d*tan(f*x)*tan(e) - 12*A*b^2*d*tan(f*x)*tan(e) + 24*
C*b^2*d*tan(f*x)*tan(e) + 12*C*a*b*c*tan(e)^2 + 6*B*b^2*c*tan(e)^2 + 6*C*a^
2*d*tan(e)^2 + 12*B*a*b*d*tan(e)^2 + 6*A*b^2*d*tan(e)^2 - 6*C*b^2*d*tan(e)^
2 - 6*B*a^2*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - 12*A*a*b*
c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 12*C*a*b*c*log(4*(tan
(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 -
 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 6*B*b^2*c*log(4*(tan(f*x)^4*tan(e
)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(e)^2 + 1)) - 6*A*a^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f
*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(t
an(e)^2 + 1)) + 6*C*a^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e)
+ tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))
 + 12*B*a*b*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 6*A*b^2*d
*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + t
an(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - 6*C*b^2*d*log(4*(tan(f
*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2
*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 12*C*a^2*c*tan(f*x) + 24*B*a*b*c*ta
n(f*x) + 12*A*b^2*c*tan(f*x) - 12*C*b^2*c*tan(f*x) + 12*B*a^2*d*tan(f*x) +
24*A*a*b*d*tan(f*x) - 24*C*a*b*d*tan(f*x) - 12*B*b^2*d*tan(f*x) + 12*C*a^2*
c*tan(e) + 24*B*a*b*c*tan(e) + 12*A*b^2*c*tan(e) - 12*C*b^2*c*tan(e) + 12*B
*a^2*d*tan(e) + 24*A*a*b*d*tan(e) - 24*C*a*b*d*tan(e) - 12*B*b^2*d*tan(e) +
 12*C*a*b*c + 6*B*b^2*c + 6*C*a^2*d + 12*B*a*b*d + 6*A*b^2*d - 9*C*b^2*d)/(
f*tan(f*x)^4*tan(e)^4 - 4*f*tan(f*x)^3*tan(e)^3 + 6*f*tan(f*x)^2*tan(e)^2 -
 4*f*tan(f*x)*tan(e) + f)
```

maple [B] time = 0.03, size = 631, normalized size = 2.54

$$-\frac{C\arctan\left(\tan\left(fx+e\right)\right)a^{2}c}{f}+\frac{C\arctan\left(\tan\left(fx+e\right)\right)b^{2}c}{f}+\frac{2Aabd\tan\left(fx+e\right)}{f}+\frac{B\left(\tan^{3}\left(fx+e\right)\right)b^{2}d}{3f}+\frac{C\left(\tan\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] -1/f*C*arctan(tan(f*x+e))*a^2*c+1/f*C*arctan(tan(f*x+e))*b^2*c-1/2/f*ln(1+t
an(f*x+e)^2)*B*b^2*c+1/3/f*B*tan(f*x+e)^3*b^2*d+1/3/f*C*tan(f*x+e)^3*b^2*c+
2/f*A*a*b*d*tan(f*x+e)-1/f*B*b^2*d*tan(f*x+e)+1/f*B*a^2*d*tan(f*x+e)+1/f*A*
b^2*c*tan(f*x+e)-1/f*B*arctan(tan(f*x+e))*a^2*d+1/f*B*arctan(tan(f*x+e))*b^
```

```
2*d-1/2/f*ln(1+tan(f*x+e)^2)*C*a^2*d+1/2/f*A*tan(f*x+e)^2*b^2*d+1/2/f*B*tan
(f*x+e)^2*b^2*c-1/2/f*ln(1+tan(f*x+e)^2)*A*b^2*d+1/2/f*ln(1+tan(f*x+e)^2)*B
*a^2*c+1/2/f*C*tan(f*x+e)^2*a^2*d+1/4/f*C*b^2*d*tan(f*x+e)^4-1/f*C*b^2*c*ta
n(f*x+e)+1/2/f*ln(1+tan(f*x+e)^2)*A*a^2*d+1/f*C*a^2*c*tan(f*x+e)+1/2/f*ln(1
+tan(f*x+e)^2)*C*b^2*d+1/f*A*arctan(tan(f*x+e))*a^2*c-1/2/f*C*tan(f*x+e)^2*
b^2*d-1/f*A*arctan(tan(f*x+e))*b^2*c-1/f*ln(1+tan(f*x+e)^2)*C*a*b*c-2/f*A*a
rctan(tan(f*x+e))*a*b*d-2/f*C*a*b*d*tan(f*x+e)+1/f*C*tan(f*x+e)^2*a*b*c+2/f
*C*arctan(tan(f*x+e))*a*b*d-1/f*ln(1+tan(f*x+e)^2)*B*a*b*d+1/f*ln(1+tan(f*x
+e)^2)*A*a*b*c+1/f*B*tan(f*x+e)^2*a*b*d+2/f*B*a*b*c*tan(f*x+e)-2/f*B*arctan
(tan(f*x+e))*a*b*c+2/3/f*C*tan(f*x+e)^3*a*b*d
```

maxima [A] time = 0.57, size = 274, normalized size = 1.10

$$\frac{3\,Cb^2d\tan\left(fx+e\right)^4+4\left(Cb^2c+\left(2\,Cab+Bb^2\right)d\right)\tan\left(fx+e\right)^3+6\left(\left(2\,Cab+Bb^2\right)c+\left(Ca^2+2\,Bab+(A-C)b^2\right)d\right)\ldots}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="maxima")
```

```
[Out] 1/12*(3*C*b^2*d*tan(f*x + e)^4 + 4*(C*b^2*c + (2*C*a*b + B*b^2)*d)*tan(f*x
+ e)^3 + 6*((2*C*a*b + B*b^2)*c + (C*a^2 + 2*B*a*b + (A - C)*b^2)*d)*tan(f*
x + e)^2 + 12*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c - (B*a^2 + 2*(A - C)
*a*b - B*b^2)*d)*(f*x + e) + 6*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c + ((A - C
)*a^2 - 2*B*a*b - (A - C)*b^2)*d)*log(tan(f*x + e)^2 + 1) + 12*((C*a^2 + 2*
B*a*b + (A - C)*b^2)*c + (B*a^2 + 2*(A - C)*a*b - B*b^2)*d)*tan(f*x + e))/f
```

mupad [B] time = 8.98, size = 300, normalized size = 1.21

$$\frac{\tan\left(e+f\,x\right)^2\,\left(\frac{A\,b^2\,d}{2}+\frac{B\,b^2\,c}{2}+\frac{C\,a^2\,d}{2}-\frac{C\,b^2\,d}{2}+B\,a\,b\,d+C\,a\,b\,c\right)}{f}-x\,\left(A\,b^2\,c-A\,a^2\,c+B\,a^2\,d+C\,a^2\,c-B\,b^2\,d-C\,\ldots\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2),x)
```

```
[Out] (tan(e + f*x)^2*((A*b^2*d)/2 + (B*b^2*c)/2 + (C*a^2*d)/2 - (C*b^2*d)/2 + B*
a*b*d + C*a*b*c))/f - x*(A*b^2*c - A*a^2*c + B*a^2*d + C*a^2*c - B*b^2*d -
C*b^2*c + 2*A*a*b*d + 2*B*a*b*c - 2*C*a*b*d) - (log(tan(e + f*x)^2 + 1)*((A
*b^2*d)/2 - (B*a^2*c)/2 - (A*a^2*d)/2 + (B*b^2*c)/2 + (C*a^2*d)/2 - (C*b^2*
d)/2 - A*a*b*c + B*a*b*d + C*a*b*c))/f + (tan(e + f*x)*(A*b^2*c + B*a^2*d +
 C*a^2*c - B*b^2*d - C*b^2*c + 2*A*a*b*d + 2*B*a*b*c - 2*C*a*b*d))/f + (tan
```

```
(e + f*x)^3*((B*b^2*d)/3 + (C*b^2*c)/3 + (2*C*a*b*d)/3))/f + (C*b^2*d*tan(e
 + f*x)^4)/(4*f)
```

sympy [A] time = 0.98, size = 617, normalized size = 2.49

$$\begin{cases} Aa^2cx + \frac{Aa^2d\log\left(\tan^2\left(e+fx\right)+1\right)}{2f} + \frac{Aabc\log\left(\tan^2\left(e+fx\right)+1\right)}{f} - 2Aabdx + \frac{2Aabd\tan\left(e+fx\right)}{f} - Ab^2cx + \frac{Ab^2c\tan\left(e+fx\right)}{f} - \frac{Ab^2 \\ x\left(a+b\tan\left(e\right)\right)^2\left(c+d\tan\left(e\right)\right)\left(A+B\tan\left(e\right)+C\tan^2\left(e\right)\right) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)
**2),x)
```

```
[Out] Piecewise((A*a**2*c*x + A*a**2*d*log(tan(e + f*x)**2 + 1)/(2*f) + A*a*b*c*l
og(tan(e + f*x)**2 + 1)/f - 2*A*a*b*d*x + 2*A*a*b*d*tan(e + f*x)/f - A*b**2
*c*x + A*b**2*c*tan(e + f*x)/f - A*b**2*d*log(tan(e + f*x)**2 + 1)/(2*f) +
A*b**2*d*tan(e + f*x)**2/(2*f) + B*a**2*c*log(tan(e + f*x)**2 + 1)/(2*f) -
B*a**2*d*x + B*a**2*d*tan(e + f*x)/f - 2*B*a*b*c*x + 2*B*a*b*c*tan(e + f*x)
/f - B*a*b*d*log(tan(e + f*x)**2 + 1)/f + B*a*b*d*tan(e + f*x)**2/f - B*b**
2*c*log(tan(e + f*x)**2 + 1)/(2*f) + B*b**2*c*tan(e + f*x)**2/(2*f) + B*b**
2*d*x + B*b**2*d*tan(e + f*x)**3/(3*f) - B*b**2*d*tan(e + f*x)/f - C*a**2*c
*x + C*a**2*c*tan(e + f*x)/f - C*a**2*d*log(tan(e + f*x)**2 + 1)/(2*f) + C*
a**2*d*tan(e + f*x)**2/(2*f) - C*a*b*c*log(tan(e + f*x)**2 + 1)/f + C*a*b*c
*tan(e + f*x)**2/f + 2*C*a*b*d*x + 2*C*a*b*d*tan(e + f*x)**3/(3*f) - 2*C*a*
b*d*tan(e + f*x)/f + C*b**2*c*x + C*b**2*c*tan(e + f*x)**3/(3*f) - C*b**2*c
*tan(e + f*x)/f + C*b**2*d*log(tan(e + f*x)**2 + 1)/(2*f) + C*b**2*d*tan(e
+ f*x)**4/(4*f) - C*b**2*d*tan(e + f*x)**2/(2*f), Ne(f, 0)), (x*(a + b*tan(
e))**2*(c + d*tan(e))*(A + B*tan(e) + C*tan(e)**2), True))
```

3.52 $\int (a+b\tan(e+fx))(c+d\tan(e+fx)) \left(A+B\tan(e+fx)+\right.$

Optimal. Leaf size=161

$$-\frac{\log(\cos(e+fx))(aAd+aBc-aCd+Abc-bBd-bcC)}{f}+x(a(Ac-Bd-cC)-b(d(A-C)+Bc))+\frac{d\tan(e+fx)(aB}{f}$$

```
[Out] (a*(A*c-B*d-C*c)-b*(B*c+(A-C)*d))*x-(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b*c)*l
n(cos(f*x+e))/f+(A*b+B*a-C*b)*d*tan(f*x+e)/f-1/6*(-3*B*b*d-3*C*a*d+C*b*c)*(
c+d*tan(f*x+e))^2/d^2/f+1/3*b*C*tan(f*x+e)*(c+d*tan(f*x+e))^2/d/f
```

Rubi [A] time = 0.24, antiderivative size = 161, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 41, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.098, Rules used = {3637, 3630, 3525, 3475}

$$-\frac{\log(\cos(e+fx))(aAd+aBc-aCd+Abc-bBd-bcC)}{f}-x(-a(Ac-Bd-cC)+bd(A-C)+bBc)+\frac{d\tan(e+fx)(aB}{f}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e
 + f*x]^2),x]
```

```
[Out] -((b*B*c + b*(A - C)*d - a*(A*c - c*C - B*d))*x) - ((A*b*c + a*B*c - b*c*C
+ a*A*d - b*B*d - a*C*d)*Log[Cos[e + f*x]])/f + ((A*b + a*B - b*C)*d*Tan[e
+ f*x])/f - ((b*c*C - 3*b*B*d - 3*a*C*d)*(c + d*Tan[e + f*x])^2)/(6*d^2*f)
+ (b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^2)/(3*d*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
```

```
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{bC\tan(e+fx)(c+d\tan(e}{3df}$$

$$= -\frac{(bcC-3bBd-3aCd)(c+}{6d^2f}$$

$$= -(bBc+b(A-C)d-a(Ac$$

$$= -(bBc+b(A-C)d-a(Ac$$

Mathematica [C] time = 1.62, size = 161, normalized size = 1.00

$$\frac{3(a+ib)(d-ic)(A+iB-C)\log(-\tan(e+fx)+i)+3(a-ib)(d+ic)(A-iB-C)\log(\tan(e+fx)+i)+6d\,\mathrm{ta}}{6f}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C
*Tan[e + f*x]^2),x]
```

```
[Out] (3*(a + I*b)*(A + I*B - C)*((-I)*c + d)*Log[I - Tan[e + f*x]] + 3*(a - I*b)
*(A - I*B - C)*(I*c + d)*Log[I + Tan[e + f*x]] + 6*(A*b + a*B - b*C)*d*Tan[
e + f*x] + ((-(b*c*C) + 3*b*B*d + 3*a*C*d)*(c + d*Tan[e + f*x])^2)/d^2 + (2
*b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^2)/d)/(6*f)
```

fricas [A] time = 0.59, size = 150, normalized size = 0.93

$$\frac{2\,Cbd\tan\left(fx+e\right)^{3}+6\left(\left(\left(A-C\right)a-Bb\right)c-\left(Ba+\left(A-C\right)b\right)d\right)fx+3\left(Cbc+\left(Ca+Bb\right)d\right)\tan\left(fx+e\right)^{2}-3\left(\left(Ba\right.\right.}{6\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x, algorithm="fricas")
```

```
[Out] 1/6*(2*C*b*d*tan(f*x + e)^3 + 6*(((A - C)*a - B*b)*c - (B*a + (A - C)*b)*d)
*f*x + 3*(C*b*c + (C*a + B*b)*d)*tan(f*x + e)^2 - 3*((B*a + (A - C)*b)*c +
((A - C)*a - B*b)*d)*log(1/(tan(f*x + e)^2 + 1)) + 6*((C*a + B*b)*c + (B*a
+ (A - C)*b)*d)*tan(f*x + e))/f
```

giac [B] time = 7.34, size = 2918, normalized size = 18.12

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x, algorithm="giac")
```

```
[Out] 1/6*(6*A*a*c*f*x*tan(f*x)^3*tan(e)^3 - 6*C*a*c*f*x*tan(f*x)^3*tan(e)^3 - 6*
B*b*c*f*x*tan(f*x)^3*tan(e)^3 - 6*B*a*d*f*x*tan(f*x)^3*tan(e)^3 - 6*A*b*d*f
*x*tan(f*x)^3*tan(e)^3 + 6*C*b*d*f*x*tan(f*x)^3*tan(e)^3 - 3*B*a*c*log(4*(t
an(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2
 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 3*A*b*c*log
(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 3*C*b*
c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 +
tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 3
*A*a*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)
^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^
3 + 3*C*a*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*ta
n(e)^3 + 3*B*b*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x
)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)
^3*tan(e)^3 - 18*A*a*c*f*x*tan(f*x)^2*tan(e)^2 + 18*C*a*c*f*x*tan(f*x)^2*ta
n(e)^2 + 18*B*b*c*f*x*tan(f*x)^2*tan(e)^2 + 18*B*a*d*f*x*tan(f*x)^2*tan(e)^
2 + 18*A*b*d*f*x*tan(f*x)^2*tan(e)^2 - 18*C*b*d*f*x*tan(f*x)^2*tan(e)^2 + 3
*C*b*c*tan(f*x)^3*tan(e)^3 + 3*C*a*d*tan(f*x)^3*tan(e)^3 + 3*B*b*d*tan(f*x)
^3*tan(e)^3 + 9*B*a*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + ta
n(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan
```

```
(f*x)^2*tan(e)^2 + 9*A*b*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e)
 + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)
)*tan(f*x)^2*tan(e)^2 - 9*C*b*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*t
an(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2
 + 1))*tan(f*x)^2*tan(e)^2 + 9*A*a*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x
)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 9*C*a*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*ta
n(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)
/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 9*B*b*d*log(4*(tan(f*x)^4*tan(e)^2 -
 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e)
 + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 6*C*a*c*tan(f*x)^3*tan(e)^2 - 6
*B*b*c*tan(f*x)^3*tan(e)^2 - 6*B*a*d*tan(f*x)^3*tan(e)^2 - 6*A*b*d*tan(f*x)
^3*tan(e)^2 + 6*C*b*d*tan(f*x)^3*tan(e)^2 - 6*C*a*c*tan(f*x)^2*tan(e)^3 - 6
*B*b*c*tan(f*x)^2*tan(e)^3 - 6*B*a*d*tan(f*x)^2*tan(e)^3 - 6*A*b*d*tan(f*x)
^2*tan(e)^3 + 6*C*b*d*tan(f*x)^2*tan(e)^3 + 18*A*a*c*f*x*tan(f*x)*tan(e) -
18*C*a*c*f*x*tan(f*x)*tan(e) - 18*B*b*c*f*x*tan(f*x)*tan(e) - 18*B*a*d*f*x*
tan(f*x)*tan(e) - 18*A*b*d*f*x*tan(f*x)*tan(e) + 18*C*b*d*f*x*tan(f*x)*tan(
e) + 3*C*b*c*tan(f*x)^3*tan(e) + 3*C*a*d*tan(f*x)^3*tan(e) + 3*B*b*d*tan(f*
x)^3*tan(e) - 3*C*b*c*tan(f*x)^2*tan(e)^2 - 3*C*a*d*tan(f*x)^2*tan(e)^2 - 3
*B*b*d*tan(f*x)^2*tan(e)^2 + 3*C*b*c*tan(f*x)*tan(e)^3 + 3*C*a*d*tan(f*x)*t
an(e)^3 + 3*B*b*d*tan(f*x)*tan(e)^3 - 2*C*b*d*tan(f*x)^3 - 9*B*a*c*log(4*(t
an(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2
 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 9*A*b*c*log(4*(
tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) + 9*C*b*c*log(4*
(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 9*A*a*d*log(4
*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x
)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) + 9*C*a*d*log(
4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*
x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) + 9*B*b*d*log
(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f
*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) + 12*C*a*c*t
an(f*x)^2*tan(e) + 12*B*b*c*tan(f*x)^2*tan(e) + 12*B*a*d*tan(f*x)^2*tan(e)
+ 12*A*b*d*tan(f*x)^2*tan(e) - 18*C*b*d*tan(f*x)^2*tan(e) + 12*C*a*c*tan(f*
x)*tan(e)^2 + 12*B*b*c*tan(f*x)*tan(e)^2 + 12*B*a*d*tan(f*x)*tan(e)^2 + 12*
A*b*d*tan(f*x)*tan(e)^2 - 18*C*b*d*tan(f*x)*tan(e)^2 - 2*C*b*d*tan(e)^3 - 6
*A*a*c*f*x + 6*C*a*c*f*x + 6*B*b*c*f*x + 6*B*a*d*f*x + 6*A*b*d*f*x - 6*C*b*
d*f*x - 3*C*b*c*tan(f*x)^2 - 3*C*a*d*tan(f*x)^2 - 3*B*b*d*tan(f*x)^2 + 3*C*
b*c*tan(f*x)*tan(e) + 3*C*a*d*tan(f*x)*tan(e) + 3*B*b*d*tan(f*x)*tan(e) - 3
*C*b*c*tan(e)^2 - 3*C*a*d*tan(e)^2 - 3*B*b*d*tan(e)^2 + 3*B*a*c*log(4*(tan(
f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 3*A*b*c*log(4*(tan(f*x)^4*tan(e)^2
 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(
e) + 1)/(tan(e)^2 + 1)) - 3*C*b*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3
```

```
*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)
^2 + 1)) + 3*A*a*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f
*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - 3*C*
a*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2
+ tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - 3*B*b*d*log(4*(tan(
f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - 6*C*a*c*tan(f*x) - 6*B*b*c*tan(f*x
) - 6*B*a*d*tan(f*x) - 6*A*b*d*tan(f*x) + 6*C*b*d*tan(f*x) - 6*C*a*c*tan(e)
 - 6*B*b*c*tan(e) - 6*B*a*d*tan(e) - 6*A*b*d*tan(e) + 6*C*b*d*tan(e) - 3*C*
b*c - 3*C*a*d - 3*B*b*d)/(f*tan(f*x)^3*tan(e)^3 - 3*f*tan(f*x)^2*tan(e)^2 +
 3*f*tan(f*x)*tan(e) - f)
```

maple [B] time = 0.02, size = 334, normalized size = 2.07

$$\frac{Cbd\left(\tan^{3}\left(fx+e\right)\right)}{3f}+\frac{B\left(\tan^{2}\left(fx+e\right)\right)bd}{2f}+\frac{C\left(\tan^{2}\left(fx+e\right)\right)ad}{2f}+\frac{C\left(\tan^{2}\left(fx+e\right)\right)bc}{2f}+\frac{Abd\tan\left(fx+e\right)}{f}+\frac{Bad$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] 1/3/f*C*b*d*tan(f*x+e)^3+1/2/f*B*tan(f*x+e)^2*b*d+1/2/f*C*tan(f*x+e)^2*a*d+
1/2/f*C*tan(f*x+e)^2*b*c+1/f*A*b*d*tan(f*x+e)+1/f*B*a*d*tan(f*x+e)+1/f*B*b*
c*tan(f*x+e)+1/f*C*a*c*tan(f*x+e)-1/f*C*b*d*tan(f*x+e)+1/2/f*ln(1+tan(f*x+e
)^2)*A*a*d+1/2/f*ln(1+tan(f*x+e)^2)*A*b*c+1/2/f*ln(1+tan(f*x+e)^2)*B*a*c-1/
2/f*ln(1+tan(f*x+e)^2)*B*b*d-1/2/f*ln(1+tan(f*x+e)^2)*a*C*d-1/2/f*ln(1+tan(
f*x+e)^2)*C*b*c+1/f*A*arctan(tan(f*x+e))*a*c-1/f*A*arctan(tan(f*x+e))*b*d-1
/f*B*arctan(tan(f*x+e))*a*d-1/f*B*arctan(tan(f*x+e))*b*c-1/f*C*arctan(tan(f
*x+e))*a*c+1/f*C*arctan(tan(f*x+e))*b*d
```

maxima [A] time = 0.44, size = 151, normalized size = 0.94

$$\frac{2\,Cbd\tan\left(fx+e\right)^{3}+3\,(Cbc+(Ca+Bb)d)\tan\left(fx+e\right)^{2}+6\,(((A-C)a-Bb)c-(Ba+(A-C)b)d)\left(fx+e\right)+3}{6}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x, algorithm="maxima")
```

```
[Out] 1/6*(2*C*b*d*tan(f*x + e)^3 + 3*(C*b*c + (C*a + B*b)*d)*tan(f*x + e)^2 + 6*
(((A - C)*a - B*b)*c - (B*a + (A - C)*b)*d)*(f*x + e) + 3*((B*a + (A - C)*b
)*c + ((A - C)*a - B*b)*d)*log(tan(f*x + e)^2 + 1) + 6*((C*a + B*b)*c + (B*
a + (A - C)*b)*d)*tan(f*x + e))/f
```

mupad [B] time = 8.84, size = 153, normalized size = 0.95

$$\frac{\ln\left(\tan\left(e+f\,x\right)^2+1\right)\,\left(\frac{A\,a\,d}{2}+\frac{A\,b\,c}{2}+\frac{B\,a\,c}{2}-\frac{B\,b\,d}{2}-\frac{C\,a\,d}{2}-\frac{C\,b\,c}{2}\right)}{f}-x\,\left(A\,b\,d-A\,a\,c+B\,a\,d+B\,b\,c+C\,a\,c-C\,b\,d\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e
 + f*x)^2),x)
```

```
[Out] (log(tan(e + f*x)^2 + 1)*((A*a*d)/2 + (A*b*c)/2 + (B*a*c)/2 - (B*b*d)/2 - (
C*a*d)/2 - (C*b*c)/2))/f - x*(A*b*d - A*a*c + B*a*d + B*b*c + C*a*c - C*b*d
) + (tan(e + f*x)^2*((B*b*d)/2 + (C*a*d)/2 + (C*b*c)/2))/f + (tan(e + f*x)*
(A*b*d + B*a*d + B*b*c + C*a*c - C*b*d))/f + (C*b*d*tan(e + f*x)^3)/(3*f)
```

sympy [A] time = 0.50, size = 326, normalized size = 2.02

$$\begin{cases} Aacx + \frac{Aad\log\left(\tan^2\left(e+fx\right)+1\right)}{2f} + \frac{Abc\log\left(\tan^2\left(e+fx\right)+1\right)}{2f} - Abdx + \frac{Abd\tan\left(e+fx\right)}{f} + \frac{Bac\log\left(\tan^2\left(e+fx\right)+1\right)}{2f} - Badx + \frac{Bad\,\mathrm{ta}}{} \\ x\left(a+b\tan\left(e\right)\right)\left(c+d\tan\left(e\right)\right)\left(A+B\tan\left(e\right)+C\tan^2\left(e\right)\right) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2
),x)
```

```
[Out] Piecewise((A*a*c*x + A*a*d*log(tan(e + f*x)**2 + 1)/(2*f) + A*b*c*log(tan(e
 + f*x)**2 + 1)/(2*f) - A*b*d*x + A*b*d*tan(e + f*x)/f + B*a*c*log(tan(e +
f*x)**2 + 1)/(2*f) - B*a*d*x + B*a*d*tan(e + f*x)/f - B*b*c*x + B*b*c*tan(e
 + f*x)/f - B*b*d*log(tan(e + f*x)**2 + 1)/(2*f) + B*b*d*tan(e + f*x)**2/(2
*f) - C*a*c*x + C*a*c*tan(e + f*x)/f - C*a*d*log(tan(e + f*x)**2 + 1)/(2*f)
 + C*a*d*tan(e + f*x)**2/(2*f) - C*b*c*log(tan(e + f*x)**2 + 1)/(2*f) + C*b
*c*tan(e + f*x)**2/(2*f) + C*b*d*x + C*b*d*tan(e + f*x)**3/(3*f) - C*b*d*ta
n(e + f*x)/f, Ne(f, 0)), (x*(a + b*tan(e))*(c + d*tan(e))*(A + B*tan(e) + C
*tan(e)**2), True))
```

3.53 $\int (c+d\tan(e+fx)) \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right)$

Optimal. Leaf size=73

$$-\frac{(d(A-C)+Bc)\log(\cos(e+fx))}{f}+x(Ac-Bd-cC)+\frac{Bd\tan(e+fx)}{f}+\frac{C(c+d\tan(e+fx))^2}{2df}$$

```
[Out] (A*c-B*d-C*c)*x-(B*c+(A-C)*d)*ln(cos(f*x+e))/f+B*d*tan(f*x+e)/f+1/2*C*(c+d*
tan(f*x+e))^2/d/f
```

Rubi [A] time = 0.06, antiderivative size = 73, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, integrand size = 31, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.097, Rules used = {3630, 3525, 3475}

$$-\frac{(d(A-C)+Bc)\log(\cos(e+fx))}{f}+x(Ac-Bd-cC)+\frac{Bd\tan(e+fx)}{f}+\frac{C(c+d\tan(e+fx))^2}{2df}$$

Antiderivative was successfully verified.

```
[In] Int[(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (A*c - c*C - B*d)*x - ((B*c + (A - C)*d)*Log[Cos[e + f*x]])/f + (B*d*Tan[e
+ f*x])/f + (C*(c + d*Tan[e + f*x])^2)/(2*d*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int (c + d\tan(e + fx)) \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right) dx = \frac{C(c + d\tan(e + fx))^2}{2df} + \int (A - C + B\tan(e$$

$$= (Ac - cC - Bd)x + \frac{Bd\tan(e + fx)}{f} + \frac{C(c + d}{}$$

$$= (Ac - cC - Bd)x - \frac{(Bc + (A - C)d)\log(\cos(e}{f}$$

Mathematica [A] time = 0.47, size = 76, normalized size = 1.04

$$\frac{-2(d(A - C) + Bc)\log(\cos(e + fx)) + 2Acfx - 2(Bd + cC)\tan^{-1}(\tan(e + fx)) + 2(Bd + cC)\tan(e + fx) + Cd}{2f}$$

Antiderivative was successfully verified.

```
[In] Integrate[(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*A*c*f*x - 2*(c*C + B*d)*ArcTan[Tan[e + f*x]] - 2*(B*c + (A - C)*d)*Log[C
os[e + f*x]] + 2*(c*C + B*d)*Tan[e + f*x] + C*d*Tan[e + f*x]^2)/(2*f)
```

fricas [A] time = 1.33, size = 74, normalized size = 1.01

$$\frac{Cd\tan\left(fx + e\right)^2 + 2\left((A - C)c - Bd\right)fx - (Bc + (A - C)d)\log\left(\frac{1}{\tan\left(fx+e\right)^2+1}\right) + 2\left(Cc + Bd\right)\tan\left(fx + e\right)}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="fr
icas")
```

```
[Out] 1/2*(C*d*tan(f*x + e)^2 + 2*((A - C)*c - B*d)*f*x - (B*c + (A - C)*d)*log(1
/(tan(f*x + e)^2 + 1)) + 2*(C*c + B*d)*tan(f*x + e))/f
```

giac [B] time = 2.93, size = 918, normalized size = 12.58

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="gi
ac")
```

```
[Out] 1/2*(2*A*c*f*x*tan(f*x)^2*tan(e)^2 - 2*C*c*f*x*tan(f*x)^2*tan(e)^2 - 2*B*d*
f*x*tan(f*x)^2*tan(e)^2 - B*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan
(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 +
 1))*tan(f*x)^2*tan(e)^2 - A*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*ta
n(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2
+ 1))*tan(f*x)^2*tan(e)^2 + C*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*t
an(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2
 + 1))*tan(f*x)^2*tan(e)^2 - 4*A*c*f*x*tan(f*x)*tan(e) + 4*C*c*f*x*tan(f*x)
*tan(e) + 4*B*d*f*x*tan(f*x)*tan(e) + C*d*tan(f*x)^2*tan(e)^2 + 2*B*c*log(4
*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x
)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) + 2*A*d*log(4*
(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 2*C*d*log(4*(
tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 2*C*c*tan(f*x)
^2*tan(e) - 2*B*d*tan(f*x)^2*tan(e) - 2*C*c*tan(f*x)*tan(e)^2 - 2*B*d*tan(f
*x)*tan(e)^2 + 2*A*c*f*x - 2*C*c*f*x - 2*B*d*f*x + C*d*tan(f*x)^2 + C*d*tan
(e)^2 - B*c*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*t
an(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - A*d*log(4*(
tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^
2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + C*d*log(4*(tan(f*x)^4*tan(e)^2
 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(
e) + 1)/(tan(e)^2 + 1)) + 2*C*c*tan(f*x) + 2*B*d*tan(f*x) + 2*C*c*tan(e) +
2*B*d*tan(e) + C*d)/(f*tan(f*x)^2*tan(e)^2 - 2*f*tan(f*x)*tan(e) + f)
```

maple [A] time = 0.02, size = 136, normalized size = 1.86

$$\frac{Cd\left(\tan^2\left(fx+e\right)\right)}{2f}+\frac{Bd\tan\left(fx+e\right)}{f}+\frac{cC\tan\left(fx+e\right)}{f}+\frac{\ln\left(1+\tan^2\left(fx+e\right)\right)Ad}{2f}+\frac{\ln\left(1+\tan^2\left(fx+e\right)\right)Bc}{2f}-\frac{\ln}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] 1/2/f*C*d*tan(f*x+e)^2+B*d*tan(f*x+e)/f+1/f*c*C*tan(f*x+e)+1/2/f*ln(1+tan(f
*x+e)^2)*A*d+1/2/f*ln(1+tan(f*x+e)^2)*B*c-1/2/f*ln(1+tan(f*x+e)^2)*C*d+1/f*
A*arctan(tan(f*x+e))*c-1/f*B*arctan(tan(f*x+e))*d-1/f*C*arctan(tan(f*x+e))*
c
```

maxima [A] time = 0.51, size = 74, normalized size = 1.01

$$\frac{Cd\tan\left(fx+e\right)^2+2\left((A-C)c-Bd\right)\left(fx+e\right)+\left(Bc+(A-C)d\right)\log\left(\tan\left(fx+e\right)^2+1\right)+2\left(Cc+Bd\right)\tan\left(fx+\right.}{2f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="ma
xima")
```

```
[Out] 1/2*(C*d*tan(f*x + e)^2 + 2*((A - C)*c - B*d)*(f*x + e) + (B*c + (A - C)*d)
*log(tan(f*x + e)^2 + 1) + 2*(C*c + B*d)*tan(f*x + e))/f
```

mupad [B] time = 8.68, size = 75, normalized size = 1.03

$$\frac{\tan\left(e+f\,x\right)\,\left(B\,d+C\,c\right)}{f}-x\,\left(B\,d-A\,c+C\,c\right)+\frac{\ln\left(\tan\left(e+f\,x\right)^2+1\right)\,\left(\frac{A\,d}{2}+\frac{B\,c}{2}-\frac{C\,d}{2}\right)}{f}+\frac{C\,d\tan\left(e+f\,x\right)^2}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] (tan(e + f*x)*(B*d + C*c))/f - x*(B*d - A*c + C*c) + (log(tan(e + f*x)^2 +
1)*((A*d)/2 + (B*c)/2 - (C*d)/2))/f + (C*d*tan(e + f*x)^2)/(2*f)
```

sympy [A] time = 0.28, size = 131, normalized size = 1.79

$$\begin{cases} Acx + \frac{Ad\log\left(\tan^2\left(e+fx\right)+1\right)}{2f} + \frac{Bc\log\left(\tan^2\left(e+fx\right)+1\right)}{2f} - Bdx + \frac{Bd\tan\left(e+fx\right)}{f} - Ccx + \frac{Cc\tan\left(e+fx\right)}{f} - \frac{Cd\log\left(\tan^2\left(e+fx\right)+1\right)}{2f} + \\ x\left(c+d\tan\left(e\right)\right)\left(A+B\tan\left(e\right)+C\tan^2\left(e\right)\right) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Piecewise((A*c*x + A*d*log(tan(e + f*x)**2 + 1)/(2*f) + B*c*log(tan(e + f*x
)**2 + 1)/(2*f) - B*d*x + B*d*tan(e + f*x)/f - C*c*x + C*c*tan(e + f*x)/f -
 C*d*log(tan(e + f*x)**2 + 1)/(2*f) + C*d*tan(e + f*x)**2/(2*f), Ne(f, 0)),
 (x*(c + d*tan(e))*(A + B*tan(e) + C*tan(e)**2), True))
```

3.54 $\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$

Optimal. Leaf size=156

$$\frac{(bc-ad)\left(Ab^2-a(bB-aC)\right)\log(a+b\tan(e+fx))}{b^2f\left(a^2+b^2\right)}+\frac{\log(\cos(e+fx))(-aAd-aBc+aCd+Abc-bBd-bcC)}{f\left(a^2+b^2\right)}+$$

```
[Out] (a*(A*c-B*d-C*c)+b*(B*c+(A-C)*d))*x/(a^2+b^2)+(-A*a*d+A*b*c-B*a*c-B*b*d+C*a
*d-C*b*c)*ln(cos(f*x+e))/(a^2+b^2)/f+(A*b^2-a*(B*b-C*a))*(-a*d+b*c)*ln(a+b*
tan(f*x+e))/b^2/(a^2+b^2)/f+C*d*tan(f*x+e)/b/f
```

Rubi [A] time = 0.35, antiderivative size = 155, normalized size of antiderivative = 0.99, number of steps used = 5, number of rules used = 5, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3637, 3626, 3617, 31, 3475}

$$\frac{(bc-ad)\left(Ab^2-a(bB-aC)\right)\log(a+b\tan(e+fx))}{b^2f\left(a^2+b^2\right)}+\frac{\log(\cos(e+fx))(-aAd-aBc+aCd+Abc-bBd-bcC)}{f\left(a^2+b^2\right)}+$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b*T
an[e + f*x]),x]
```

```
[Out] ((b*B*c + b*(A - C)*d + a*(A*c - c*C - B*d))*x)/(a^2 + b^2) + ((A*b*c - a*B
*c - b*c*C - a*A*d - b*B*d + a*C*d)*Log[Cos[e + f*x]])/((a^2 + b^2)*f) + ((
A*b^2 - a*(b*B - a*C))*(b*c - a*d)*Log[a + b*Tan[e + f*x]])/(b^2*(a^2 + b^2
)*f) + (C*d*Tan[e + f*x])/(b*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \frac{Cd\tan(e+fx)}{bf} - \frac{\int \frac{-Abc+aCd-b(Bc+(A-C)d)\tan(e}{a+b\,\mathrm{ta}}}{}$$

$$= \frac{(bBc+b(A-C)d+a(Ac-cC-Bd))x}{a^2+b^2} + \frac{Cd}{}$$

$$= \frac{(bBc+b(A-C)d+a(Ac-cC-Bd))x}{a^2+b^2} + \frac{(A}{}$$

$$= \frac{(bBc+b(A-C)d+a(Ac-cC-Bd))x}{a^2+b^2} + \frac{(A}{}$$

Mathematica [C] time = 1.19, size = 148, normalized size = 0.95

$$\frac{\frac{2(bc-ad)\left(a(aC-bB)+Ab^2\right)\log(a+b\tan(e+fx))}{b^2\left(a^2+b^2\right)}+\frac{(d-ic)(A+iB-C)\log(-\tan(e+fx)+i)}{a+ib}+\frac{(d+ic)(A-iB-C)\log(\tan(e+fx)+i)}{a-ib}+\frac{2Cd\tan(e+fx)}{b}}{2f}$$

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x]),x]
```

```
[Out] (((A + I*B - C)*((-I)*c + d)*Log[I - Tan[e + f*x]])/(a + I*b) + ((A - I*B -
 C)*(I*c + d)*Log[I + Tan[e + f*x]])/(a - I*b) + (2*(A*b^2 + a*(-(b*B) + a*
C))*(b*c - a*d)*Log[a + b*Tan[e + f*x]])/(b^2*(a^2 + b^2)) + (2*C*d*Tan[e +
 f*x])/b)/(2*f)
```

fricas [A] time = 1.44, size = 226, normalized size = 1.45

$$\frac{2\left(\left((A-C)ab^2+Bb^3\right)c-\left(Bab^2-(A-C)b^3\right)d\right)fx+2\left(Ca^2b+Cb^3\right)d\tan\left(fx+e\right)+\left(\left(Ca^2b-Bab^2+Ab^3\right)c-\left(\right.\right.}{2}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a*b^2 + B*b^3)*c - (B*a*b^2 - (A - C)*b^3)*d)*f*x + 2*(C*a
^2*b + C*b^3)*d*tan(f*x + e) + ((C*a^2*b - B*a*b^2 + A*b^3)*c - (C*a^3 - B*
a^2*b + A*a*b^2)*d)*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(ta
n(f*x + e)^2 + 1)) - ((C*a^2*b + C*b^3)*c - (C*a^3 - B*a^2*b + C*a*b^2 - B*
b^3)*d)*log(1/(tan(f*x + e)^2 + 1)))/((a^2*b^2 + b^4)*f)
```

giac [A] time = 1.79, size = 186, normalized size = 1.19

$$\frac{\frac{2\,Cd\tan\left(fx+e\right)}{b}+\frac{2\,(Aac-Cac+Bbc-Bad+Abd-Cbd)\left(fx+e\right)}{a^2+b^2}+\frac{(Bac-Abc+Cbc+Aad-Cad+Bbd)\log\left(\tan\left(fx+e\right)^2+1\right)}{a^2+b^2}+\frac{2\left(Ca^2bc-Bab^2c+Ab^3c-Ca\right.}{}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*d*tan(f*x + e)/b + 2*(A*a*c - C*a*c + B*b*c - B*a*d + A*b*d - C*b*
d)*(f*x + e)/(a^2 + b^2) + (B*a*c - A*b*c + C*b*c + A*a*d - C*a*d + B*b*d)*
log(tan(f*x + e)^2 + 1)/(a^2 + b^2) + 2*(C*a^2*b*c - B*a*b^2*c + A*b^3*c -
C*a^3*d + B*a^2*b*d - A*a*b^2*d)*log(abs(b*tan(f*x + e) + a))/(a^2*b^2 + b^
4))/f
```

maple [B] time = 0.24, size = 506, normalized size = 3.24

$$\frac{Cd\tan\left(fx+e\right)}{bf}-\frac{\ln\left(a+b\tan\left(fx+e\right)\right)Aad}{f\left(a^2+b^2\right)}+\frac{b\ln\left(a+b\tan\left(fx+e\right)\right)Ac}{f\left(a^2+b^2\right)}+\frac{\ln\left(a+b\tan\left(fx+e\right)\right)Ba^2d}{fb\left(a^2+b^2\right)}-\frac{\ln\left(a\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e)),x)
```

```
[Out] C*d*tan(f*x+e)/b/f-1/f/(a^2+b^2)*ln(a+b*tan(f*x+e))*A*a*d+1/f*b/(a^2+b^2)*l
n(a+b*tan(f*x+e))*A*c+1/f/b/(a^2+b^2)*ln(a+b*tan(f*x+e))*B*a^2*d-1/f/(a^2+b
^2)*ln(a+b*tan(f*x+e))*B*a*c-1/f/b^2/(a^2+b^2)*ln(a+b*tan(f*x+e))*a^3*C*d+1
/f/b/(a^2+b^2)*ln(a+b*tan(f*x+e))*C*a^2*c+1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2
)*A*a*d-1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*A*b*c+1/2/f/(a^2+b^2)*ln(1+tan(f
*x+e)^2)*B*a*c+1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*B*b*d-1/2/f/(a^2+b^2)*ln(
1+tan(f*x+e)^2)*a*C*d+1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*C*b*c+1/f/(a^2+b^2
)*A*arctan(tan(f*x+e))*a*c+1/f/(a^2+b^2)*A*arctan(tan(f*x+e))*b*d-1/f/(a^2+
b^2)*B*arctan(tan(f*x+e))*a*d+1/f/(a^2+b^2)*B*arctan(tan(f*x+e))*b*c-1/f/(a
^2+b^2)*C*arctan(tan(f*x+e))*a*c-1/f/(a^2+b^2)*C*arctan(tan(f*x+e))*b*d
```

maxima [A] time = 0.44, size = 183, normalized size = 1.17

$$\frac{\frac{2\,Cd\tan\left(fx+e\right)}{b}+\frac{2\,(((A-C)a+Bb)c-(Ba-(A-C)b)d)\left(fx+e\right)}{a^2+b^2}+\frac{2\left(\left(Ca^2b-Bab^2+Ab^3\right)c-\left(Ca^3-Ba^2b+Aab^2\right)d\right)\log\left(b\tan\left(fx+e\right)+a\right)}{a^2b^2+b^4}+((Ba-(A-C)}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*d*tan(f*x + e)/b + 2*(((A - C)*a + B*b)*c - (B*a - (A - C)*b)*d)*(
f*x + e)/(a^2 + b^2) + 2*((C*a^2*b - B*a*b^2 + A*b^3)*c - (C*a^3 - B*a^2*b
+ A*a*b^2)*d)*log(b*tan(f*x + e) + a)/(a^2*b^2 + b^4) + ((B*a - (A - C)*b)*
c + ((A - C)*a + B*b)*d)*log(tan(f*x + e)^2 + 1)/(a^2 + b^2))/f
```

mupad [B] time = 10.13, size = 186, normalized size = 1.19

$$\frac{\ln\left(\tan\left(e+f\,x\right)-\mathrm{i}\right)\,\left(A\,d+B\,c-C\,d-A\,c\,1\mathrm{i}+B\,d\,1\mathrm{i}+C\,c\,1\mathrm{i}\right)}{2\,f\,\left(a+b\,1\mathrm{i}\right)}+\frac{\ln\left(\tan\left(e+f\,x\right)+1\mathrm{i}\right)\,\left(B\,d+A\,d\,1\mathrm{i}+B\,c\,1\mathrm{i}-\right.}{2\,f\,\left(b+a\,1\mathrm{i}\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b*t
an(e + f*x)),x)
```

```
[Out] (log(tan(e + f*x) - 1i)*(A*d - A*c*1i + B*c + B*d*1i + C*c*1i - C*d))/(2*f*
(a + b*1i)) + (log(tan(e + f*x) + 1i)*(A*d*1i - A*c + B*c*1i + B*d + C*c -
C*d*1i))/(2*f*(a*1i + b)) - (log(a + b*tan(e + f*x))*(b^2*(A*a*d + B*a*c) -
 b*(B*a^2*d + C*a^2*c) - A*b^3*c + C*a^3*d))/(f*(b^4 + a^2*b^2)) + (C*d*tan
(e + f*x))/(b*f)
```

sympy [A] time = 2.37, size = 2429, normalized size = 15.57

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e)
),x)
```

```
[Out] Piecewise((zoo*x*(c + d*tan(e))*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(a,
0) & Eq(b, 0) & Eq(f, 0)), (-I*A*c*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) +
2*I*b*f) - A*c*f*x/(-2*b*f*tan(e + f*x) + 2*I*b*f) - I*A*c/(-2*b*f*tan(e +
f*x) + 2*I*b*f) - A*d*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) + I*
A*d*f*x/(-2*b*f*tan(e + f*x) + 2*I*b*f) + A*d/(-2*b*f*tan(e + f*x) + 2*I*b*
f) - B*c*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) + I*B*c*f*x/(-2*b
*f*tan(e + f*x) + 2*I*b*f) + B*c/(-2*b*f*tan(e + f*x) + 2*I*b*f) - I*B*d*f*
x*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) - B*d*f*x/(-2*b*f*tan(e + f*
x) + 2*I*b*f) - B*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e + f
*x) + 2*I*b*f) + I*B*d*log(tan(e + f*x)**2 + 1)/(-2*b*f*tan(e + f*x) + 2*I*
b*f) + I*B*d/(-2*b*f*tan(e + f*x) + 2*I*b*f) - I*C*c*f*x*tan(e + f*x)/(-2*b
*f*tan(e + f*x) + 2*I*b*f) - C*c*f*x/(-2*b*f*tan(e + f*x) + 2*I*b*f) - C*c*
log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) + I*C
*c*log(tan(e + f*x)**2 + 1)/(-2*b*f*tan(e + f*x) + 2*I*b*f) + I*C*c/(-2*b*f
*tan(e + f*x) + 2*I*b*f) + 3*C*d*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*
I*b*f) - 3*I*C*d*f*x/(-2*b*f*tan(e + f*x) + 2*I*b*f) - I*C*d*log(tan(e + f*
x)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) - C*d*log(tan(e + f
*x)**2 + 1)/(-2*b*f*tan(e + f*x) + 2*I*b*f) - 2*C*d*tan(e + f*x)**2/(-2*b*f
*tan(e + f*x) + 2*I*b*f) - 3*C*d/(-2*b*f*tan(e + f*x) + 2*I*b*f), Eq(a, -I*
b)), (I*A*c*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - A*c*f*x/(-2*
b*f*tan(e + f*x) - 2*I*b*f) + I*A*c/(-2*b*f*tan(e + f*x) - 2*I*b*f) - A*d*f
*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - I*A*d*f*x/(-2*b*f*tan(e +
 f*x) - 2*I*b*f) + A*d/(-2*b*f*tan(e + f*x) - 2*I*b*f) - B*c*f*x*tan(e + f*
x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - I*B*c*f*x/(-2*b*f*tan(e + f*x) - 2*I*b
*f) + B*c/(-2*b*f*tan(e + f*x) - 2*I*b*f) + I*B*d*f*x*tan(e + f*x)/(-2*b*f*
tan(e + f*x) - 2*I*b*f) - B*d*f*x/(-2*b*f*tan(e + f*x) - 2*I*b*f) - B*d*log
(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - I*B*d*
log(tan(e + f*x)**2 + 1)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - I*B*d/(-2*b*f*ta
n(e + f*x) - 2*I*b*f) + I*C*c*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b
*f) - C*c*f*x/(-2*b*f*tan(e + f*x) - 2*I*b*f) - C*c*log(tan(e + f*x)**2 + 1
)*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - I*C*c*log(tan(e + f*x)**2
+ 1)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - I*C*c/(-2*b*f*tan(e + f*x) - 2*I*b*f
) + 3*C*d*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) + 3*I*C*d*f*x/(-
2*b*f*tan(e + f*x) - 2*I*b*f) + I*C*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)
/(-2*b*f*tan(e + f*x) - 2*I*b*f) - C*d*log(tan(e + f*x)**2 + 1)/(-2*b*f*tan
(e + f*x) - 2*I*b*f) - 2*C*d*tan(e + f*x)**2/(-2*b*f*tan(e + f*x) - 2*I*b*f
```

```
) - 3*C*d/(-2*b*f*tan(e + f*x) - 2*I*b*f), Eq(a, I*b)), ((A*c*x + A*d*log(t
an(e + f*x)**2 + 1)/(2*f) + B*c*log(tan(e + f*x)**2 + 1)/(2*f) - B*d*x + B*
d*tan(e + f*x)/f - C*c*x + C*c*tan(e + f*x)/f - C*d*log(tan(e + f*x)**2 + 1
)/(2*f) + C*d*tan(e + f*x)**2/(2*f))/a, Eq(b, 0)), (x*(c + d*tan(e))*(A + B
*tan(e) + C*tan(e)**2)/(a + b*tan(e)), Eq(f, 0)), (2*A*a*b**2*c*f*x/(2*a**2
*b**2*f + 2*b**4*f) - 2*A*a*b**2*d*log(a/b + tan(e + f*x))/(2*a**2*b**2*f +
 2*b**4*f) + A*a*b**2*d*log(tan(e + f*x)**2 + 1)/(2*a**2*b**2*f + 2*b**4*f)
 + 2*A*b**3*c*log(a/b + tan(e + f*x))/(2*a**2*b**2*f + 2*b**4*f) - A*b**3*c
*log(tan(e + f*x)**2 + 1)/(2*a**2*b**2*f + 2*b**4*f) + 2*A*b**3*d*f*x/(2*a*
*2*b**2*f + 2*b**4*f) + 2*B*a**2*b*d*log(a/b + tan(e + f*x))/(2*a**2*b**2*f
 + 2*b**4*f) - 2*B*a*b**2*c*log(a/b + tan(e + f*x))/(2*a**2*b**2*f + 2*b**4
*f) + B*a*b**2*c*log(tan(e + f*x)**2 + 1)/(2*a**2*b**2*f + 2*b**4*f) - 2*B*
a*b**2*d*f*x/(2*a**2*b**2*f + 2*b**4*f) + 2*B*b**3*c*f*x/(2*a**2*b**2*f + 2
*b**4*f) + B*b**3*d*log(tan(e + f*x)**2 + 1)/(2*a**2*b**2*f + 2*b**4*f) - 2
*C*a**3*d*log(a/b + tan(e + f*x))/(2*a**2*b**2*f + 2*b**4*f) + 2*C*a**2*b*c
*log(a/b + tan(e + f*x))/(2*a**2*b**2*f + 2*b**4*f) + 2*C*a**2*b*d*tan(e +
f*x)/(2*a**2*b**2*f + 2*b**4*f) - 2*C*a*b**2*c*f*x/(2*a**2*b**2*f + 2*b**4*
f) - C*a*b**2*d*log(tan(e + f*x)**2 + 1)/(2*a**2*b**2*f + 2*b**4*f) + C*b**
3*c*log(tan(e + f*x)**2 + 1)/(2*a**2*b**2*f + 2*b**4*f) - 2*C*b**3*d*f*x/(2
*a**2*b**2*f + 2*b**4*f) + 2*C*b**3*d*tan(e + f*x)/(2*a**2*b**2*f + 2*b**4*
f), True))
```

3.55 $\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$

Optimal. Leaf size=265

$$-\frac{(bc-ad)\left(Ab^2-a(bB-aC)\right)}{b^2f\left(a^2+b^2\right)(a+b\tan(e+fx))}+\frac{\log(\cos(e+fx))\left(-\left(a^2(d(A-C)+Bc)\right)+2ab(Ac-Bd-cC)+b^2(d(A-C)\right.}{f\left(a^2+b^2\right)^2}$$

```
[Out] (a^2*(A*c-B*d-C*c)-b^2*(A*c-B*d-C*c)+2*a*b*(B*c+(A-C)*d))*x/(a^2+b^2)^2+(2*
a*b*(A*c-B*d-C*c)-a^2*(B*c+(A-C)*d)+b^2*(B*c+(A-C)*d))*ln(cos(f*x+e))/(a^2+
b^2)^2/f+(a^4*C*d+b^4*(A*d+B*c)+2*a*b^3*(A*c-B*d-C*c)-a^2*b^2*(B*c+(A-3*C)*
d))*ln(a+b*tan(f*x+e))/b^2/(a^2+b^2)^2/f-(A*b^2-a*(B*b-C*a))*(-a*d+b*c)/b^2
/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] time = 0.47, antiderivative size = 265, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3635, 3626, 3617, 31, 3475}

$$-\frac{(bc-ad)\left(Ab^2-a(bB-aC)\right)}{b^2f\left(a^2+b^2\right)(a+b\tan(e+fx))}+\frac{\left(-a^2b^2(d(A-3C)+Bc)+a^4Cd+2ab^3(Ac-Bd-cC)+b^4(Ad+Bc)\right)\log(a}{b^2f\left(a^2+b^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b*T
an[e + f*x])^2,x]
```

```
[Out] ((a^2*(A*c - c*C - B*d) - b^2*(A*c - c*C - B*d) + 2*a*b*(B*c + (A - C)*d))*
x)/(a^2 + b^2)^2 + ((2*a*b*(A*c - c*C - B*d) - a^2*(B*c + (A - C)*d) + b^2*
(B*c + (A - C)*d))*Log[Cos[e + f*x]])/((a^2 + b^2)^2*f) + ((a^4*C*d + b^4*(
B*c + A*d) + 2*a*b^3*(A*c - c*C - B*d) - a^2*b^2*(B*c + (A - 3*C)*d))*Log[a
 + b*Tan[e + f*x]])/(b^2*(a^2 + b^2)^2*f) - ((A*b^2 - a*(b*B - a*C))*(b*c -
 a*d))/(b^2*(a^2 + b^2)*f*(a + b*Tan[e + f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)(bc-ad)}{b^2\left(a^2+b^2\right)f(a+b\tan(e+fx))}+\frac{\int \frac{a^2Cd+\cdots}{\cdots}}{\cdots}$$

$$= \frac{\left(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)+2ab\cdots\right.}{\left(a^2+b^2\right)^2}$$

$$= \frac{\left(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)+2ab\cdots\right.}{\left(a^2+b^2\right)^2}$$

$$= \frac{\left(a^2(Ac-cC-Bd)-b^2(Ac-cC-Bd)+2ab\cdots\right.}{\left(a^2+b^2\right)^2}$$

Mathematica [C] time = 6.86, size = 589, normalized size = 2.22

$$-2ia\tan^{-1}(\tan(e+fx))(a+b\tan(e+fx))\left(a^4Cd-a^2b^2(d(A-3C)+Bc)+2ab^3(Ac-Bd-cC)+b^4(Ad+Bc)\right)$$

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^2,x]
```

```
[Out] (a^2*(2*(a + I*b)^2*(A*b^2*(c - I*d) + I*a^2*C*d + 2*a*b*C*d + b^2*((-I)*B*
c - c*C - B*d))*(e + f*x) - 2*(a^2 + b^2)^2*C*d*Log[Cos[e + f*x]] + (a^4*C*
d + b^4*(B*c + A*d) + 2*a*b^3*(A*c - c*C - B*d) - a^2*b^2*(B*c + (A - 3*C)*
d))*Log[(a*Cos[e + f*x] + b*Sin[e + f*x])^2]) + b*(2*(a + I*b)*((-I)*A*b^4*
c + I*a^4*C*d*(I + e + f*x) + a*b^3*(A*c*(1 + I*e + I*f*x) - I*c*C*(e + f*x
) - I*B*d*(e + f*x) + B*c*(I + e + f*x) + A*d*(I + e + f*x)) - I*a^2*b^2*(I
*A*c*(e + f*x) - 2*C*d*(e + f*x) + B*c*(-I + e + f*x) + A*d*(-I + e + f*x)
- I*c*C*(I + e + f*x) - I*B*d*(I + e + f*x)) + a^3*b*(c*C + d*(B + C*(I + e
 + f*x)))) - 2*a*(a^2 + b^2)^2*C*d*Log[Cos[e + f*x]] + a*(a^4*C*d + b^4*(B*
c + A*d) + 2*a*b^3*(A*c - c*C - B*d) - a^2*b^2*(B*c + (A - 3*C)*d))*Log[(a*
Cos[e + f*x] + b*Sin[e + f*x])^2])*Tan[e + f*x] - (2*I)*a*(a^4*C*d + b^4*(B
*c + A*d) + 2*a*b^3*(A*c - c*C - B*d) - a^2*b^2*(B*c + (A - 3*C)*d))*ArcTan
[Tan[e + f*x]]*(a + b*Tan[e + f*x]))/(2*a*b^2*(a^2 + b^2)^2*f*(a + b*Tan[e
+ f*x]))
```

fricas [B] time = 1.34, size = 556, normalized size = 2.10

$$2\left(\left((A-C)a^3b^2+2\,Ba^2b^3-(A-C)ab^4\right)c-\left(Ba^3b^2-2\,(A-C)a^2b^3-Bab^4\right)d\right)fx-2\left(Ca^2b^3-Bab^4+Ab^5\right)c+2$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a^3*b^2 + 2*B*a^2*b^3 - (A - C)*a*b^4)*c - (B*a^3*b^2 - 2*
(A - C)*a^2*b^3 - B*a*b^4)*d)*f*x - 2*(C*a^2*b^3 - B*a*b^4 + A*b^5)*c + 2*(
C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*d - ((B*a^3*b^2 - 2*(A - C)*a^2*b^3 - B*a*
b^4)*c - (C*a^5 - (A - 3*C)*a^3*b^2 - 2*B*a^2*b^3 + A*a*b^4)*d + ((B*a^2*b^
3 - 2*(A - C)*a*b^4 - B*b^5)*c - (C*a^4*b - (A - 3*C)*a^2*b^3 - 2*B*a*b^4 +
 A*b^5)*d)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2
)/(tan(f*x + e)^2 + 1)) - ((C*a^4*b + 2*C*a^2*b^3 + C*b^5)*d*tan(f*x + e) +
 (C*a^5 + 2*C*a^3*b^2 + C*a*b^4)*d)*log(1/(tan(f*x + e)^2 + 1)) + 2*(((A -
```

```
C)*a^2*b^3 + 2*B*a*b^4 - (A - C)*b^5)*c - (B*a^2*b^3 - 2*(A - C)*a*b^4 - B
*b^5)*d)*f*x + (C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c - (C*a^4*b - B*a^3*b^2 +
A*a^2*b^3)*d)*tan(f*x + e))/((a^4*b^3 + 2*a^2*b^5 + b^7)*f*tan(f*x + e) +
(a^5*b^2 + 2*a^3*b^4 + a*b^6)*f)
```

giac [B] time = 2.35, size = 531, normalized size = 2.00

$$\frac{2\left(Aa^2c-Ca^2c+2\,Babc-Ab^2c+Cb^2c-Ba^2d+2\,Aabd-2\,Cabd+Bb^2d\right)(fx+e)}{a^4+2\,a^2b^2+b^4}+\frac{\left(Ba^2c-2\,Aabc+2\,Cabc-Bb^2c+Aa^2d-Ca^2d+2\,Babd-Ab^2d+Cb^2d\right)\log\left(\mathrm{ta}\right.}{a^4+2\,a^2b^2+b^4}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c - C*a^2*c + 2*B*a*b*c - A*b^2*c + C*b^2*c - B*a^2*d + 2*A*a
*b*d - 2*C*a*b*d + B*b^2*d)*(f*x + e)/(a^4 + 2*a^2*b^2 + b^4) + (B*a^2*c -
2*A*a*b*c + 2*C*a*b*c - B*b^2*c + A*a^2*d - C*a^2*d + 2*B*a*b*d - A*b^2*d +
C*b^2*d)*log(tan(f*x + e)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2*(B*a^2*b^2*c
- 2*A*a*b^3*c + 2*C*a*b^3*c - B*b^4*c - C*a^4*d + A*a^2*b^2*d - 3*C*a^2*b^2
*d + 2*B*a*b^3*d - A*b^4*d)*log(abs(b*tan(f*x + e) + a))/(a^4*b^2 + 2*a^2*b
^4 + b^6) + 2*(B*a^2*b^2*c*tan(f*x + e) - 2*A*a*b^3*c*tan(f*x + e) + 2*C*a*
b^3*c*tan(f*x + e) - B*b^4*c*tan(f*x + e) - C*a^4*d*tan(f*x + e) + A*a^2*b^
2*d*tan(f*x + e) - 3*C*a^2*b^2*d*tan(f*x + e) + 2*B*a*b^3*d*tan(f*x + e) -
A*b^4*d*tan(f*x + e) - C*a^4*c + 2*B*a^3*b*c - 3*A*a^2*b^2*c + C*a^2*b^2*c
- A*b^4*c - B*a^4*d + 2*A*a^3*b*d - 2*C*a^3*b*d + B*a^2*b^2*d)/((a^4*b + 2*
a^2*b^3 + b^5)*(b*tan(f*x + e) + a)))/f
```

maple [B] time = 0.30, size = 948, normalized size = 3.58

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2,x)
```

```
[Out] 1/f/(a^2+b^2)^2*A*arctan(tan(f*x+e))*a^2*c-1/f/(a^2+b^2)^2*A*arctan(tan(f*x
+e))*b^2*c-1/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*A*b^2*d+1/2/f/(a^2+b^2)^2*l
n(1+tan(f*x+e)^2)*A*a^2*d-1/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*C*a^2*d-1/f/
(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*a^2*d-2/f/(a^2+b^2)^2*b*ln(a+b*tan(f*x+e))
*B*a*d+1/f/(a^2+b^2)^2/b^2*ln(a+b*tan(f*x+e))*a^4*C*d-1/f/(a^2+b^2)^2*ln(1+
tan(f*x+e)^2)*A*a*b*c+1/f/(a^2+b^2)/(a+b*tan(f*x+e))*A*a*d+3/f/(a^2+b^2)^2*
ln(a+b*tan(f*x+e))*C*a^2*d+1/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*C*a*b*c+2/f/(
a^2+b^2)^2*A*arctan(tan(f*x+e))*a*b*d+2/f/(a^2+b^2)^2*B*arctan(tan(f*x+e))*
a*b*c-2/f/(a^2+b^2)^2*C*arctan(tan(f*x+e))*a*b*d-1/f/b/(a^2+b^2)/(a+b*tan(f
```

```
*x+e))*B*a^2*d+1/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*B*a*b*d+1/f/b^2/(a^2+b^2)
/(a+b*tan(f*x+e))*a^3*C*d-1/f/b/(a^2+b^2)/(a+b*tan(f*x+e))*C*a^2*c+2/f/(a^2
+b^2)^2*b*ln(a+b*tan(f*x+e))*A*a*c-2/f/(a^2+b^2)^2*b*ln(a+b*tan(f*x+e))*C*a
*c-1/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*B*b^2*c+1/2/f/(a^2+b^2)^2*ln(1+tan(
f*x+e)^2)*B*a^2*c-1/f/(a^2+b^2)^2*B*arctan(tan(f*x+e))*a^2*d+1/f/(a^2+b^2)^
2*B*arctan(tan(f*x+e))*b^2*d+1/f/(a^2+b^2)^2*b^2*ln(a+b*tan(f*x+e))*B*c+1/2
/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*C*b^2*d-1/f/(a^2+b^2)^2*C*arctan(tan(f*x+
e))*a^2*c+1/f/(a^2+b^2)^2*C*arctan(tan(f*x+e))*b^2*c-1/f/(a^2+b^2)^2*ln(a+b
*tan(f*x+e))*B*a^2*c+1/f/(a^2+b^2)^2*b^2*ln(a+b*tan(f*x+e))*A*d-1/f*b/(a^2+
b^2)/(a+b*tan(f*x+e))*A*c+1/f/(a^2+b^2)/(a+b*tan(f*x+e))*B*a*c
```

maxima [A] time = 0.46, size = 338, normalized size = 1.28

$$\frac{\frac{2\left(\left((A-C)a^2+2\,Bab-(A-C)b^2\right)c-\left(Ba^2-2\,(A-C)ab-Bb^2\right)d\right)(fx+e)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(\left(Ba^2b^2-2\,(A-C)ab^3-Bb^4\right)c-\left(Ca^4-(A-3\,C)a^2b^2-2\,Bab^3+Ab^4\right)d\right)\log\left(b\tan\left(fx\right.\right.}{a^4b^2+2\,a^2b^4+b^6}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c - (B*a^2 - 2*(A - C)*a*b -
B*b^2)*d)*(f*x + e)/(a^4 + 2*a^2*b^2 + b^4) - 2*((B*a^2*b^2 - 2*(A - C)*a*b
^3 - B*b^4)*c - (C*a^4 - (A - 3*C)*a^2*b^2 - 2*B*a*b^3 + A*b^4)*d)*log(b*ta
n(f*x + e) + a)/(a^4*b^2 + 2*a^2*b^4 + b^6) + ((B*a^2 - 2*(A - C)*a*b - B*b
^2)*c + ((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*d)*log(tan(f*x + e)^2 + 1)/(a
^4 + 2*a^2*b^2 + b^4) - 2*((C*a^2*b - B*a*b^2 + A*b^3)*c - (C*a^3 - B*a^2*b
 + A*a*b^2)*d)/(a^3*b^2 + a*b^4 + (a^2*b^3 + b^5)*tan(f*x + e)))/f
```

mupad [B] time = 21.14, size = 1875, normalized size = 7.08

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b*t
an(e + f*x))^2,x)
```

```
[Out] (log(a + b*tan(e + f*x))*(b^4*(A*d + B*c) - b^3*(2*B*a*d - 2*A*a*c + 2*C*a*
c) - b^2*(A*a^2*d + B*a^2*c - 3*C*a^2*d) + C*a^4*d))/(f*(b^6 + 2*a^2*b^4 +
a^4*b^2)) - (log((A*B*b^4*d^2 - A*B*b^4*c^2 + B*C*a^4*d^2 + B*C*b^4*c^2 - A
^2*b^4*c*d + B^2*b^4*c*d + C^2*a^4*c*d - A^2*a*b^3*c^2 + A^2*a*b^3*d^2 + B^
2*a*b^3*c^2 - B^2*a*b^3*d^2 - C^2*a*b^3*c^2 + C^2*a*b^3*d^2 + A*B*a^2*b^2*c
^2 - A*B*a^2*b^2*d^2 - B*C*a^2*b^2*c^2 + 3*B*C*a^2*b^2*d^2 + A^2*a^2*b^2*c*
d - B^2*a^2*b^2*c*d + 3*C^2*a^2*b^2*c*d - A*C*a^4*c*d + A*C*b^4*c*d + 2*A*C
```

```
*a*b^3*c^2 - 2*A*C*a*b^3*d^2 - 4*A*C*a^2*b^2*c*d + 4*A*B*a*b^3*c*d - 4*B*C*
a*b^3*c*d)/(b*(a^2 + b^2)^2) + (tan(e + f*x)*(A^2*b^4*c^2 + B^2*b^4*d^2 + C
^2*a^4*d^2 + C^2*b^4*c^2 + C^2*b^4*d^2 + A^2*a^2*b^2*d^2 + B^2*a^2*b^2*c^2
+ 3*C^2*a^2*b^2*d^2 - A*C*a^4*d^2 - 2*A*C*b^4*c^2 - A*C*b^4*d^2 - 4*A*C*a^2
*b^2*d^2 - 2*A*B*b^4*c*d - B*C*a^4*c*d + B*C*b^4*c*d - 2*A*B*a*b^3*c^2 + 2*
A*B*a*b^3*d^2 + 2*B*C*a*b^3*c^2 - 2*B*C*a*b^3*d^2 - 2*A^2*a*b^3*c*d + 2*B^2
*a*b^3*c*d - 2*C^2*a*b^3*c*d + 2*A*B*a^2*b^2*c*d - 4*B*C*a^2*b^2*c*d + 4*A*
C*a*b^3*c*d))/(b*(a^2 + b^2)^2) + ((c + d*1i)*(A + B*1i - C)*(A*b*c - B*b*d
 - 4*C*a*d - C*b*c + (tan(e + f*x)*(3*A*b^4*d + 3*B*b^4*c + 2*C*a^4*d - 5*C
*b^4*d + 4*A*a*b^3*c - 4*B*a*b^3*d - 4*C*a*b^3*c - A*a^2*b^2*d - B*a^2*b^2*
c + C*a^2*b^2*d))/(b*(a^2 + b^2)) + (b*(c + d*1i)*(4*a*b - a^2*tan(e + f*x)
 + 3*b^2*tan(e + f*x))*(A + B*1i - C)*1i)/(a*1i - b)^2)*1i)/(2*(a*1i - b)^2
))*(A*c + A*d*1i + B*c*1i - B*d - C*c - C*d*1i))/(2*f*(2*a*b - a^2*1i + b^2
*1i)) - (log((A*B*b^4*d^2 - A*B*b^4*c^2 + B*C*a^4*d^2 + B*C*b^4*c^2 - A^2*b
^4*c*d + B^2*b^4*c*d + C^2*a^4*c*d - A^2*a*b^3*c^2 + A^2*a*b^3*d^2 + B^2*a*
b^3*c^2 - B^2*a*b^3*d^2 - C^2*a*b^3*c^2 + C^2*a*b^3*d^2 + A*B*a^2*b^2*c^2 -
 A*B*a^2*b^2*d^2 - B*C*a^2*b^2*c^2 + 3*B*C*a^2*b^2*d^2 + A^2*a^2*b^2*c*d -
B^2*a^2*b^2*c*d + 3*C^2*a^2*b^2*c*d - A*C*a^4*c*d + A*C*b^4*c*d + 2*A*C*a*b
^3*c^2 - 2*A*C*a*b^3*d^2 - 4*A*C*a^2*b^2*c*d + 4*A*B*a*b^3*c*d - 4*B*C*a*b^
3*c*d)/(b*(a^2 + b^2)^2) + (tan(e + f*x)*(A^2*b^4*c^2 + B^2*b^4*d^2 + C^2*a
^4*d^2 + C^2*b^4*c^2 + C^2*b^4*d^2 + A^2*a^2*b^2*d^2 + B^2*a^2*b^2*c^2 + 3*
C^2*a^2*b^2*d^2 - A*C*a^4*d^2 - 2*A*C*b^4*c^2 - A*C*b^4*d^2 - 4*A*C*a^2*b^2
*d^2 - 2*A*B*b^4*c*d - B*C*a^4*c*d + B*C*b^4*c*d - 2*A*B*a*b^3*c^2 + 2*A*B*
a*b^3*d^2 + 2*B*C*a*b^3*c^2 - 2*B*C*a*b^3*d^2 - 2*A^2*a*b^3*c*d + 2*B^2*a*b
^3*c*d - 2*C^2*a*b^3*c*d + 2*A*B*a^2*b^2*c*d - 4*B*C*a^2*b^2*c*d + 4*A*C*a*
b^3*c*d))/(b*(a^2 + b^2)^2) + ((c*1i + d)*(B*1i - A + C)*(A*b*c - B*b*d - 4
*C*a*d - C*b*c + (tan(e + f*x)*(3*A*b^4*d + 3*B*b^4*c + 2*C*a^4*d - 5*C*b^4
*d + 4*A*a*b^3*c - 4*B*a*b^3*d - 4*C*a*b^3*c - A*a^2*b^2*d - B*a^2*b^2*c +
C*a^2*b^2*d))/(b*(a^2 + b^2)) + (b*(c*1i + d)*(4*a*b - a^2*tan(e + f*x) + 3
*b^2*tan(e + f*x))*(B*1i - A + C))/(a*1i + b)^2))/(2*(a*1i + b)^2))*(A*c*1i
 + A*d + B*c - B*d*1i - C*c*1i - C*d))/(2*f*(a*b*2i - a^2 + b^2)) - (A*b^3*
c - C*a^3*d - A*a*b^2*d - B*a*b^2*c + B*a^2*b*d + C*a^2*b*c)/(b^2*f*(a^2 +
b^2)*(a + b*tan(e + f*x)))
```

sympy [A] time = 3.95, size = 9721, normalized size = 36.68

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e)
)**2,x)
```

```
[Out] Piecewise((zoo*x*(c + d*tan(e))*(A + B*tan(e) + C*tan(e)**2)/tan(e)**2, Eq(
a, 0) & Eq(b, 0) & Eq(f, 0)), (A*c*f*x*tan(e + f*x)**2/(-4*b**2*f*tan(e + f
*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 2*I*A*c*f*x*tan(e + f*x)/(-4
```

```
*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - A*c*f*x/(-4
*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + A*c*tan(e +
 f*x)/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 2*
I*A*c/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - I*
A*d*f*x*tan(e + f*x)**2/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x
) + 4*b**2*f) - 2*A*d*f*x*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**
2*f*tan(e + f*x) + 4*b**2*f) + I*A*d*f*x/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b
**2*f*tan(e + f*x) + 4*b**2*f) - I*A*d*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)
**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - I*B*c*f*x*tan(e + f*x)**2/(-4*b
**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 2*B*c*f*x*tan
(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f)
 + I*B*c*f*x/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f
) - I*B*c*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x)
 + 4*b**2*f) - B*d*f*x*tan(e + f*x)**2/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**
2*f*tan(e + f*x) + 4*b**2*f) + 2*I*B*d*f*x*tan(e + f*x)/(-4*b**2*f*tan(e +
f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + B*d*f*x/(-4*b**2*f*tan(e +
f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + 3*B*d*tan(e + f*x)/(-4*b**2
*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 2*I*B*d/(-4*b**2
*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - C*c*f*x*tan(e +
f*x)**2/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) +
2*I*C*c*f*x*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*
x) + 4*b**2*f) + C*c*f*x/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*
x) + 4*b**2*f) + 3*C*c*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f
*tan(e + f*x) + 4*b**2*f) - 2*I*C*c/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f
*tan(e + f*x) + 4*b**2*f) - 3*I*C*d*f*x*tan(e + f*x)**2/(-4*b**2*f*tan(e +
f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 6*C*d*f*x*tan(e + f*x)/(-4*
b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + 3*I*C*d*f*x/
(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 2*C*d*lo
g(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**
2*f*tan(e + f*x) + 4*b**2*f) + 4*I*C*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x
)/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + 2*C*d*
log(tan(e + f*x)**2 + 1)/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f*tan(e + f*
x) + 4*b**2*f) + 5*I*C*d*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2
*f*tan(e + f*x) + 4*b**2*f) + 4*C*d/(-4*b**2*f*tan(e + f*x)**2 + 8*I*b**2*f
*tan(e + f*x) + 4*b**2*f), Eq(a, -I*b)), (A*c*f*x*tan(e + f*x)**2/(-4*b**2*
f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + 2*I*A*c*f*x*tan(e
 + f*x)/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) -
A*c*f*x/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) +
A*c*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b
**2*f) + 2*I*A*c/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b
**2*f) + I*A*d*f*x*tan(e + f*x)**2/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*
tan(e + f*x) + 4*b**2*f) - 2*A*d*f*x*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**
2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - I*A*d*f*x/(-4*b**2*f*tan(e + f*x)
**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + I*A*d*tan(e + f*x)/(-4*b**2*f*t
an(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + I*B*c*f*x*tan(e + f*
```

```
x)**2/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 2*
B*c*f*x*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) +
 4*b**2*f) - I*B*c*f*x/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x)
 + 4*b**2*f) + I*B*c*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*t
an(e + f*x) + 4*b**2*f) - B*d*f*x*tan(e + f*x)**2/(-4*b**2*f*tan(e + f*x)**
2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 2*I*B*d*f*x*tan(e + f*x)/(-4*b**2
*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + B*d*f*x/(-4*b**2
*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + 3*B*d*tan(e + f*
x)/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + 2*I*B
*d/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - C*c*f
*x*tan(e + f*x)**2/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4
*b**2*f) - 2*I*C*c*f*x*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f
*tan(e + f*x) + 4*b**2*f) + C*c*f*x/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f
*tan(e + f*x) + 4*b**2*f) + 3*C*c*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 -
 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + 2*I*C*c/(-4*b**2*f*tan(e + f*x)**2 -
 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + 3*I*C*d*f*x*tan(e + f*x)**2/(-4*b**2
*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 6*C*d*f*x*tan(e
+ f*x)/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 3
*I*C*d*f*x/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f)
 - 2*C*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(-4*b**2*f*tan(e + f*x)**
2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) - 4*I*C*d*log(tan(e + f*x)**2 + 1)*
tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*
f) + 2*C*d*log(tan(e + f*x)**2 + 1)/(-4*b**2*f*tan(e + f*x)**2 - 8*I*b**2*f
*tan(e + f*x) + 4*b**2*f) - 5*I*C*d*tan(e + f*x)/(-4*b**2*f*tan(e + f*x)**2
 - 8*I*b**2*f*tan(e + f*x) + 4*b**2*f) + 4*C*d/(-4*b**2*f*tan(e + f*x)**2 -
 8*I*b**2*f*tan(e + f*x) + 4*b**2*f), Eq(a, I*b)), ((A*c*x + A*d*log(tan(e
+ f*x)**2 + 1)/(2*f) + B*c*log(tan(e + f*x)**2 + 1)/(2*f) - B*d*x + B*d*tan
(e + f*x)/f - C*c*x + C*c*tan(e + f*x)/f - C*d*log(tan(e + f*x)**2 + 1)/(2*
f) + C*d*tan(e + f*x)**2/(2*f))/a**2, Eq(b, 0)), (x*(c + d*tan(e))*(A + B*t
an(e) + C*tan(e)**2)/(a + b*tan(e))**2, Eq(f, 0)), (2*A*a**3*b**2*c*f*x/(2*
a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*ta
n(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*a**3*b**2*d*log(a/b
+ tan(e + f*x))/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f
 + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + A*a**
3*b**2*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*
x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan
(e + f*x)) + 2*A*a**3*b**2*d/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) +
4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e +
f*x)) + 2*A*a**2*b**3*c*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan
(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**
7*f*tan(e + f*x)) + 4*A*a**2*b**3*c*log(a/b + tan(e + f*x))/(2*a**5*b**2*f
+ 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) +
 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*a**2*b**3*c*log(tan(e + f*x)**2
+ 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b
**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*a**2*b**3*c/
```

```
(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f
*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 4*A*a**2*b**3*d*f*x/(
2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*
tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*a**2*b**3*d*log(a/
b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x)
+ 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e
+ f*x)) + A*a**2*b**3*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*
f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x)
 + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*a*b**4*c*f*x/(2*a**5*b**2*f +
2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2
*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 4*A*a*b**4*c*log(a/b + tan(e + f*x))*t
an(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4
*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*a*b**
4*c*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*ta
n(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b*
*7*f*tan(e + f*x)) + 4*A*a*b**4*d*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*
b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6
*f + 2*b**7*f*tan(e + f*x)) + 2*A*a*b**4*d*log(a/b + tan(e + f*x))/(2*a**5*
b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e +
 f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - A*a*b**4*d*log(tan(e + f*x)**
2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2
*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*A*a*b**4*d/(
2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*
tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*b**5*c*f*x*tan(e +
 f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*
b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*A*b**5*c/(2*a
**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan
(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*A*b**5*d*log(a/b + tan(
e + f*x))*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3
*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x))
- A*b**5*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b*
*3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f
 + 2*b**7*f*tan(e + f*x)) - 2*B*a**4*b*d/(2*a**5*b**2*f + 2*a**4*b**3*f*tan
(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**
7*f*tan(e + f*x)) - 2*B*a**3*b**2*c*log(a/b + tan(e + f*x))/(2*a**5*b**2*f
+ 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) +
 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + B*a**3*b**2*c*log(tan(e + f*x)**2 +
1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**
5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*a**3*b**2*c/(2
*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*t
an(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*B*a**3*b**2*d*f*x/(2*
a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*ta
n(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 4*B*a**2*b**3*c*f*x/(2*a
**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan
(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*B*a**2*b**3*c*log(a/b +
```

```
tan(e + f*x))*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4
*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f
*x)) + B*a**2*b**3*c*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f +
 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) +
2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*B*a**2*b**3*d*f*x*tan(e + f*x)/(2*a
**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan
(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 4*B*a**2*b**3*d*log(a/b +
 tan(e + f*x))/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f
+ 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*a*
*2*b**3*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f
*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*ta
n(e + f*x)) - 2*B*a**2*b**3*d/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) +
 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e +
 f*x)) + 4*B*a*b**4*c*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e
 + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*
f*tan(e + f*x)) + 2*B*a*b**4*c*log(a/b + tan(e + f*x))/(2*a**5*b**2*f + 2*a
**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*
b**6*f + 2*b**7*f*tan(e + f*x)) - B*a*b**4*c*log(tan(e + f*x)**2 + 1)/(2*a*
*5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(
e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*a*b**4*c/(2*a**5*b**2*
f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x)
 + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*a*b**4*d*f*x/(2*a**5*b**2*f +
2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2
*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 4*B*a*b**4*d*log(a/b + tan(e + f*x))*t
an(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4
*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*B*a*b**
4*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*ta
n(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b*
*7*f*tan(e + f*x)) + 2*B*b**5*c*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**
5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e
 + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - B*b**5*c*log(tan(e + f*x)**
2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b*
*4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2
*B*b**5*d*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*
a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*
x)) + 2*C*a**5*d*log(a/b + tan(e + f*x))/(2*a**5*b**2*f + 2*a**4*b**3*f*tan
(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**
7*f*tan(e + f*x)) + 2*C*a**5*d/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x)
+ 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e
+ f*x)) - 2*C*a**4*b*c/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3
*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x))
+ 2*C*a**4*b*d*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4
*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**
6*f + 2*b**7*f*tan(e + f*x)) - 2*C*a**3*b**2*c*f*x/(2*a**5*b**2*f + 2*a**4*
b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6
```

```
*f + 2*b**7*f*tan(e + f*x)) + 6*C*a**3*b**2*d*log(a/b + tan(e + f*x))/(2*a*
*5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(
e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - C*a**3*b**2*d*log(tan(e +
f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f +
4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*C*a**3
*b**2*d/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**
2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*C*a**2*b**3
*c*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b*
*4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 4
*C*a**2*b**3*c*log(a/b + tan(e + f*x))/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e
 + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*
f*tan(e + f*x)) + 2*C*a**2*b**3*c*log(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f +
 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) +
2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - 2*C*a**2*b**3*c/(2*a**5*b**2*f + 2*a*
*4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b
**6*f + 2*b**7*f*tan(e + f*x)) - 4*C*a**2*b**3*d*f*x/(2*a**5*b**2*f + 2*a**
4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b*
*6*f + 2*b**7*f*tan(e + f*x)) + 6*C*a**2*b**3*d*log(a/b + tan(e + f*x))*tan
(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a
**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) - C*a**2*b**3
*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan
(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**
7*f*tan(e + f*x)) + 2*C*a*b**4*c*f*x/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e +
 f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*
tan(e + f*x)) - 4*C*a*b**4*c*log(a/b + tan(e + f*x))*tan(e + f*x)/(2*a**5*b
**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e +
f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) + 2*C*a*b**4*c*log(tan(e + f*x)*
*2 + 1)*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b
**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)) -
4*C*a*b**4*d*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) +
 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e +
 f*x)) + C*a*b**4*d*log(tan(e + f*x)**2 + 1)/(2*a**5*b**2*f + 2*a**4*b**3*f
*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**6*f + 2
*b**7*f*tan(e + f*x)) + 2*C*b**5*c*f*x*tan(e + f*x)/(2*a**5*b**2*f + 2*a**4
*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**5*f*tan(e + f*x) + 2*a*b**
6*f + 2*b**7*f*tan(e + f*x)) + C*b**5*d*log(tan(e + f*x)**2 + 1)*tan(e + f*
x)/(2*a**5*b**2*f + 2*a**4*b**3*f*tan(e + f*x) + 4*a**3*b**4*f + 4*a**2*b**
5*f*tan(e + f*x) + 2*a*b**6*f + 2*b**7*f*tan(e + f*x)), True))
```

3.56 $\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$

Optimal. Leaf size=320

$$-\frac{(bc-ad)\left(Ab^2-a(bB-aC)\right)}{2b^2f\left(a^2+b^2\right)(a+b\tan(e+fx))^2}-\frac{a^4Cd-a^2b^2(d(A-3C)+Bc)+2ab^3(Ac-Bd-cC)+b^4(Ad+Bc)}{b^2f\left(a^2+b^2\right)^2(a+b\tan(e+fx))}+\frac{(-(}{}$$

```
[Out] (a^3*(A*c-B*d-C*c)-3*a*b^2*(A*c-B*d-C*c)+3*a^2*b*(B*c+(A-C)*d)-b^3*(B*c+(A-
C)*d))*x/(a^2+b^2)^3+(3*a^2*b*(A*c-B*d-C*c)-b^3*(A*c-B*d-C*c)-a^3*(B*c+(A-C
)*d)+3*a*b^2*(B*c+(A-C)*d))*ln(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^2)^3/f-1/2
*(A*b^2-a*(B*b-C*a))*(-a*d+b*c)/b^2/(a^2+b^2)/f/(a+b*tan(f*x+e))^2+(-a^4*C*
d-b^4*(A*d+B*c)-2*a*b^3*(A*c-B*d-C*c)+a^2*b^2*(B*c+(A-3*C)*d))/b^2/(a^2+b^2
)^2/f/(a+b*tan(f*x+e))
```

Rubi [A] time = 0.70, antiderivative size = 320, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.093, Rules used = {3635, 3628, 3531, 3530}

$$-\frac{(bc-ad)\left(Ab^2-a(bB-aC)\right)}{2b^2f\left(a^2+b^2\right)(a+b\tan(e+fx))^2}-\frac{-a^2b^2(d(A-3C)+Bc)+a^4Cd+2ab^3(Ac-Bd-cC)+b^4(Ad+Bc)}{b^2f\left(a^2+b^2\right)^2(a+b\tan(e+fx))}+\frac{(3}{}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b*T
an[e + f*x])^3,x]
```

```
[Out] ((a^3*(A*c - c*C - B*d) - 3*a*b^2*(A*c - c*C - B*d) + 3*a^2*b*(B*c + (A - C
)*d) - b^3*(B*c + (A - C)*d))*x)/(a^2 + b^2)^3 + ((3*a^2*b*(A*c - c*C - B*d
) - b^3*(A*c - c*C - B*d) - a^3*(B*c + (A - C)*d) + 3*a*b^2*(B*c + (A - C)*
d))*Log[a*Cos[e + f*x] + b*Sin[e + f*x]])/((a^2 + b^2)^3*f) - ((A*b^2 - a*(
b*B - a*C))*(b*c - a*d))/(2*b^2*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^2) - (a^
4*C*d + b^4*(B*c + A*d) + 2*a*b^3*(A*c - c*C - B*d) - a^2*b^2*(B*c + (A - 3
*C)*d))/(b^2*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)(bc-ad)}{2b^2\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}+\frac{\int \frac{a^2C\cdots}{\cdots}}{\cdots}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)(bc-ad)}{2b^2\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}-\frac{a^4Cd\cdots}{\cdots}$$

$$= \frac{\left(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)+3\cdots\right.}{\left(a^2+\cdots\right.}$$

$$= \frac{\left(a^3(Ac-cC-Bd)-3ab^2(Ac-cC-Bd)+3\cdots\right.}{\left(a^2+\cdots\right.}$$

Mathematica [C] time = 6.38, size = 379, normalized size = 1.18

$$-\frac{C(c+d\tan(e+fx))}{bf(a+b\tan(e+fx))^2}-\frac{-\frac{-aCd-bBd+bcC}{2bf(a+b\tan(e+fx))^2}+\frac{\frac{\left(2ab^2(d(A-C)+Bc)-2b^3(Ac-Bd-cC)\right)\left(-\frac{2ab}{\left(a^2+b^2\right)^2(a+b\tan(e+fx))}-\frac{b}{2\left(a^2+b^2\right)(a+b\tan(e+fx))^2}+\frac{b\left(3a^2\cdots\right.}{\cdots}\right.}{b}}{\cdots}}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^3,x]
```

```
[Out] -((C*(c + d*Tan[e + f*x]))/(b*f*(a + b*Tan[e + f*x])^2)) - (-1/2*(b*c*C - b
*B*d - a*C*d)/(b*f*(a + b*Tan[e + f*x])^2) + (((-2*b^3*(A*c - c*C - B*d) +
2*a*b^2*(B*c + (A - C)*d))*(-1/2*Log[I - Tan[e + f*x]]/(I*a - b)^3 + Log[I
+ Tan[e + f*x]]/(2*(I*a + b)^3) + (b*(3*a^2 - b^2)*Log[a + b*Tan[e + f*x]])
/(a^2 + b^2)^3 - b/(2*(a^2 + b^2)*(a + b*Tan[e + f*x])^2) - (2*a*b)/((a^2 +
 b^2)^2*(a + b*Tan[e + f*x]))))/b - 2*b*(B*c + (A - C)*d)*(((-1/2*I)*Log[I
- Tan[e + f*x]])/(a + I*b)^2 + ((I/2)*Log[I + Tan[e + f*x]])/(a - I*b)^2 +
(2*a*b*Log[a + b*Tan[e + f*x]])/(a^2 + b^2)^2 - b/((a^2 + b^2)*(a + b*Tan[e
 + f*x]))))/(2*b*f))/b
```

fricas [B] time = 2.06, size = 987, normalized size = 3.08

$$\frac{2\left(\left(\left((A-C)a^5+3\,Ba^4b-3\,(A-C)a^3b^2-Ba^2b^3\right)c-\left(Ba^5-3\,(A-C)a^4b-3\,Ba^3b^2+(A-C)a^2b^3\right)d\right)fx+\left(2\left(\left(\cdots\right.\right.\right.\right.}{\cdots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a^5 + 3*B*a^4*b - 3*(A - C)*a^3*b^2 - B*a^2*b^3)*c - (B*a^
5 - 3*(A - C)*a^4*b - 3*B*a^3*b^2 + (A - C)*a^2*b^3)*d)*f*x + (2*(((A - C)*
a^3*b^2 + 3*B*a^2*b^3 - 3*(A - C)*a*b^4 - B*b^5)*c - (B*a^3*b^2 - 3*(A - C)
*a^2*b^3 - 3*B*a*b^4 + (A - C)*b^5)*d)*f*x + (C*a^4*b - 3*B*a^3*b^2 + 5*(A
- C)*a^2*b^3 + 3*B*a*b^4 - A*b^5)*c + (C*a^5 + B*a^4*b - (3*A - 7*C)*a^3*b^
2 - 5*B*a^2*b^3 + 3*A*a*b^4)*d)*tan(f*x + e)^2 - (3*C*a^4*b - 5*B*a^3*b^2 +
 (7*A - 3*C)*a^2*b^3 + B*a*b^4 + A*b^5)*c + (C*a^5 - 3*B*a^4*b + 5*(A - C)*
a^3*b^2 + 3*B*a^2*b^3 - A*a*b^4)*d - (((B*a^3*b^2 - 3*(A - C)*a^2*b^3 - 3*B
*a*b^4 + (A - C)*b^5)*c + ((A - C)*a^3*b^2 + 3*B*a^2*b^3 - 3*(A - C)*a*b^4
- B*b^5)*d)*tan(f*x + e)^2 + (B*a^5 - 3*(A - C)*a^4*b - 3*B*a^3*b^2 + (A -
C)*a^2*b^3)*c + ((A - C)*a^5 + 3*B*a^4*b - 3*(A - C)*a^3*b^2 - B*a^2*b^3)*d
 + 2*((B*a^4*b - 3*(A - C)*a^3*b^2 - 3*B*a^2*b^3 + (A - C)*a*b^4)*c + ((A -
 C)*a^4*b + 3*B*a^3*b^2 - 3*(A - C)*a^2*b^3 - B*a*b^4)*d)*tan(f*x + e))*log
((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) + 2*
(2*(((A - C)*a^4*b + 3*B*a^3*b^2 - 3*(A - C)*a^2*b^3 - B*a*b^4)*c - (B*a^4*
b - 3*(A - C)*a^3*b^2 - 3*B*a^2*b^3 + (A - C)*a*b^4)*d)*f*x + (C*a^5 - 2*B*
a^4*b + 3*(A - C)*a^3*b^2 + 3*B*a^2*b^3 - (3*A - 2*C)*a*b^4 - B*b^5)*c + (B
*a^5 - (2*A - 3*C)*a^4*b - 3*B*a^3*b^2 + 3*(A - C)*a^2*b^3 + 2*B*a*b^4 - A*
b^5)*d)*tan(f*x + e))/((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*f*tan(f*x +
e)^2 + 2*(a^7*b + 3*a^5*b^3 + 3*a^3*b^5 + a*b^7)*f*tan(f*x + e) + (a^8 + 3*
a^6*b^2 + 3*a^4*b^4 + a^2*b^6)*f)
```

giac [B] time = 3.17, size = 1037, normalized size = 3.24

$$\frac{\frac{2\left(Aa^3c-Ca^3c+3\,Ba^2bc-3\,Aab^2c+3\,Cab^2c-Bb^3c-Ba^3d+3\,Aa^2bd-3\,Ca^2bd+3\,Bab^2d-Ab^3d+Cb^3d\right)(fx+e)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}+\frac{\left(Ba^3c-3\,Aa^2bc+3\,Ca^2bc-3\,Bab^2c+Ab^3c-\right.}{}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^3*c - C*a^3*c + 3*B*a^2*b*c - 3*A*a*b^2*c + 3*C*a*b^2*c - B*b^3
*c - B*a^3*d + 3*A*a^2*b*d - 3*C*a^2*b*d + 3*B*a*b^2*d - A*b^3*d + C*b^3*d)
*(f*x + e)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + (B*a^3*c - 3*A*a^2*b*c + 3
*C*a^2*b*c - 3*B*a*b^2*c + A*b^3*c - C*b^3*c + A*a^3*d - C*a^3*d + 3*B*a^2*
b*d - 3*A*a*b^2*d + 3*C*a*b^2*d - B*b^3*d)*log(tan(f*x + e)^2 + 1)/(a^6 + 3
*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(B*a^3*b*c - 3*A*a^2*b^2*c + 3*C*a^2*b^2*c
- 3*B*a*b^3*c + A*b^4*c - C*b^4*c + A*a^3*b*d - C*a^3*b*d + 3*B*a^2*b^2*d -
```

```
 3*A*a*b^3*d + 3*C*a*b^3*d - B*b^4*d)*log(abs(b*tan(f*x + e) + a))/(a^6*b +
 3*a^4*b^3 + 3*a^2*b^5 + b^7) + (3*B*a^3*b^4*c*tan(f*x + e)^2 - 9*A*a^2*b^5
*c*tan(f*x + e)^2 + 9*C*a^2*b^5*c*tan(f*x + e)^2 - 9*B*a*b^6*c*tan(f*x + e)
^2 + 3*A*b^7*c*tan(f*x + e)^2 - 3*C*b^7*c*tan(f*x + e)^2 + 3*A*a^3*b^4*d*ta
n(f*x + e)^2 - 3*C*a^3*b^4*d*tan(f*x + e)^2 + 9*B*a^2*b^5*d*tan(f*x + e)^2
- 9*A*a*b^6*d*tan(f*x + e)^2 + 9*C*a*b^6*d*tan(f*x + e)^2 - 3*B*b^7*d*tan(f
*x + e)^2 + 8*B*a^4*b^3*c*tan(f*x + e) - 22*A*a^3*b^4*c*tan(f*x + e) + 22*C
*a^3*b^4*c*tan(f*x + e) - 18*B*a^2*b^5*c*tan(f*x + e) + 2*A*a*b^6*c*tan(f*x
 + e) - 2*C*a*b^6*c*tan(f*x + e) - 2*B*b^7*c*tan(f*x + e) - 2*C*a^6*b*d*tan
(f*x + e) + 8*A*a^4*b^3*d*tan(f*x + e) - 14*C*a^4*b^3*d*tan(f*x + e) + 22*B
*a^3*b^4*d*tan(f*x + e) - 18*A*a^2*b^5*d*tan(f*x + e) + 12*C*a^2*b^5*d*tan(
f*x + e) - 2*B*a*b^6*d*tan(f*x + e) - 2*A*b^7*d*tan(f*x + e) - C*a^6*b*c +
6*B*a^5*b^2*c - 14*A*a^4*b^3*c + 11*C*a^4*b^3*c - 7*B*a^3*b^4*c - 3*A*a^2*b
^5*c - B*a*b^6*c - A*b^7*c - C*a^7*d - B*a^6*b*d + 6*A*a^5*b^2*d - 9*C*a^5*
b^2*d + 11*B*a^4*b^3*d - 7*A*a^3*b^4*d + 4*C*a^3*b^4*d - A*a*b^6*d)/((a^6*b
^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*(b*tan(f*x + e) + a)^2))/f
```

maple [B] time = 0.33, size = 1513, normalized size = 4.73

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3,x)
```

```
[Out] 1/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*b^3*c+1/2/f/(a^2+b^2)/(a+b*tan(f*x+e))
^2*A*a*d-3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A*a*b^2*d+3/2/f/(a^2+b^2)^3*l
n(1+tan(f*x+e)^2)*B*a^2*b*d-1/f/(a^2+b^2)^2/b^2/(a+b*tan(f*x+e))*a^4*C*d+2/
f/(a^2+b^2)^2*b/(a+b*tan(f*x+e))*C*a*c+3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)
*C*a*b^2*d-2/f/(a^2+b^2)^2*b/(a+b*tan(f*x+e))*A*a*c+3/f/(a^2+b^2)^3*B*arcta
n(tan(f*x+e))*a*b^2*d-3/f/(a^2+b^2)^3*C*arctan(tan(f*x+e))*a^2*b*d+3/f/(a^2
+b^2)^3*C*arctan(tan(f*x+e))*a*b^2*c+1/f/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a^2
*c+1/2/f/(a^2+b^2)/(a+b*tan(f*x+e))^2*B*a*c+1/f/(a^2+b^2)^3*ln(a+b*tan(f*x+
e))*B*b^3*d+1/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*a^3*C*d-3/f/(a^2+b^2)^3*ln(a
+b*tan(f*x+e))*C*a^2*b*c-1/2/f/b/(a^2+b^2)/(a+b*tan(f*x+e))^2*B*a^2*d+1/2/f
/b^2/(a^2+b^2)/(a+b*tan(f*x+e))^2*a^3*C*d-1/2/f/b/(a^2+b^2)/(a+b*tan(f*x+e)
)^2*C*a^2*c+3/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a*b^2*d+2/f/(a^2+b^2)^2*b/
(a+b*tan(f*x+e))*B*a*d-3/f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*a*b^2*c-3/2/f/(
a^2+b^2)^3*ln(1+tan(f*x+e)^2)*B*a*b^2*c-3/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*
B*a^2*b*d+3/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a*b^2*c+3/f/(a^2+b^2)^3*ln(a
+b*tan(f*x+e))*A*a^2*b*c-3/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*a*b^2*d+3/2/f
/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*a^2*b*c+3/f/(a^2+b^2)^3*B*arctan(tan(f*x+
e))*a^2*b*c+3/f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*a^2*b*d-3/2/f/(a^2+b^2)^3*
ln(1+tan(f*x+e)^2)*A*a^2*b*c-1/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a^3*d+1/2
/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*B*a^3*c-1/2/f*b/(a^2+b^2)/(a+b*tan(f*x+e)
)^2*A*c-1/f/(a^2+b^2)^2*b^2/(a+b*tan(f*x+e))*A*d-1/f/(a^2+b^2)^2*b^2/(a+b*t
```

```
an(f*x+e))*B*c+1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A*a^3*d-1/f/(a^2+b^2)^3
*ln(a+b*tan(f*x+e))*A*b^3*c-1/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^3*c+1/f/
(a^2+b^2)^2/(a+b*tan(f*x+e))*A*a^2*d+1/f/(a^2+b^2)^3*C*arctan(tan(f*x+e))*b
^3*d+1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A*b^3*c-1/2/f/(a^2+b^2)^3*ln(1+ta
n(f*x+e)^2)*B*b^3*d-3/f/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a^2*d-1/2/f/(a^2+b^2
)^3*ln(1+tan(f*x+e)^2)*a^3*C*d-1/f/(a^2+b^2)^3*C*arctan(tan(f*x+e))*a^3*c-1
/f/(a^2+b^2)^3*B*arctan(tan(f*x+e))*a^3*d-1/f/(a^2+b^2)^3*B*arctan(tan(f*x+
e))*b^3*c-1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*b^3*c+1/f/(a^2+b^2)^3*A*ar
ctan(tan(f*x+e))*a^3*c-1/f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*b^3*d
```

maxima [A] time = 0.64, size = 574, normalized size = 1.79

$$\frac{\frac{2\left(\left(\left((A-C)a^3+3\,Ba^2b-3\,(A-C)ab^2-Bb^3\right)c-\left(Ba^3-3\,(A-C)a^2b-3\,Bab^2+(A-C)b^3\right)d\right)\left(fx+e\right)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{2\left(\left(Ba^3-3\,(A-C)a^2b-3\,Bab^2+(A-C)b^3\right)c+\left((A-C)a^3+3\,B\cdots\right.\right.}{a^6+3\,a^4b^2+3\,a^2b\cdots}}{\cdots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c - (B*a^3 - 3*
(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*d)*(f*x + e)/(a^6 + 3*a^4*b^2 + 3*
a^2*b^4 + b^6) - 2*((B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c +
 ((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*d)*log(b*tan(f*x + e)
+ a)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) + ((B*a^3 - 3*(A - C)*a^2*b - 3*B*
a*b^2 + (A - C)*b^3)*c + ((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3
)*d)*log(tan(f*x + e)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - ((C*a^4*
b - 3*B*a^3*b^2 + (5*A - 3*C)*a^2*b^3 + B*a*b^4 + A*b^5)*c + (C*a^5 + B*a^4
*b - (3*A - 5*C)*a^3*b^2 - 3*B*a^2*b^3 + A*a*b^4)*d - 2*((B*a^2*b^3 - 2*(A
- C)*a*b^4 - B*b^5)*c - (C*a^4*b - (A - 3*C)*a^2*b^3 - 2*B*a*b^4 + A*b^5)*d
)*tan(f*x + e))/(a^6*b^2 + 2*a^4*b^4 + a^2*b^6 + (a^4*b^4 + 2*a^2*b^6 + b^8
)*tan(f*x + e)^2 + 2*(a^5*b^3 + 2*a^3*b^5 + a*b^7)*tan(f*x + e)))/f
```

mupad [B] time = 15.89, size = 502, normalized size = 1.57

$$-\frac{\frac{A\,b^5\,c+C\,a^5\,d+A\,a\,b^4\,d+B\,a\,b^4\,c+B\,a^4\,b\,d+C\,a^4\,b\,c+5\,A\,a^2\,b^3\,c-3\,A\,a^3\,b^2\,d-3\,B\,a^3\,b^2\,c-3\,B\,a^2\,b^3\,d-3\,C\,a^2\,b^3\,c+5\,C\,a^3\,b^2\,d}{2\,b^2\,\left(a^4+2\,a^2\,b^2+b^4\right)}+\frac{\tan\left(e+f\,x\right)\,\left(A\,b^4\,d+B\,b^4\cdots\right.}{\cdots}}{f\,\left(a^2+2\,a\,b\,\tan\left(e+f\,x\right)+b^2\,\tan\left(e+f\,x\right)^2\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b*t
an(e + f*x))^3,x)
```

```
[Out] - ((A*b^5*c + C*a^5*d + A*a*b^4*d + B*a*b^4*c + B*a^4*b*d + C*a^4*b*c + 5*A
*a^2*b^3*c - 3*A*a^3*b^2*d - 3*B*a^3*b^2*c - 3*B*a^2*b^3*d - 3*C*a^2*b^3*c
+ 5*C*a^3*b^2*d)/(2*b^2*(a^4 + b^4 + 2*a^2*b^2)) + (tan(e + f*x)*(A*b^4*d +
 B*b^4*c + C*a^4*d + 2*A*a*b^3*c - 2*B*a*b^3*d - 2*C*a*b^3*c - A*a^2*b^2*d
- B*a^2*b^2*c + 3*C*a^2*b^2*d))/(b*(a^4 + b^4 + 2*a^2*b^2)))/(f*(a^2 + b^2*
tan(e + f*x)^2 + 2*a*b*tan(e + f*x))) - (log(tan(e + f*x) + 1i)*(A*d*1i - A
*c + B*c*1i + B*d + C*c - C*d*1i))/(2*f*(a*b^2*3i - 3*a^2*b - a^3*1i + b^3)
) - (log(tan(e + f*x) - 1i)*(A*d - A*c*1i + B*c + B*d*1i + C*c*1i - C*d))/(
2*f*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i)) - (log(a + b*tan(e + f*x))*(a^3*(A
*d + B*c - C*d) - b^3*(B*d - A*c + C*c) + a^2*b*(3*B*d - 3*A*c + 3*C*c) - a
*b^2*(3*A*d + 3*B*c - 3*C*d)))/(f*(a^6 + b^6 + 3*a^2*b^4 + 3*a^4*b^2))
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e)
)**3,x)
```

```
[Out] Exception raised: AttributeError
```

3.57 $\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)\right.$

Optimal. Leaf size=661

$$\frac{\log(\cos(e+fx))\left(-\left(a^3\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\right)+3a^2b\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)+3ab^2\left(2cd(A-\right.\right.}{f}$$

```
[Out] -(a^3*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-3*a*b^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2
-d^2))+3*a^2*b*(2*c*(A-C)*d+B*(c^2-d^2))-b^3*(2*c*(A-C)*d+B*(c^2-d^2)))*x+(
3*a^2*b*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-b^3*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d
^2))-a^3*(2*c*(A-C)*d+B*(c^2-d^2))+3*a*b^2*(2*c*(A-C)*d+B*(c^2-d^2)))*ln(co
s(f*x+e))/f+d*(3*a^2*b*(A*c-B*d-C*c)-b^3*(A*c-B*d-C*c)+a^3*(B*c+(A-C)*d)-3*
a*b^2*(B*c+(A-C)*d))*tan(f*x+e)/f+1/2*(a^3*B-3*a*b^2*B+3*a^2*b*(A-C)-b^3*(A
-C))*(c+d*tan(f*x+e))^2/f+1/60*(4*a^3*C*d^3-3*a^2*b*d^2*(-16*B*d+3*C*c)+3*a
*b^2*d*(2*c^2*C-5*B*c*d+20*(A-C)*d^2)-b^3*(c^3*C-2*B*c^2*d+5*c*(A-C)*d^2+20
*B*d^3))*(c+d*tan(f*x+e))^3/d^4/f+1/20*b*(5*b*(A*b+B*a-C*b)*d^2+(-a*d+b*c)*
(-2*B*b*d-C*a*d+C*b*c))*tan(f*x+e)*(c+d*tan(f*x+e))^3/d^3/f-1/10*(-2*B*b*d-
C*a*d+C*b*c)*(a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3/d^2/f+1/6*C*(a+b*tan(f*x
+e))^3*(c+d*tan(f*x+e))^3/d/f
```

Rubi [A] time = 2.38, antiderivative size = 661, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3647, 3637, 3630, 3528, 3525, 3475}

$$\frac{(c+d\tan(e+fx))^3\left(-3a^2bd^2(3cC-16Bd)+4a^3Cd^3+3ab^2d\left(20d^2(A-C)-5Bcd+2c^2C\right)+b^3\left(-\left(5cd^2(A-C\right.\right.\right.}{60d^4f}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] -((a^3*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - 3*a*b^2*(c^2*C + 2*B*c*d
 - C*d^2 - A*(c^2 - d^2)) + 3*a^2*b*(2*c*(A - C)*d + B*(c^2 - d^2)) - b^3*(
2*c*(A - C)*d + B*(c^2 - d^2)))*x) + ((3*a^2*b*(c^2*C + 2*B*c*d - C*d^2 - A
*(c^2 - d^2)) - b^3*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - a^3*(2*c*(A
 - C)*d + B*(c^2 - d^2)) + 3*a*b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))*Log[Cos
[e + f*x]])/f + (d*(3*a^2*b*(A*c - c*C - B*d) - b^3*(A*c - c*C - B*d) + a^3
*(B*c + (A - C)*d) - 3*a*b^2*(B*c + (A - C)*d))*Tan[e + f*x])/f + ((a^3*B -
 3*a*b^2*B + 3*a^2*b*(A - C) - b^3*(A - C))*(c + d*Tan[e + f*x])^2)/(2*f) +
 ((4*a^3*C*d^3 - 3*a^2*b*d^2*(3*c*C - 16*B*d) + 3*a*b^2*d*(2*c^2*C - 5*B*c*
d + 20*(A - C)*d^2) - b^3*(c^3*C - 2*B*c^2*d + 5*c*(A - C)*d^2 + 20*B*d^3))
*(c + d*Tan[e + f*x])^3)/(60*d^4*f) + (b*(5*b*(A*b + a*B - b*C)*d^2 + (b*c
- a*d)*(b*c*C - 2*b*B*d - a*C*d))*Tan[e + f*x]*(c + d*Tan[e + f*x])^3)/(20*
```

```
d^3*f) - ((b*c*C - 2*b*B*d - a*C*d)*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f
*x])^3)/(10*d^2*f) + (C*(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^3)/(6*d
*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\frac{C(a+b\tan(e+fx))^3(c+}{6df}$$

$$=-\frac{(bcC-2bBd-aCd)(a+}{}$$

$$=\frac{b\left(5b(Ab+aB-bC)d^2+(\right.}{}$$

$$=\frac{\left(4a^3Cd^3-3a^2bd^2(3cC-1\right.}{}$$

$$=\frac{\left(a^3B-3ab^2B+3a^2b(A-\right.}{}$$

$$=-\left(a^3\left(c^2C+2Bcd-Cd^2-\right.\right.$$

$$=-\left(a^3\left(c^2C+2Bcd-Cd^2-\right.\right.$$

Mathematica [C] time = 6.65, size = 573, normalized size = 0.87

$$\frac{C(a+b\tan(e+fx))^3(c+d\tan(e+fx))^3}{6df}+\frac{-\frac{3(-aCd-2bBd+bcC)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df}+\frac{3b\tan(e+fx)(c+d\tan(e+fx))^3\left(5bd^2(\right.}{}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^3)/(6*d*f) + ((-3*(b*c*C - 2
*b*B*d - a*C*d)*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^3)/(5*d*f) + ((
3*b*(5*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(b*c*C - 2*b*B*d - a*C*d))*Tan
[e + f*x]*(c + d*Tan[e + f*x])^3)/(2*d*f) - (((-24*a*d*(5*b*(A*b + a*B - b*
C)*d^2 + (b*c - a*d)*(b*c*C - 2*b*B*d - a*C*d)) + b*(-120*(a^2*B - b^2*B +
2*a*b*(A - C))*d^3 + 6*c*(5*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(b*c*C -
2*b*B*d - a*C*d))))*(c + d*Tan[e + f*x])^3)/(3*d*f) - (60*(d^2*(3*a^2*b*(A*
c - c*C + B*d) - b^3*(A*c - c*C + B*d) + a^3*(B*c - (A - C)*d) - 3*a*b^2*(B
*c - (A - C)*d))*(I*(c + I*d)^2*Log[I - Tan[e + f*x]] - I*(c - I*d)^2*Log[I
 + Tan[e + f*x]] - 2*d^2*Tan[e + f*x]) + (a^3*B - 3*a*b^2*B + 3*a^2*b*(A -
C) - b^3*(A - C))*d^2*((I*c - d)^3*Log[I - Tan[e + f*x]] - (I*c + d)^3*Log[
I + Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3*Tan[e + f*x]^2)))/f)/(4*d))/
(5*d))/(6*d)
```

fricas [A] time = 1.24, size = 690, normalized size = 1.04

$$\frac{10\,Cb^{3}d^{2}\tan\left(fx+e\right)^{6}+12\left(2\,Cb^{3}cd+\left(3\,Cab^{2}+Bb^{3}\right)d^{2}\right)\tan\left(fx+e\right)^{5}+15\left(Cb^{3}c^{2}+2\left(3\,Cab^{2}+Bb^{3}\right)cd+\left(3\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] 1/60*(10*C*b^3*d^2*tan(f*x + e)^6 + 12*(2*C*b^3*c*d + (3*C*a*b^2 + B*b^3)*d
^2)*tan(f*x + e)^5 + 15*(C*b^3*c^2 + 2*(3*C*a*b^2 + B*b^3)*c*d + (3*C*a^2*b
 + 3*B*a*b^2 + (A - C)*b^3)*d^2)*tan(f*x + e)^4 + 20*((3*C*a*b^2 + B*b^3)*c
^2 + 2*(3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c*d + (C*a^3 + 3*B*a^2*b + 3*(
A - C)*a*b^2 - B*b^3)*d^2)*tan(f*x + e)^3 + 60*(((A - C)*a^3 - 3*B*a^2*b -
3*(A - C)*a*b^2 + B*b^3)*c^2 - 2*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A
- C)*b^3)*c*d - ((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*d^2)*f*
x + 30*((3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c^2 + 2*(C*a^3 + 3*B*a^2*b +
3*(A - C)*a*b^2 - B*b^3)*c*d + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A -
C)*b^3)*d^2)*tan(f*x + e)^2 - 30*((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A
 - C)*b^3)*c^2 + 2*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d
- (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d^2)*log(1/(tan(f*x +
 e)^2 + 1)) + 60*((C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*c^2 + 2*(B*
a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d + ((A - C)*a^3 - 3*B*a
^2*b - 3*(A - C)*a*b^2 + B*b^3)*d^2)*tan(f*x + e))/f
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.03, size = 1807, normalized size = 2.73

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

```
[Out] -1/f*B*tan(f*x+e)^2*b^3*c*d-3/f*ln(1+tan(f*x+e)^2)*B*a^2*b*c*d+3/f*ln(1+tan
(f*x+e)^2)*C*a*b^2*c*d+2/f*C*tan(f*x+e)^3*a^2*b*c*d+3/f*A*tan(f*x+e)^2*a*b^
2*c*d+3/2/f*C*tan(f*x+e)^4*a*b^2*c*d+6/f*A*a^2*b*c*d*tan(f*x+e)+3/f*B*tan(f
*x+e)^2*a^2*b*c*d-6/f*B*a*b^2*c*d*tan(f*x+e)-6/f*C*a^2*b*c*d*tan(f*x+e)-3/f
*C*tan(f*x+e)^2*a*b^2*c*d+2/f*B*tan(f*x+e)^3*a*b^2*c*d-6/f*A*arctan(tan(f*x
+e))*a^2*b*c*d-3/f*ln(1+tan(f*x+e)^2)*A*a*b^2*c*d+6/f*B*arctan(tan(f*x+e))*
a*b^2*c*d+6/f*C*arctan(tan(f*x+e))*a^2*b*c*d+3/2/f*ln(1+tan(f*x+e)^2)*B*a*b
^2*d^2+1/f*B*b^3*d^2*tan(f*x+e)-3/2/f*ln(1+tan(f*x+e)^2)*C*a^2*b*c^2+3/2/f*
ln(1+tan(f*x+e)^2)*A*a^2*b*c^2-3/2/f*ln(1+tan(f*x+e)^2)*A*a^2*b*d^2+2/f*A*a
rctan(tan(f*x+e))*b^3*c*d+3/4/f*C*tan(f*x+e)^4*a^2*b*d^2+1/f*A*tan(f*x+e)^3
*a*b^2*d^2+1/f*B*tan(f*x+e)^3*a^2*b*d^2+3/f*A*a*b^2*c^2*tan(f*x+e)+3/2/f*C*
tan(f*x+e)^2*a^2*b*c^2-1/f*C*tan(f*x+e)^3*a*b^2*d^2-2/3/f*C*tan(f*x+e)^3*b^
3*c*d-2/f*B*arctan(tan(f*x+e))*a^3*c*d-3/2/f*B*tan(f*x+e)^2*a*b^2*d^2-3/f*A
*arctan(tan(f*x+e))*a*b^2*c^2+1/f*C*arctan(tan(f*x+e))*a^3*d^2+1/2/f*B*tan(
f*x+e)^2*a^3*d^2-1/f*C*arctan(tan(f*x+e))*a^3*c^2+1/f*B*arctan(tan(f*x+e))*
b^3*c^2-1/f*B*arctan(tan(f*x+e))*b^3*d^2+1/3/f*B*tan(f*x+e)^3*b^3*c^2+1/2/f
*A*tan(f*x+e)^2*b^3*c^2-1/2/f*A*tan(f*x+e)^2*b^3*d^2+1/4/f*A*tan(f*x+e)^4*b
^3*d^2+1/5/f*B*tan(f*x+e)^5*b^3*d^2-1/f*A*arctan(tan(f*x+e))*a^3*d^2-1/3/f*
B*tan(f*x+e)^3*b^3*d^2+1/3/f*C*tan(f*x+e)^3*a^3*d^2-1/f*B*b^3*c^2*tan(f*x+e
)-1/2/f*ln(1+tan(f*x+e)^2)*C*b^3*d^2+1/f*A*arctan(tan(f*x+e))*a^3*c^2+1/2/f
*ln(1+tan(f*x+e)^2)*B*a^3*c^2+1/2/f*C*tan(f*x+e)^2*b^3*d^2+1/4/f*C*tan(f*x+
e)^4*b^3*c^2+1/2/f*ln(1+tan(f*x+e)^2)*C*b^3*c^2-1/2/f*ln(1+tan(f*x+e)^2)*A*
b^3*c^2+1/f*C*a^3*c^2*tan(f*x+e)-1/f*C*a^3*d^2*tan(f*x+e)-1/2/f*C*tan(f*x+e
)^2*b^3*c^2+1/2/f*ln(1+tan(f*x+e)^2)*A*b^3*d^2+1/6/f*C*b^3*d^2*tan(f*x+e)^6
+1/f*A*a^3*d^2*tan(f*x+e)-1/2/f*ln(1+tan(f*x+e)^2)*B*a^3*d^2-1/4/f*C*tan(f*
x+e)^4*b^3*d^2+3/2/f*A*tan(f*x+e)^2*a^2*b*d^2+3/f*C*a*b^2*d^2*tan(f*x+e)-3/
f*C*a*b^2*c^2*tan(f*x+e)+2/f*B*a^3*c*d*tan(f*x+e)+1/f*C*tan(f*x+e)^2*a^3*c*
d+1/f*ln(1+tan(f*x+e)^2)*B*b^3*c*d-3/f*A*a*b^2*d^2*tan(f*x+e)+2/3/f*A*tan(f
*x+e)^3*b^3*c*d+3/f*B*arctan(tan(f*x+e))*a^2*b*d^2+3/f*B*a^2*b*c^2*tan(f*x+
e)-1/f*ln(1+tan(f*x+e)^2)*C*a^3*c*d+1/f*ln(1+tan(f*x+e)^2)*A*a^3*c*d+3/2/f*
```

```
ln(1+tan(f*x+e)^2)*C*a^2*b*d^2+3/2/f*B*tan(f*x+e)^2*a*b^2*c^2+3/f*A*arctan(
tan(f*x+e))*a*b^2*d^2+1/f*C*tan(f*x+e)^3*a*b^2*c^2+1/2/f*B*tan(f*x+e)^4*b^3
*c*d+2/f*C*b^3*c*d*tan(f*x+e)-3/f*B*arctan(tan(f*x+e))*a^2*b*c^2-3/2/f*ln(1
+tan(f*x+e)^2)*B*a*b^2*c^2+3/f*C*arctan(tan(f*x+e))*a*b^2*c^2-3/f*C*arctan(
tan(f*x+e))*a*b^2*d^2+3/4/f*B*tan(f*x+e)^4*a*b^2*d^2-3/f*B*a^2*b*d^2*tan(f*
x+e)-2/f*A*b^3*c*d*tan(f*x+e)-2/f*C*arctan(tan(f*x+e))*b^3*c*d+3/5/f*C*tan(
f*x+e)^5*a*b^2*d^2-3/2/f*C*tan(f*x+e)^2*a^2*b*d^2+2/5/f*C*tan(f*x+e)^5*b^3*
c*d
```

maxima [A] time = 0.63, size = 691, normalized size = 1.05

$$\frac{10\,Cb^3d^2\tan\left(fx+e\right)^6+12\left(2\,Cb^3cd+\left(3\,Cab^2+Bb^3\right)d^2\right)\tan\left(fx+e\right)^5+15\left(Cb^3c^2+2\left(3\,Cab^2+Bb^3\right)cd+\left(3\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
```

```
[Out] 1/60*(10*C*b^3*d^2*tan(f*x + e)^6 + 12*(2*C*b^3*c*d + (3*C*a*b^2 + B*b^3)*d
^2)*tan(f*x + e)^5 + 15*(C*b^3*c^2 + 2*(3*C*a*b^2 + B*b^3)*c*d + (3*C*a^2*b
 + 3*B*a*b^2 + (A - C)*b^3)*d^2)*tan(f*x + e)^4 + 20*((3*C*a*b^2 + B*b^3)*c
^2 + 2*(3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c*d + (C*a^3 + 3*B*a^2*b + 3*(
A - C)*a*b^2 - B*b^3)*d^2)*tan(f*x + e)^3 + 30*((3*C*a^2*b + 3*B*a*b^2 + (A
 - C)*b^3)*c^2 + 2*(C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*c*d + (B*a
^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d^2)*tan(f*x + e)^2 + 60*((
(A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c^2 - 2*(B*a^3 + 3*(A -
C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d - ((A - C)*a^3 - 3*B*a^2*b - 3*(A -
 C)*a*b^2 + B*b^3)*d^2)*(f*x + e) + 30*((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^
2 - (A - C)*b^3)*c^2 + 2*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3
)*c*d - (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d^2)*log(tan(f*
x + e)^2 + 1) + 60*((C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*c^2 + 2*(
B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d + ((A - C)*a^3 - 3*B
*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*d^2)*tan(f*x + e))/f
```

mupad [B] time = 9.29, size = 891, normalized size = 1.35

$$x\,\left(A\,a^3\,c^2-A\,a^3\,d^2+B\,b^3\,c^2-C\,a^3\,c^2-B\,b^3\,d^2+C\,a^3\,d^2+2\,A\,b^3\,c\,d-2\,B\,a^3\,c\,d-2\,C\,b^3\,c\,d-3\,A\,a\,b^2\,c^2+3\,A\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^3*(c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] x*(A*a^3*c^2 - A*a^3*d^2 + B*b^3*c^2 - C*a^3*c^2 - B*b^3*d^2 + C*a^3*d^2 +
2*A*b^3*c*d - 2*B*a^3*c*d - 2*C*b^3*c*d - 3*A*a*b^2*c^2 + 3*A*a*b^2*d^2 - 3
*B*a^2*b*c^2 + 3*B*a^2*b*d^2 + 3*C*a*b^2*c^2 - 3*C*a*b^2*d^2 - 6*A*a^2*b*c*
d + 6*B*a*b^2*c*d + 6*C*a^2*b*c*d) - (tan(e + f*x)*(B*b^3*c^2 - A*a^3*d^2 -
 b^2*d*(B*b*d + 3*C*a*d + 2*C*b*c) - C*a^3*c^2 + C*a^3*d^2 + 2*A*b^3*c*d -
2*B*a^3*c*d - 3*A*a*b^2*c^2 + 3*A*a*b^2*d^2 - 3*B*a^2*b*c^2 + 3*B*a^2*b*d^2
 + 3*C*a*b^2*c^2 - 6*A*a^2*b*c*d + 6*B*a*b^2*c*d + 6*C*a^2*b*c*d))/f - (log
(tan(e + f*x)^2 + 1)*((A*b^3*c^2)/2 - (B*a^3*c^2)/2 - (A*b^3*d^2)/2 + (B*a^
3*d^2)/2 - (C*b^3*c^2)/2 + (C*b^3*d^2)/2 - A*a^3*c*d - B*b^3*c*d + C*a^3*c*
d - (3*A*a^2*b*c^2)/2 + (3*A*a^2*b*d^2)/2 + (3*B*a*b^2*c^2)/2 - (3*B*a*b^2*
d^2)/2 + (3*C*a^2*b*c^2)/2 - (3*C*a^2*b*d^2)/2 + 3*A*a*b^2*c*d + 3*B*a^2*b*
c*d - 3*C*a*b^2*c*d))/f + (tan(e + f*x)^4*((A*b^3*d^2)/4 + (C*b^3*c^2)/4 -
(C*b^3*d^2)/4 + (B*b^3*c*d)/2 + (3*B*a*b^2*d^2)/4 + (3*C*a^2*b*d^2)/4 + (3*
C*a*b^2*c*d)/2))/f + (tan(e + f*x)^3*((B*b^3*c^2)/3 - (b^2*d*(B*b*d + 3*C*a
*d + 2*C*b*c))/3 + (C*a^3*d^2)/3 + (2*A*b^3*c*d)/3 + A*a*b^2*d^2 + B*a^2*b*
d^2 + C*a*b^2*c^2 + 2*B*a*b^2*c*d + 2*C*a^2*b*c*d))/f + (tan(e + f*x)^2*((A
*b^3*c^2)/2 - (A*b^3*d^2)/2 + (B*a^3*d^2)/2 - (C*b^3*c^2)/2 + (C*b^3*d^2)/2
 - B*b^3*c*d + C*a^3*c*d + (3*A*a^2*b*d^2)/2 + (3*B*a*b^2*c^2)/2 - (3*B*a*b
^2*d^2)/2 + (3*C*a^2*b*c^2)/2 - (3*C*a^2*b*d^2)/2 + 3*A*a*b^2*c*d + 3*B*a^2
*b*c*d - 3*C*a*b^2*c*d))/f + (b^2*d*tan(e + f*x)^5*(B*b*d + 3*C*a*d + 2*C*b
*c))/(5*f) + (C*b^3*d^2*tan(e + f*x)^6)/(6*f)
```

sympy [A] time = 3.18, size = 1819, normalized size = 2.75

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**3*(c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
```

```
[Out] Piecewise((A*a**3*c**2*x + A*a**3*c*d*log(tan(e + f*x)**2 + 1)/f - A*a**3*d
**2*x + A*a**3*d**2*tan(e + f*x)/f + 3*A*a**2*b*c**2*log(tan(e + f*x)**2 +
1)/(2*f) - 6*A*a**2*b*c*d*x + 6*A*a**2*b*c*d*tan(e + f*x)/f - 3*A*a**2*b*d*
*2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*A*a**2*b*d**2*tan(e + f*x)**2/(2*f) -
 3*A*a*b**2*c**2*x + 3*A*a*b**2*c**2*tan(e + f*x)/f - 3*A*a*b**2*c*d*log(ta
n(e + f*x)**2 + 1)/f + 3*A*a*b**2*c*d*tan(e + f*x)**2/f + 3*A*a*b**2*d**2*x
 + A*a*b**2*d**2*tan(e + f*x)**3/f - 3*A*a*b**2*d**2*tan(e + f*x)/f - A*b**
3*c**2*log(tan(e + f*x)**2 + 1)/(2*f) + A*b**3*c**2*tan(e + f*x)**2/(2*f) +
 2*A*b**3*c*d*x + 2*A*b**3*c*d*tan(e + f*x)**3/(3*f) - 2*A*b**3*c*d*tan(e +
 f*x)/f + A*b**3*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + A*b**3*d**2*tan(e +
f*x)**4/(4*f) - A*b**3*d**2*tan(e + f*x)**2/(2*f) + B*a**3*c**2*log(tan(e +
 f*x)**2 + 1)/(2*f) - 2*B*a**3*c*d*x + 2*B*a**3*c*d*tan(e + f*x)/f - B*a**3
*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + B*a**3*d**2*tan(e + f*x)**2/(2*f) -
3*B*a**2*b*c**2*x + 3*B*a**2*b*c**2*tan(e + f*x)/f - 3*B*a**2*b*c*d*log(tan
(e + f*x)**2 + 1)/f + 3*B*a**2*b*c*d*tan(e + f*x)**2/f + 3*B*a**2*b*d**2*x
```

```
+ B*a**2*b*d**2*tan(e + f*x)**3/f - 3*B*a**2*b*d**2*tan(e + f*x)/f - 3*B*a*
b**2*c**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*a*b**2*c**2*tan(e + f*x)**2/
(2*f) + 6*B*a*b**2*c*d*x + 2*B*a*b**2*c*d*tan(e + f*x)**3/f - 6*B*a*b**2*c*
d*tan(e + f*x)/f + 3*B*a*b**2*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*a*b
**2*d**2*tan(e + f*x)**4/(4*f) - 3*B*a*b**2*d**2*tan(e + f*x)**2/(2*f) + B*
b**3*c**2*x + B*b**3*c**2*tan(e + f*x)**3/(3*f) - B*b**3*c**2*tan(e + f*x)/
f + B*b**3*c*d*log(tan(e + f*x)**2 + 1)/f + B*b**3*c*d*tan(e + f*x)**4/(2*f
) - B*b**3*c*d*tan(e + f*x)**2/f - B*b**3*d**2*x + B*b**3*d**2*tan(e + f*x)
**5/(5*f) - B*b**3*d**2*tan(e + f*x)**3/(3*f) + B*b**3*d**2*tan(e + f*x)/f
- C*a**3*c**2*x + C*a**3*c**2*tan(e + f*x)/f - C*a**3*c*d*log(tan(e + f*x)*
*2 + 1)/f + C*a**3*c*d*tan(e + f*x)**2/f + C*a**3*d**2*x + C*a**3*d**2*tan(
e + f*x)**3/(3*f) - C*a**3*d**2*tan(e + f*x)/f - 3*C*a**2*b*c**2*log(tan(e
+ f*x)**2 + 1)/(2*f) + 3*C*a**2*b*c**2*tan(e + f*x)**2/(2*f) + 6*C*a**2*b*c
*d*x + 2*C*a**2*b*c*d*tan(e + f*x)**3/f - 6*C*a**2*b*c*d*tan(e + f*x)/f + 3
*C*a**2*b*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*a**2*b*d**2*tan(e + f*x
)**4/(4*f) - 3*C*a**2*b*d**2*tan(e + f*x)**2/(2*f) + 3*C*a*b**2*c**2*x + C*
a*b**2*c**2*tan(e + f*x)**3/f - 3*C*a*b**2*c**2*tan(e + f*x)/f + 3*C*a*b**2
*c*d*log(tan(e + f*x)**2 + 1)/f + 3*C*a*b**2*c*d*tan(e + f*x)**4/(2*f) - 3*
C*a*b**2*c*d*tan(e + f*x)**2/f - 3*C*a*b**2*d**2*x + 3*C*a*b**2*d**2*tan(e
+ f*x)**5/(5*f) - C*a*b**2*d**2*tan(e + f*x)**3/f + 3*C*a*b**2*d**2*tan(e +
 f*x)/f + C*b**3*c**2*log(tan(e + f*x)**2 + 1)/(2*f) + C*b**3*c**2*tan(e +
f*x)**4/(4*f) - C*b**3*c**2*tan(e + f*x)**2/(2*f) - 2*C*b**3*c*d*x + 2*C*b*
*3*c*d*tan(e + f*x)**5/(5*f) - 2*C*b**3*c*d*tan(e + f*x)**3/(3*f) + 2*C*b**
3*c*d*tan(e + f*x)/f - C*b**3*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + C*b**3*
d**2*tan(e + f*x)**6/(6*f) - C*b**3*d**2*tan(e + f*x)**4/(4*f) + C*b**3*d**
2*tan(e + f*x)**2/(2*f), Ne(f, 0)), (x*(a + b*tan(e))**3*(c + d*tan(e))**2*
(A + B*tan(e) + C*tan(e)**2), True))
```

3.58 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2 \left(A + B\tan(e + fx\right.$

Optimal. Leaf size=443

$$\frac{\log(\cos(e+fx))\left(-\left(a^2\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\right)+2ab\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)+b^2\left(2cd(A-C)\right.\right.}{f}$$

```
[Out] -(a^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-b^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2
))+2*a*b*(2*c*(A-C)*d+B*(c^2-d^2)))*x+(2*a*b*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^
2))-a^2*(2*c*(A-C)*d+B*(c^2-d^2))+b^2*(2*c*(A-C)*d+B*(c^2-d^2)))*ln(cos(f*x
+e))/f+d*(2*a*b*(A*c-B*d-C*c)+a^2*(B*c+(A-C)*d)-b^2*(B*c+(A-C)*d))*tan(f*x+
e)/f+1/2*(a^2*B-b^2*B+2*a*b*(A-C))*(c+d*tan(f*x+e))^2/f+1/60*(8*a^2*C*d^2-1
0*a*b*d*(-4*B*d+C*c)+b^2*(2*c^2*C-5*B*c*d+20*(A-C)*d^2))*(c+d*tan(f*x+e))^3
/d^3/f-1/20*b*(-5*B*b*d-2*C*a*d+2*C*b*c)*tan(f*x+e)*(c+d*tan(f*x+e))^3/d^2/
f+1/5*C*(a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3/d/f
```

Rubi [A] time = 1.28, antiderivative size = 443, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3647, 3637, 3630, 3528, 3525, 3475}

$$\frac{(c+d\tan(e+fx))^3\left(8a^2Cd^2-10abd(cC-4Bd)+b^2\left(20d^2(A-C)-5Bcd+2c^2C\right)\right)}{60d^3f}+\frac{\log(\cos(e+fx))\left(a^2\left(-\left(2\right.\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] -((a^2*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C + 2*B*c*d - C
*d^2 - A*(c^2 - d^2)) + 2*a*b*(2*c*(A - C)*d + B*(c^2 - d^2)))*x) + ((2*a*b
*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - a^2*(2*c*(A - C)*d + B*(c^2 -
d^2)) + b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))*Log[Cos[e + f*x]])/f + (d*(2*a
*b*(A*c - c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A - C)*d))*Tan[e
 + f*x])/f + ((a^2*B - b^2*B + 2*a*b*(A - C))*(c + d*Tan[e + f*x])^2)/(2*f)
 + ((8*a^2*C*d^2 - 10*a*b*d*(c*C - 4*B*d) + b^2*(2*c^2*C - 5*B*c*d + 20*(A
- C)*d^2))*(c + d*Tan[e + f*x])^3)/(60*d^3*f) - (b*(2*b*c*C - 5*b*B*d - 2*a
*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^3)/(20*d^2*f) + (C*(a + b*Tan[e + f
*x])^2*(c + d*Tan[e + f*x])^3)/(5*d*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\frac{C(a+b\tan(e+fx))^2(c+\cdots}{5df}$$

$$=-\frac{b(2bcC-5bBd-2aCd)\cdots}{\cdots}$$

$$=\frac{\left(8a^2Cd^2-10abd(cC-4B\cdots\right.}{\cdots}$$

$$=\frac{\left(a^2B-b^2B+2ab(A-C)\right)\cdots}{2f}$$

$$=-\left(a^2\left(c^2C+2Bcd-Cd^2-\cdots\right.\right.$$

$$=-\left(a^2\left(c^2C+2Bcd-Cd^2-\cdots\right.\right.$$

Mathematica [C] time = 6.50, size = 383, normalized size = 0.86

$$\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}{5df}+\frac{\frac{b\tan(e+fx)(2aCd+5bBd-2bcC)(c+d\tan(e+fx))^3}{4df}-\frac{\frac{(c+d\tan(e+fx))^3\left(-8a^2Cd^2+10abd(cC-4Bd)-\cdots\right.}{3df}\cdots}{\cdots}}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^3)/(5*d*f) + ((b*(-2*b*c*C +
 5*b*B*d + 2*a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^3)/(4*d*f) - (((-8*a^
2*C*d^2 + 10*a*b*d*(c*C - 4*B*d) - b^2*(2*c^2*C - 5*B*c*d + 20*(A - C)*d^2)
)*(c + d*Tan[e + f*x])^3)/(3*d*f) - (10*(d*(2*a*b*(A*c - c*C + B*d) + a^2*(
B*c - (A - C)*d) - b^2*(B*c - (A - C)*d))*(I*(c + I*d)^2*Log[I - Tan[e + f*
x]] - I*(c - I*d)^2*Log[I + Tan[e + f*x]] - 2*d^2*Tan[e + f*x]) + (a^2*B -
b^2*B + 2*a*b*(A - C))*d*((I*c - d)^3*Log[I - Tan[e + f*x]] - (I*c + d)^3*L
og[I + Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3*Tan[e + f*x]^2)))/f)/(4*d
))/(5*d)
```

fricas [A] time = 1.15, size = 462, normalized size = 1.04

$$\frac{12\,Cb^2d^2\tan\left(fx+e\right)^5+15\left(2\,Cb^2cd+\left(2\,Cab+Bb^2\right)d^2\right)\tan\left(fx+e\right)^4+20\left(Cb^2c^2+2\left(2\,Cab+Bb^2\right)cd+\left(Ca^2\cdots\right.\right.}{\cdots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] 1/60*(12*C*b^2*d^2*tan(f*x + e)^5 + 15*(2*C*b^2*c*d + (2*C*a*b + B*b^2)*d^2
)*tan(f*x + e)^4 + 20*(C*b^2*c^2 + 2*(2*C*a*b + B*b^2)*c*d + (C*a^2 + 2*B*a
*b + (A - C)*b^2)*d^2)*tan(f*x + e)^3 + 60*(((A - C)*a^2 - 2*B*a*b - (A - C
)*b^2)*c^2 - 2*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d - ((A - C)*a^2 - 2*B*a*b
 - (A - C)*b^2)*d^2)*f*x + 30*((2*C*a*b + B*b^2)*c^2 + 2*(C*a^2 + 2*B*a*b +
 (A - C)*b^2)*c*d + (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^2)*tan(f*x + e)^2 - 3
0*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2 + 2*((A - C)*a^2 - 2*B*a*b - (A - C)
*b^2)*c*d - (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^2)*log(1/(tan(f*x + e)^2 + 1)
) + 60*((C*a^2 + 2*B*a*b + (A - C)*b^2)*c^2 + 2*(B*a^2 + 2*(A - C)*a*b - B*
b^2)*c*d + ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^2)*tan(f*x + e))/f
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.03, size = 1165, normalized size = 2.63

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

```
[Out] 4/3/f*C*tan(f*x+e)^3*a*b*c*d+2/f*B*tan(f*x+e)^2*a*b*c*d+4/f*A*a*b*c*d*tan(f
*x+e)+4/f*C*arctan(tan(f*x+e))*a*b*c*d-4/f*C*a*b*c*d*tan(f*x+e)-2/f*ln(1+ta
n(f*x+e)^2)*B*a*b*c*d-4/f*A*arctan(tan(f*x+e))*a*b*c*d+2/3/f*B*tan(f*x+e)^3
*b^2*c*d+1/f*C*b^2*d^2*tan(f*x+e)+1/f*A*arctan(tan(f*x+e))*b^2*d^2+1/f*A*b^
2*c^2*tan(f*x+e)-1/2/f*B*tan(f*x+e)^2*b^2*d^2+1/2/f*B*tan(f*x+e)^2*a^2*d^2+
1/f*A*a^2*d^2*tan(f*x+e)+1/2/f*ln(1+tan(f*x+e)^2)*B*a^2*c^2+1/f*A*arctan(ta
n(f*x+e))*a^2*c^2-1/f*A*b^2*d^2*tan(f*x+e)-1/f*C*arctan(tan(f*x+e))*a^2*c^2
+1/4/f*B*tan(f*x+e)^4*b^2*d^2-1/2/f*ln(1+tan(f*x+e)^2)*B*a^2*d^2+1/f*C*arct
an(tan(f*x+e))*a^2*d^2+1/f*C*arctan(tan(f*x+e))*b^2*c^2+1/3/f*C*tan(f*x+e)^
3*b^2*c^2+1/2/f*B*tan(f*x+e)^2*b^2*c^2+1/3/f*A*tan(f*x+e)^3*b^2*d^2-1/2/f*l
n(1+tan(f*x+e)^2)*B*b^2*c^2+1/f*C*a^2*c^2*tan(f*x+e)-1/f*C*b^2*c^2*tan(f*x+
```

```
e)-1/f*a^2*C*d^2*tan(f*x+e)+1/5/f*C*b^2*d^2*tan(f*x+e)^5-1/3/f*C*tan(f*x+e)
^3*b^2*d^2-1/f*A*arctan(tan(f*x+e))*a^2*d^2+1/2/f*ln(1+tan(f*x+e)^2)*B*b^2*
d^2+1/3/f*C*tan(f*x+e)^3*a^2*d^2-1/f*A*arctan(tan(f*x+e))*b^2*c^2-1/f*C*arc
tan(tan(f*x+e))*b^2*d^2-2/f*B*b^2*c*d*tan(f*x+e)+2/3/f*B*tan(f*x+e)^3*a*b*d
^2+2/f*B*a^2*c*d*tan(f*x+e)+1/f*C*tan(f*x+e)^2*a^2*c*d+2/f*B*a*b*c^2*tan(f*
x+e)-1/f*C*tan(f*x+e)^2*a*b*d^2-1/f*C*tan(f*x+e)^2*b^2*c*d+1/f*ln(1+tan(f*x
+e)^2)*C*a*b*d^2+1/f*A*tan(f*x+e)^2*a*b*d^2-2/f*B*arctan(tan(f*x+e))*a^2*c*
d-2/f*B*arctan(tan(f*x+e))*a*b*c^2+1/2/f*C*tan(f*x+e)^4*a*b*d^2+2/f*B*arcta
n(tan(f*x+e))*b^2*c*d+1/f*ln(1+tan(f*x+e)^2)*A*a^2*c*d+1/f*ln(1+tan(f*x+e)^
2)*A*a*b*c^2-1/f*ln(1+tan(f*x+e)^2)*A*a*b*d^2-1/f*ln(1+tan(f*x+e)^2)*A*b^2*
c*d+1/2/f*C*tan(f*x+e)^4*b^2*c*d+2/f*B*arctan(tan(f*x+e))*a*b*d^2-2/f*B*a*b
*d^2*tan(f*x+e)+1/f*ln(1+tan(f*x+e)^2)*C*b^2*c*d+1/f*C*tan(f*x+e)^2*a*b*c^2
+1/f*A*tan(f*x+e)^2*b^2*c*d-1/f*ln(1+tan(f*x+e)^2)*C*a^2*c*d-1/f*ln(1+tan(f
*x+e)^2)*C*a*b*c^2
```

maxima [A] time = 0.46, size = 463, normalized size = 1.05

$$12\,Cb^2d^2\tan\left(fx+e\right)^5+15\left(2\,Cb^2cd+\left(2\,Cab+Bb^2\right)d^2\right)\tan\left(fx+e\right)^4+20\left(Cb^2c^2+2\left(2\,Cab+Bb^2\right)cd+\left(Ca^2\right.\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
```

```
[Out] 1/60*(12*C*b^2*d^2*tan(f*x + e)^5 + 15*(2*C*b^2*c*d + (2*C*a*b + B*b^2)*d^2
)*tan(f*x + e)^4 + 20*(C*b^2*c^2 + 2*(2*C*a*b + B*b^2)*c*d + (C*a^2 + 2*B*a
*b + (A - C)*b^2)*d^2)*tan(f*x + e)^3 + 30*((2*C*a*b + B*b^2)*c^2 + 2*(C*a^
2 + 2*B*a*b + (A - C)*b^2)*c*d + (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^2)*tan(f
*x + e)^2 + 60*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^2 - 2*(B*a^2 + 2*(A
 - C)*a*b - B*b^2)*c*d - ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^2)*(f*x +
e) + 30*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2 + 2*((A - C)*a^2 - 2*B*a*b - (
A - C)*b^2)*c*d - (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^2)*log(tan(f*x + e)^2 +
 1) + 60*((C*a^2 + 2*B*a*b + (A - C)*b^2)*c^2 + 2*(B*a^2 + 2*(A - C)*a*b -
B*b^2)*c*d + ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^2)*tan(f*x + e))/f
```

mupad [B] time = 9.12, size = 561, normalized size = 1.27

$$x\left(A\,a^2\,c^2-A\,a^2\,d^2-A\,b^2\,c^2+A\,b^2\,d^2-C\,a^2\,c^2+C\,a^2\,d^2+C\,b^2\,c^2-C\,b^2\,d^2-2\,B\,a\,b\,c^2+2\,B\,a\,b\,d^2-2\,B\,a^2\,c\,d\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] x*(A*a^2*c^2 - A*a^2*d^2 - A*b^2*c^2 + A*b^2*d^2 - C*a^2*c^2 + C*a^2*d^2 +
C*b^2*c^2 - C*b^2*d^2 - 2*B*a*b*c^2 + 2*B*a*b*d^2 - 2*B*a^2*c*d + 2*B*b^2*c
*d - 4*A*a*b*c*d + 4*C*a*b*c*d) - (log(tan(e + f*x)^2 + 1)*((B*a^2*d^2)/2 -
 (B*a^2*c^2)/2 + (B*b^2*c^2)/2 - (B*b^2*d^2)/2 - A*a*b*c^2 + A*a*b*d^2 - A*
a^2*c*d + C*a*b*c^2 + A*b^2*c*d - C*a*b*d^2 + C*a^2*c*d - C*b^2*c*d + 2*B*a
*b*c*d))/f + (tan(e + f*x)^2*((B*a^2*d^2)/2 + (B*b^2*c^2)/2 - (b*d*(B*b*d +
 2*C*a*d + 2*C*b*c))/2 + A*a*b*d^2 + C*a*b*c^2 + A*b^2*c*d + C*a^2*c*d + 2*
B*a*b*c*d))/f + (tan(e + f*x)^3*((A*b^2*d^2)/3 + (C*a^2*d^2)/3 + (C*b^2*c^2
)/3 - (C*b^2*d^2)/3 + (2*B*a*b*d^2)/3 + (2*B*b^2*c*d)/3 + (4*C*a*b*c*d)/3))
/f + (tan(e + f*x)*(A*a^2*d^2 + A*b^2*c^2 - A*b^2*d^2 + C*a^2*c^2 - C*a^2*d
^2 - C*b^2*c^2 + C*b^2*d^2 + 2*B*a*b*c^2 - 2*B*a*b*d^2 + 2*B*a^2*c*d - 2*B*
b^2*c*d + 4*A*a*b*c*d - 4*C*a*b*c*d))/f + (b*d*tan(e + f*x)^4*(B*b*d + 2*C*
a*d + 2*C*b*c))/(4*f) + (C*b^2*d^2*tan(e + f*x)^5)/(5*f)
```

sympy [A] time = 1.91, size = 1134, normalized size = 2.56

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
```

```
[Out] Piecewise((A*a**2*c**2*x + A*a**2*c*d*log(tan(e + f*x)**2 + 1)/f - A*a**2*d
**2*x + A*a**2*d**2*tan(e + f*x)/f + A*a*b*c**2*log(tan(e + f*x)**2 + 1)/f
- 4*A*a*b*c*d*x + 4*A*a*b*c*d*tan(e + f*x)/f - A*a*b*d**2*log(tan(e + f*x)*
*2 + 1)/f + A*a*b*d**2*tan(e + f*x)**2/f - A*b**2*c**2*x + A*b**2*c**2*tan(
e + f*x)/f - A*b**2*c*d*log(tan(e + f*x)**2 + 1)/f + A*b**2*c*d*tan(e + f*x
)**2/f + A*b**2*d**2*x + A*b**2*d**2*tan(e + f*x)**3/(3*f) - A*b**2*d**2*ta
n(e + f*x)/f + B*a**2*c**2*log(tan(e + f*x)**2 + 1)/(2*f) - 2*B*a**2*c*d*x
+ 2*B*a**2*c*d*tan(e + f*x)/f - B*a**2*d**2*log(tan(e + f*x)**2 + 1)/(2*f)
+ B*a**2*d**2*tan(e + f*x)**2/(2*f) - 2*B*a*b*c**2*x + 2*B*a*b*c**2*tan(e +
 f*x)/f - 2*B*a*b*c*d*log(tan(e + f*x)**2 + 1)/f + 2*B*a*b*c*d*tan(e + f*x)
**2/f + 2*B*a*b*d**2*x + 2*B*a*b*d**2*tan(e + f*x)**3/(3*f) - 2*B*a*b*d**2*
tan(e + f*x)/f - B*b**2*c**2*log(tan(e + f*x)**2 + 1)/(2*f) + B*b**2*c**2*t
an(e + f*x)**2/(2*f) + 2*B*b**2*c*d*x + 2*B*b**2*c*d*tan(e + f*x)**3/(3*f)
- 2*B*b**2*c*d*tan(e + f*x)/f + B*b**2*d**2*log(tan(e + f*x)**2 + 1)/(2*f)
+ B*b**2*d**2*tan(e + f*x)**4/(4*f) - B*b**2*d**2*tan(e + f*x)**2/(2*f) - C
*a**2*c**2*x + C*a**2*c**2*tan(e + f*x)/f - C*a**2*c*d*log(tan(e + f*x)**2
+ 1)/f + C*a**2*c*d*tan(e + f*x)**2/f + C*a**2*d**2*x + C*a**2*d**2*tan(e +
 f*x)**3/(3*f) - C*a**2*d**2*tan(e + f*x)/f - C*a*b*c**2*log(tan(e + f*x)**
2 + 1)/f + C*a*b*c**2*tan(e + f*x)**2/f + 4*C*a*b*c*d*x + 4*C*a*b*c*d*tan(e
 + f*x)**3/(3*f) - 4*C*a*b*c*d*tan(e + f*x)/f + C*a*b*d**2*log(tan(e + f*x)
**2 + 1)/f + C*a*b*d**2*tan(e + f*x)**4/(2*f) - C*a*b*d**2*tan(e + f*x)**2/
f + C*b**2*c**2*x + C*b**2*c**2*tan(e + f*x)**3/(3*f) - C*b**2*c**2*tan(e +
 f*x)/f + C*b**2*c*d*log(tan(e + f*x)**2 + 1)/f + C*b**2*c*d*tan(e + f*x)**
```

```
4/(2*f) - C*b**2*c*d*tan(e + f*x)**2/f - C*b**2*d**2*x + C*b**2*d**2*tan(e
+ f*x)**5/(5*f) - C*b**2*d**2*tan(e + f*x)**3/(3*f) + C*b**2*d**2*tan(e + f
*x)/f, Ne(f, 0)), (x*(a + b*tan(e))**2*(c + d*tan(e))**2*(A + B*tan(e) + C*
tan(e)**2), True))
```

3.59 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)\right.$

Optimal. Leaf size=266

$$-\frac{\log(\cos(e+fx))\left(A\left(2acd+b\left(c^2-d^2\right)\right)+a\left(Bc^2-Bd^2-2cCd\right)-b\left(2Bcd+c^2C-Cd^2\right)\right)}{f}-x\left(a\left(-A\left(c^2-d^2\right)\right.\right.$$

```
[Out] -(a*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))+b*(2*c*(A-C)*d+B*(c^2-d^2)))*x-(a*(B*
c^2-B*d^2-2*C*c*d)-b*(2*B*c*d+C*c^2-C*d^2)+A*(2*a*c*d+b*(c^2-d^2)))*ln(cos(
f*x+e))/f+d*(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b*c)*tan(f*x+e)/f+1/2*(A*b+B*a
-C*b)*(c+d*tan(f*x+e))^2/f-1/12*(-4*B*b*d-4*C*a*d+C*b*c)*(c+d*tan(f*x+e))^3
/d^2/f+1/4*b*C*tan(f*x+e)*(c+d*tan(f*x+e))^3/d/f
```

Rubi [A] time = 0.47, antiderivative size = 264, normalized size of antiderivative = 0.99, number of steps used = 5, number of rules used = 5, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3637, 3630, 3528, 3525, 3475}

$$-\frac{\log(\cos(e+fx))\left(2aAcd+aB\left(c^2-d^2\right)-2acCd+Ab\left(c^2-d^2\right)-b\left(2Bcd+c^2C-Cd^2\right)\right)}{f}-x\left(a\left(-A\left(c^2-d^2\right)\right.\right.$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan
[e + f*x]^2),x]
```

```
[Out] -((a*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + b*(2*c*(A - C)*d + B*(c^2
- d^2)))*x) - ((2*a*A*c*d - 2*a*c*C*d + A*b*(c^2 - d^2) + a*B*(c^2 - d^2) -
 b*(c^2*C + 2*B*c*d - C*d^2))*Log[Cos[e + f*x]])/f + (d*(A*b*c + a*B*c - b*
c*C + a*A*d - b*B*d - a*C*d)*Tan[e + f*x])/f + ((A*b + a*B - b*C)*(c + d*Ta
n[e + f*x])^2)/(2*f) - ((b*c*C - 4*b*B*d - 4*a*C*d)*(c + d*Tan[e + f*x])^3)
/(12*d^2*f) + (b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^3)/(4*d*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{bC\tan(e+fx)(c+d\tan(e}{4df}$$

$$= -\frac{(bcC-4bBd-4aCd)(c+}{12d^2f}$$

$$= \frac{(Ab+aB-bC)(c+d\tan(e}{2f}$$

$$= -\Big(a\left(c^2C+2Bcd-Cd^2-A\right.$$

$$= -\Big(a\left(c^2C+2Bcd-Cd^2-A\right.$$

Mathematica [C] time = 2.89, size = 241, normalized size = 0.91

$$\frac{6(-aAd+aBc+aCd+Abc+bBd-bcC)\left(-2d^2\tan(e+fx)+i\left((c+id)^2\log(-\tan(e+fx)+i)-(c-id)^2\log\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] (((-(b*c*C) + 4*b*B*d + 4*a*C*d)*(c + d*Tan[e + f*x])^3)/d + 3*b*C*Tan[e +
f*x]*(c + d*Tan[e + f*x])^3 + 6*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*
C*d)*(I*((c + I*d)^2*Log[I - Tan[e + f*x]] - (c - I*d)^2*Log[I + Tan[e + f*
x]]) - 2*d^2*Tan[e + f*x]) + 6*(A*b + a*B - b*C)*((I*c - d)^3*Log[I - Tan[e
 + f*x]] - (I*c + d)^3*Log[I + Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3*T
an[e + f*x]^2))/(12*d*f)
```

fricas [A] time = 0.60, size = 259, normalized size = 0.97

$$\frac{3\,Cbd^2\tan\left(fx+e\right)^4+4\left(2\,Cbcd+(Ca+Bb)d^2\right)\tan\left(fx+e\right)^3+12\left(((A-C)a-Bb)c^2-2\,(Ba+(A-C)b)cd\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="fricas")
```

```
[Out] 1/12*(3*C*b*d^2*tan(f*x + e)^4 + 4*(2*C*b*c*d + (C*a + B*b)*d^2)*tan(f*x +
e)^3 + 12*(((A - C)*a - B*b)*c^2 - 2*(B*a + (A - C)*b)*c*d - ((A - C)*a - B
*b)*d^2)*f*x + 6*(C*b*c^2 + 2*(C*a + B*b)*c*d + (B*a + (A - C)*b)*d^2)*tan(
f*x + e)^2 - 6*((B*a + (A - C)*b)*c^2 + 2*((A - C)*a - B*b)*c*d - (B*a + (A
 - C)*b)*d^2)*log(1/(tan(f*x + e)^2 + 1)) + 12*((C*a + B*b)*c^2 + 2*(B*a +
(A - C)*b)*c*d + ((A - C)*a - B*b)*d^2)*tan(f*x + e))/f
```

giac [B] time = 33.12, size = 6502, normalized size = 24.44

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="giac")
```

```
[Out] 1/12*(12*A*a*c^2*f*x*tan(f*x)^4*tan(e)^4 - 12*C*a*c^2*f*x*tan(f*x)^4*tan(e)
^4 - 12*B*b*c^2*f*x*tan(f*x)^4*tan(e)^4 - 24*B*a*c*d*f*x*tan(f*x)^4*tan(e)^
```

4 - 24*A*b*c*d*f*x*tan(f*x)^4*tan(e)^4 + 24*C*b*c*d*f*x*tan(f*x)^4*tan(e)^4 - 12*A*a*d^2*f*x*tan(f*x)^4*tan(e)^4 + 12*C*a*d^2*f*x*tan(f*x)^4*tan(e)^4 + 12*B*b*d^2*f*x*tan(f*x)^4*tan(e)^4 - 6*B*a*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 6*A*b*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 6*C*b*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 12*A*a*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 12*C*a*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 12*B*b*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 6*B*a*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 + 6*A*b*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 6*C*b*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 48*A*a*c^2*f*x*tan(f*x)^3*tan(e)^3 + 48*C*a*c^2*f*x*tan(f*x)^3*tan(e)^3 + 48*B*b*c^2*f*x*tan(f*x)^3*tan(e)^3 + 96*B*a*c*d*f*x*tan(f*x)^3*tan(e)^3 + 96*A*b*c*d*f*x*tan(f*x)^3*tan(e)^3 - 96*C*b*c*d*f*x*tan(f*x)^3*tan(e)^3 + 48*A*a*d^2*f*x*tan(f*x)^3*tan(e)^3 - 48*C*a*d^2*f*x*tan(f*x)^3*tan(e)^3 - 48*B*b*d^2*f*x*tan(f*x)^3*tan(e)^3 + 6*C*b*c^2*tan(f*x)^4*tan(e)^4 + 12*C*a*c*d*tan(f*x)^4*tan(e)^4 + 12*B*b*c*d*tan(f*x)^4*tan(e)^4 + 6*B*a*d^2*tan(f*x)^4*tan(e)^4 + 6*A*b*d^2*tan(f*x)^4*tan(e)^4 - 9*C*b*d^2*tan(f*x)^4*tan(e)^4 + 24*B*a*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 24*A*b*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*C*b*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 48*A*a*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 48*C*a*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 48*B*b*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*B*a*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*A*b*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*t

an(f*x)^3*tan(e)^3 + 24*C*b*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*t
an(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2
 + 1))*tan(f*x)^3*tan(e)^3 - 12*C*a*c^2*tan(f*x)^4*tan(e)^3 - 12*B*b*c^2*ta
n(f*x)^4*tan(e)^3 - 24*B*a*c*d*tan(f*x)^4*tan(e)^3 - 24*A*b*c*d*tan(f*x)^4*
tan(e)^3 + 24*C*b*c*d*tan(f*x)^4*tan(e)^3 - 12*A*a*d^2*tan(f*x)^4*tan(e)^3
+ 12*C*a*d^2*tan(f*x)^4*tan(e)^3 + 12*B*b*d^2*tan(f*x)^4*tan(e)^3 - 12*C*a*
c^2*tan(f*x)^3*tan(e)^4 - 12*B*b*c^2*tan(f*x)^3*tan(e)^4 - 24*B*a*c*d*tan(f
*x)^3*tan(e)^4 - 24*A*b*c*d*tan(f*x)^3*tan(e)^4 + 24*C*b*c*d*tan(f*x)^3*tan
(e)^4 - 12*A*a*d^2*tan(f*x)^3*tan(e)^4 + 12*C*a*d^2*tan(f*x)^3*tan(e)^4 + 1
2*B*b*d^2*tan(f*x)^3*tan(e)^4 + 72*A*a*c^2*f*x*tan(f*x)^2*tan(e)^2 - 72*C*a
*c^2*f*x*tan(f*x)^2*tan(e)^2 - 72*B*b*c^2*f*x*tan(f*x)^2*tan(e)^2 - 144*B*a
*c*d*f*x*tan(f*x)^2*tan(e)^2 - 144*A*b*c*d*f*x*tan(f*x)^2*tan(e)^2 + 144*C*
b*c*d*f*x*tan(f*x)^2*tan(e)^2 - 72*A*a*d^2*f*x*tan(f*x)^2*tan(e)^2 + 72*C*a
*d^2*f*x*tan(f*x)^2*tan(e)^2 + 72*B*b*d^2*f*x*tan(f*x)^2*tan(e)^2 + 6*C*b*c
^2*tan(f*x)^4*tan(e)^2 + 12*C*a*c*d*tan(f*x)^4*tan(e)^2 + 12*B*b*c*d*tan(f*
x)^4*tan(e)^2 + 6*B*a*d^2*tan(f*x)^4*tan(e)^2 + 6*A*b*d^2*tan(f*x)^4*tan(e)
^2 - 6*C*b*d^2*tan(f*x)^4*tan(e)^2 - 12*C*b*c^2*tan(f*x)^3*tan(e)^3 - 24*C*
a*c*d*tan(f*x)^3*tan(e)^3 - 24*B*b*c*d*tan(f*x)^3*tan(e)^3 - 12*B*a*d^2*tan
(f*x)^3*tan(e)^3 - 12*A*b*d^2*tan(f*x)^3*tan(e)^3 + 24*C*b*d^2*tan(f*x)^3*t
an(e)^3 + 6*C*b*c^2*tan(f*x)^2*tan(e)^4 + 12*C*a*c*d*tan(f*x)^2*tan(e)^4 +
12*B*b*c*d*tan(f*x)^2*tan(e)^4 + 6*B*a*d^2*tan(f*x)^2*tan(e)^4 + 6*A*b*d^2*
tan(f*x)^2*tan(e)^4 - 6*C*b*d^2*tan(f*x)^2*tan(e)^4 - 8*C*b*c*d*tan(f*x)^4*
tan(e) - 4*C*a*d^2*tan(f*x)^4*tan(e) - 4*B*b*d^2*tan(f*x)^4*tan(e) - 36*B*a
*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2
- 36*A*b*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*t
an(e)^2 + 36*C*b*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(
f*x)^2*tan(e)^2 - 72*A*a*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(
e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 +
1))*tan(f*x)^2*tan(e)^2 + 72*C*a*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x
)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 72*B*b*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2
*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) +
 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*B*a*d^2*log(4*(tan(f*x)^4*tan(
e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*
tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*A*b*d^2*log(4*(tan(f*x
)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*t
an(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 36*C*b*d^2*log(4*
(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)
^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 36*C*a*c^
2*tan(f*x)^3*tan(e)^2 + 36*B*b*c^2*tan(f*x)^3*tan(e)^2 + 72*B*a*c*d*tan(f*x
)^3*tan(e)^2 + 72*A*b*c*d*tan(f*x)^3*tan(e)^2 - 96*C*b*c*d*tan(f*x)^3*tan(e
)^2 + 36*A*a*d^2*tan(f*x)^3*tan(e)^2 - 48*C*a*d^2*tan(f*x)^3*tan(e)^2 - 48*

```
B*b*d^2*tan(f*x)^3*tan(e)^2 + 36*C*a*c^2*tan(f*x)^2*tan(e)^3 + 36*B*b*c^2*t
an(f*x)^2*tan(e)^3 + 72*B*a*c*d*tan(f*x)^2*tan(e)^3 + 72*A*b*c*d*tan(f*x)^2
*tan(e)^3 - 96*C*b*c*d*tan(f*x)^2*tan(e)^3 + 36*A*a*d^2*tan(f*x)^2*tan(e)^3
 - 48*C*a*d^2*tan(f*x)^2*tan(e)^3 - 48*B*b*d^2*tan(f*x)^2*tan(e)^3 - 8*C*b*
c*d*tan(f*x)*tan(e)^4 - 4*C*a*d^2*tan(f*x)*tan(e)^4 - 4*B*b*d^2*tan(f*x)*ta
n(e)^4 + 3*C*b*d^2*tan(f*x)^4 - 48*A*a*c^2*f*x*tan(f*x)*tan(e) + 48*C*a*c^2
*f*x*tan(f*x)*tan(e) + 48*B*b*c^2*f*x*tan(f*x)*tan(e) + 96*B*a*c*d*f*x*tan(
f*x)*tan(e) + 96*A*b*c*d*f*x*tan(f*x)*tan(e) - 96*C*b*c*d*f*x*tan(f*x)*tan(
e) + 48*A*a*d^2*f*x*tan(f*x)*tan(e) - 48*C*a*d^2*f*x*tan(f*x)*tan(e) - 48*B
*b*d^2*f*x*tan(f*x)*tan(e) - 12*C*b*c^2*tan(f*x)^3*tan(e) - 24*C*a*c*d*tan(
f*x)^3*tan(e) - 24*B*b*c*d*tan(f*x)^3*tan(e) - 12*B*a*d^2*tan(f*x)^3*tan(e)
 - 12*A*b*d^2*tan(f*x)^3*tan(e) + 24*C*b*d^2*tan(f*x)^3*tan(e) + 12*C*b*c^2
*tan(f*x)^2*tan(e)^2 + 24*C*a*c*d*tan(f*x)^2*tan(e)^2 + 24*B*b*c*d*tan(f*x)
^2*tan(e)^2 + 12*B*a*d^2*tan(f*x)^2*tan(e)^2 + 12*A*b*d^2*tan(f*x)^2*tan(e)
^2 - 12*C*b*d^2*tan(f*x)^2*tan(e)^2 - 12*C*b*c^2*tan(f*x)*tan(e)^3 - 24*C*a
*c*d*tan(f*x)*tan(e)^3 - 24*B*b*c*d*tan(f*x)*tan(e)^3 - 12*B*a*d^2*tan(f*x)
*tan(e)^3 - 12*A*b*d^2*tan(f*x)*tan(e)^3 + 24*C*b*d^2*tan(f*x)*tan(e)^3 + 3
*C*b*d^2*tan(e)^4 + 8*C*b*c*d*tan(f*x)^3 + 4*C*a*d^2*tan(f*x)^3 + 4*B*b*d^2
*tan(f*x)^3 + 24*B*a*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) +
 tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*
tan(f*x)*tan(e) + 24*A*b*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(
e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 +
1))*tan(f*x)*tan(e) - 24*C*b*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*
tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^
2 + 1))*tan(f*x)*tan(e) + 48*A*a*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x
)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(e)^2 + 1))*tan(f*x)*tan(e) - 48*C*a*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan
(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(e)^2 + 1))*tan(f*x)*tan(e) - 48*B*b*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2
*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) +
 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 24*B*a*d^2*log(4*(tan(f*x)^4*tan(e)^2
 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(
e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 24*A*b*d^2*log(4*(tan(f*x)^4*tan(
e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*
tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) + 24*C*b*d^2*log(4*(tan(f*x)^4*
tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f
*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 36*C*a*c^2*tan(f*x)^2*tan
(e) - 36*B*b*c^2*tan(f*x)^2*tan(e) - 72*B*a*c*d*tan(f*x)^2*tan(e) - 72*A*b*
c*d*tan(f*x)^2*tan(e) + 96*C*b*c*d*tan(f*x)^2*tan(e) - 36*A*a*d^2*tan(f*x)^
2*tan(e) + 48*C*a*d^2*tan(f*x)^2*tan(e) + 48*B*b*d^2*tan(f*x)^2*tan(e) - 36
*C*a*c^2*tan(f*x)*tan(e)^2 - 36*B*b*c^2*tan(f*x)*tan(e)^2 - 72*B*a*c*d*tan(
f*x)*tan(e)^2 - 72*A*b*c*d*tan(f*x)*tan(e)^2 + 96*C*b*c*d*tan(f*x)*tan(e)^2
 - 36*A*a*d^2*tan(f*x)*tan(e)^2 + 48*C*a*d^2*tan(f*x)*tan(e)^2 + 48*B*b*d^2
*tan(f*x)*tan(e)^2 + 8*C*b*c*d*tan(e)^3 + 4*C*a*d^2*tan(e)^3 + 4*B*b*d^2*ta
n(e)^3 + 12*A*a*c^2*f*x - 12*C*a*c^2*f*x - 12*B*b*c^2*f*x - 24*B*a*c*d*f*x
```

```
- 24*A*b*c*d*f*x + 24*C*b*c*d*f*x - 12*A*a*d^2*f*x + 12*C*a*d^2*f*x + 12*B*
b*d^2*f*x + 6*C*b*c^2*tan(f*x)^2 + 12*C*a*c*d*tan(f*x)^2 + 12*B*b*c*d*tan(f
*x)^2 + 6*B*a*d^2*tan(f*x)^2 + 6*A*b*d^2*tan(f*x)^2 - 6*C*b*d^2*tan(f*x)^2
- 12*C*b*c^2*tan(f*x)*tan(e) - 24*C*a*c*d*tan(f*x)*tan(e) - 24*B*b*c*d*tan(
f*x)*tan(e) - 12*B*a*d^2*tan(f*x)*tan(e) - 12*A*b*d^2*tan(f*x)*tan(e) + 24*
C*b*d^2*tan(f*x)*tan(e) + 6*C*b*c^2*tan(e)^2 + 12*C*a*c*d*tan(e)^2 + 12*B*b
*c*d*tan(e)^2 + 6*B*a*d^2*tan(e)^2 + 6*A*b*d^2*tan(e)^2 - 6*C*b*d^2*tan(e)^
2 - 6*B*a*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - 6*A*b*c^2
*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + t
an(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 6*C*b*c^2*log(4*(tan(f
*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2
*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - 12*A*a*c*d*log(4*(tan(f*x)^4*tan(e)
^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(e)^2 + 1)) + 12*C*a*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f
*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(t
an(e)^2 + 1)) + 12*B*b*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e)
 + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)
) + 6*B*a*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 6*A*b*d^2
*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + t
an(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - 6*C*b*d^2*log(4*(tan(f
*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2
*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 12*C*a*c^2*tan(f*x) + 12*B*b*c^2*ta
n(f*x) + 24*B*a*c*d*tan(f*x) + 24*A*b*c*d*tan(f*x) - 24*C*b*c*d*tan(f*x) +
12*A*a*d^2*tan(f*x) - 12*C*a*d^2*tan(f*x) - 12*B*b*d^2*tan(f*x) + 12*C*a*c^
2*tan(e) + 12*B*b*c^2*tan(e) + 24*B*a*c*d*tan(e) + 24*A*b*c*d*tan(e) - 24*C
*b*c*d*tan(e) + 12*A*a*d^2*tan(e) - 12*C*a*d^2*tan(e) - 12*B*b*d^2*tan(e) +
 6*C*b*c^2 + 12*C*a*c*d + 12*B*b*c*d + 6*B*a*d^2 + 6*A*b*d^2 - 9*C*b*d^2)/(
f*tan(f*x)^4*tan(e)^4 - 4*f*tan(f*x)^3*tan(e)^3 + 6*f*tan(f*x)^2*tan(e)^2 -
 4*f*tan(f*x)*tan(e) + f)
```

maple [B] time = 0.03, size = 631, normalized size = 2.37

$$-\frac{Ca\,d^2\tan\left(fx+e\right)}{f}+\frac{Cb\,d^2\left(\tan^4\left(fx+e\right)\right)}{4f}+\frac{C\left(\tan^2\left(fx+e\right)\right)acd}{f}-\frac{\ln\left(1+\tan^2\left(fx+e\right)\right)Cacd}{f}+\frac{\ln\left(1+\tan^2\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] 1/4/f*C*b*d^2*tan(f*x+e)^4-1/f*C*a*d^2*tan(f*x+e)+1/2/f*B*tan(f*x+e)^2*a*d^
2-1/f*B*arctan(tan(f*x+e))*b*c^2+1/f*A*a*d^2*tan(f*x+e)-1/2/f*C*tan(f*x+e)^
2*b*d^2-1/f*C*arctan(tan(f*x+e))*a*c^2-1/f*A*arctan(tan(f*x+e))*a*d^2-1/2/f
*ln(1+tan(f*x+e)^2)*A*b*d^2+1/2/f*ln(1+tan(f*x+e)^2)*B*a*c^2-1/2/f*ln(1+tan
```

```
(f*x+e)^2)*B*a*d^2+1/2/f*A*tan(f*x+e)^2*b*d^2+1/f*C*a*c^2*tan(f*x+e)+1/2/f*
ln(1+tan(f*x+e)^2)*A*b*c^2+1/f*B*b*c^2*tan(f*x+e)-1/f*B*b*d^2*tan(f*x+e)+1/
3/f*C*tan(f*x+e)^3*a*d^2+1/f*C*tan(f*x+e)^2*a*c*d+1/f*C*arctan(tan(f*x+e))*
a*d^2+1/2/f*C*tan(f*x+e)^2*b*c^2+1/f*B*arctan(tan(f*x+e))*b*d^2+1/2/f*ln(1+
tan(f*x+e)^2)*C*b*d^2+2/f*A*b*c*d*tan(f*x+e)-1/f*ln(1+tan(f*x+e)^2)*C*a*c*d
-2/f*C*b*c*d*tan(f*x+e)+1/f*ln(1+tan(f*x+e)^2)*A*a*c*d+2/f*C*arctan(tan(f*x
+e))*b*c*d+1/f*A*arctan(tan(f*x+e))*a*c^2-1/2/f*ln(1+tan(f*x+e)^2)*C*b*c^2+
1/3/f*B*tan(f*x+e)^3*b*d^2-1/f*ln(1+tan(f*x+e)^2)*B*b*c*d+2/3/f*C*tan(f*x+e
)^3*b*c*d+2/f*B*a*c*d*tan(f*x+e)-2/f*B*arctan(tan(f*x+e))*a*c*d-2/f*A*arcta
n(tan(f*x+e))*b*c*d+1/f*B*tan(f*x+e)^2*b*c*d
```

maxima [A] time = 0.55, size = 260, normalized size = 0.98

$$\frac{3\,Cbd^2\tan\left(fx+e\right)^4+4\left(2\,Cbcd+(Ca+Bb)d^2\right)\tan\left(fx+e\right)^3+6\left(Cbc^2+2\,(Ca+Bb)cd+(Ba+(A-C)b)d^2\right)\cdots}{\cdots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="maxima")
```

```
[Out] 1/12*(3*C*b*d^2*tan(f*x + e)^4 + 4*(2*C*b*c*d + (C*a + B*b)*d^2)*tan(f*x +
e)^3 + 6*(C*b*c^2 + 2*(C*a + B*b)*c*d + (B*a + (A - C)*b)*d^2)*tan(f*x + e)
^2 + 12*(((A - C)*a - B*b)*c^2 - 2*(B*a + (A - C)*b)*c*d - ((A - C)*a - B*b
)*d^2)*(f*x + e) + 6*((B*a + (A - C)*b)*c^2 + 2*((A - C)*a - B*b)*c*d - (B*
a + (A - C)*b)*d^2)*log(tan(f*x + e)^2 + 1) + 12*((C*a + B*b)*c^2 + 2*(B*a
+ (A - C)*b)*c*d + ((A - C)*a - B*b)*d^2)*tan(f*x + e))/f
```

mupad [B] time = 9.01, size = 300, normalized size = 1.13

$$\frac{\tan\left(e+f\,x\right)^2\,\left(\frac{A\,b\,d^2}{2}+\frac{B\,a\,d^2}{2}+\frac{C\,b\,c^2}{2}-\frac{C\,b\,d^2}{2}+B\,b\,c\,d+C\,a\,c\,d\right)}{f}-x\,\left(A\,a\,d^2-A\,a\,c^2+B\,b\,c^2+C\,a\,c^2-B\,b\,d^2-C\,\cdots\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2),x)
```

```
[Out] (tan(e + f*x)^2*((A*b*d^2)/2 + (B*a*d^2)/2 + (C*b*c^2)/2 - (C*b*d^2)/2 + B*
b*c*d + C*a*c*d))/f - x*(A*a*d^2 - A*a*c^2 + B*b*c^2 + C*a*c^2 - B*b*d^2 -
C*a*d^2 + 2*A*b*c*d + 2*B*a*c*d - 2*C*b*c*d) - (log(tan(e + f*x)^2 + 1)*((A
*b*d^2)/2 - (B*a*c^2)/2 - (A*b*c^2)/2 + (B*a*d^2)/2 + (C*b*c^2)/2 - (C*b*d^
2)/2 - A*a*c*d + B*b*c*d + C*a*c*d))/f + (tan(e + f*x)*(A*a*d^2 + B*b*c^2 +
 C*a*c^2 - B*b*d^2 - C*a*d^2 + 2*A*b*c*d + 2*B*a*c*d - 2*C*b*c*d))/f + (tan
```

```
(e + f*x)^3*((B*b*d^2)/3 + (C*a*d^2)/3 + (2*C*b*c*d)/3))/f + (C*b*d^2*tan(e
 + f*x)^4)/(4*f)
```

sympy [A] time = 0.97, size = 617, normalized size = 2.32

$$\begin{cases} Aac^2x + \frac{Aacd\log\left(\tan^2\left(e+fx\right)+1\right)}{f} - Aad^2x + \frac{Aad^2\tan\left(e+fx\right)}{f} + \frac{Abc^2\log\left(\tan^2\left(e+fx\right)+1\right)}{2f} - 2Abcdx + \frac{2Abcd\tan\left(e+fx\right)}{f} - \frac{Abd \\ x\left(a+b\tan\left(e\right)\right)\left(c+d\tan\left(e\right)\right)^2\left(A+B\tan\left(e\right)+C\tan^2\left(e\right)\right) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)
**2),x)
```

```
[Out] Piecewise((A*a*c**2*x + A*a*c*d*log(tan(e + f*x)**2 + 1)/f - A*a*d**2*x + A
*a*d**2*tan(e + f*x)/f + A*b*c**2*log(tan(e + f*x)**2 + 1)/(2*f) - 2*A*b*c*
d*x + 2*A*b*c*d*tan(e + f*x)/f - A*b*d**2*log(tan(e + f*x)**2 + 1)/(2*f) +
A*b*d**2*tan(e + f*x)**2/(2*f) + B*a*c**2*log(tan(e + f*x)**2 + 1)/(2*f) -
2*B*a*c*d*x + 2*B*a*c*d*tan(e + f*x)/f - B*a*d**2*log(tan(e + f*x)**2 + 1)/
(2*f) + B*a*d**2*tan(e + f*x)**2/(2*f) - B*b*c**2*x + B*b*c**2*tan(e + f*x)
/f - B*b*c*d*log(tan(e + f*x)**2 + 1)/f + B*b*c*d*tan(e + f*x)**2/f + B*b*d
**2*x + B*b*d**2*tan(e + f*x)**3/(3*f) - B*b*d**2*tan(e + f*x)/f - C*a*c**2
*x + C*a*c**2*tan(e + f*x)/f - C*a*c*d*log(tan(e + f*x)**2 + 1)/f + C*a*c*d
*tan(e + f*x)**2/f + C*a*d**2*x + C*a*d**2*tan(e + f*x)**3/(3*f) - C*a*d**2
*tan(e + f*x)/f - C*b*c**2*log(tan(e + f*x)**2 + 1)/(2*f) + C*b*c**2*tan(e
+ f*x)**2/(2*f) + 2*C*b*c*d*x + 2*C*b*c*d*tan(e + f*x)**3/(3*f) - 2*C*b*c*d
*tan(e + f*x)/f + C*b*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + C*b*d**2*tan(e
+ f*x)**4/(4*f) - C*b*d**2*tan(e + f*x)**2/(2*f), Ne(f, 0)), (x*(a + b*tan(
e))*(c + d*tan(e))**2*(A + B*tan(e) + C*tan(e)**2), True))
```

3.60 $\int (c+d\tan(e+fx))^2 \left(A + B\tan(e + fx) + C\tan^2(e + fx\right.$

Optimal. Leaf size=131

$$-\frac{\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\log(\cos(e+fx))}{f}-x\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)+\frac{d\tan(e+fx)(d(A-C)+}{f}$$

```
[Out] -(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))*x-(2*c*(A-C)*d+B*(c^2-d^2))*ln(cos(f*x+e
))/f+d*(B*c+(A-C)*d)*tan(f*x+e)/f+1/2*B*(c+d*tan(f*x+e))^2/f+1/3*C*(c+d*tan
(f*x+e))^3/d/f
```

Rubi [A] time = 0.16, antiderivative size = 131, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 33, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.121, Rules used = {3630, 3528, 3525, 3475}

$$-\frac{\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\log(\cos(e+fx))}{f}-x\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)+\frac{d\tan(e+fx)(d(A-C)+}{f}$$

Antiderivative was successfully verified.

```
[In] Int[(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -((c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2))*x) - ((2*c*(A - C)*d + B*(c^2 -
 d^2))*Log[Cos[e + f*x]])/f + (d*(B*c + (A - C)*d)*Tan[e + f*x])/f + (B*(c
+ d*Tan[e + f*x])^2)/(2*f) + (C*(c + d*Tan[e + f*x])^3)/(3*d*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
```

```
0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int (c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx=\frac{C(c+d\tan(e+fx))^3}{3df}+\int (A-C+B\tan(e$$

$$=\frac{B(c+d\tan(e+fx))^2}{2f}+\frac{C(c+d\tan(e+fx)}{3df}$$

$$=-\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)x+\frac{d(Bc}{}$$

$$=-\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)x-\frac{(2c(}{}$$

Mathematica [C] time = 1.18, size = 176, normalized size = 1.34

$$\frac{3(d(C-A)+Bc)\left(-2d^2\tan(e+fx)+i\left((c+id)^2\log(-\tan(e+fx)+i)-(c-id)^2\log(\tan(e+fx)+i)\right)\right)+3B}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*C*(c + d*Tan[e + f*x])^3 + 3*(B*c + (-A + C)*d)*(I*((c + I*d)^2*Log[I -
Tan[e + f*x]] - (c - I*d)^2*Log[I + Tan[e + f*x]]) - 2*d^2*Tan[e + f*x]) +
3*B*((I*c - d)^3*Log[I - Tan[e + f*x]] - (I*c + d)^3*Log[I + Tan[e + f*x]]
+ 6*c*d^2*Tan[e + f*x] + d^3*Tan[e + f*x]^2))/(6*d*f)
```

fricas [A] time = 0.56, size = 134, normalized size = 1.02

$$\frac{2\,Cd^2\tan\left(fx+e\right)^3+6\left((A-C)c^2-2\,Bcd-(A-C)d^2\right)fx+3\left(2\,Ccd+Bd^2\right)\tan\left(fx+e\right)^2-3\left(Bc^2+2\,(A-C\right.}{6\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
fricas")
```

```
[Out] 1/6*(2*C*d^2*tan(f*x + e)^3 + 6*((A - C)*c^2 - 2*B*c*d - (A - C)*d^2)*f*x +
 3*(2*C*c*d + B*d^2)*tan(f*x + e)^2 - 3*(B*c^2 + 2*(A - C)*c*d - B*d^2)*log
(1/(tan(f*x + e)^2 + 1)) + 6*(C*c^2 + 2*B*c*d + (A - C)*d^2)*tan(f*x + e))/
f
```

giac [B] time = 5.57, size = 2128, normalized size = 16.24

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
giac")
```

```
[Out] 1/6*(6*A*c^2*f*x*tan(f*x)^3*tan(e)^3 - 6*C*c^2*f*x*tan(f*x)^3*tan(e)^3 - 12
*B*c*d*f*x*tan(f*x)^3*tan(e)^3 - 6*A*d^2*f*x*tan(f*x)^3*tan(e)^3 + 6*C*d^2*
f*x*tan(f*x)^3*tan(e)^3 - 3*B*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3
*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)
^2 + 1))*tan(f*x)^3*tan(e)^3 - 6*A*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f
*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(t
an(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 6*C*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*
tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) +
1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 3*B*d^2*log(4*(tan(f*x)^4*tan(e)^2
 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(
e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 18*A*c^2*f*x*tan(f*x)^2*tan(e
)^2 + 18*C*c^2*f*x*tan(f*x)^2*tan(e)^2 + 36*B*c*d*f*x*tan(f*x)^2*tan(e)^2 +
 18*A*d^2*f*x*tan(f*x)^2*tan(e)^2 - 18*C*d^2*f*x*tan(f*x)^2*tan(e)^2 + 6*C*
c*d*tan(f*x)^3*tan(e)^3 + 3*B*d^2*tan(f*x)^3*tan(e)^3 + 9*B*c^2*log(4*(tan(
f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 18*A*c*d*log(4
*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x
)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 18*C*c*d
*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + t
an(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 9*
B*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2
 - 6*C*c^2*tan(f*x)^3*tan(e)^2 - 12*B*c*d*tan(f*x)^3*tan(e)^2 - 6*A*d^2*tan
(f*x)^3*tan(e)^2 + 6*C*d^2*tan(f*x)^3*tan(e)^2 - 6*C*c^2*tan(f*x)^2*tan(e)^
3 - 12*B*c*d*tan(f*x)^2*tan(e)^3 - 6*A*d^2*tan(f*x)^2*tan(e)^3 + 6*C*d^2*ta
n(f*x)^2*tan(e)^3 + 18*A*c^2*f*x*tan(f*x)*tan(e) - 18*C*c^2*f*x*tan(f*x)*ta
n(e) - 36*B*c*d*f*x*tan(f*x)*tan(e) - 18*A*d^2*f*x*tan(f*x)*tan(e) + 18*C*d
```

```
^2*f*x*tan(f*x)*tan(e) + 6*C*c*d*tan(f*x)^3*tan(e) + 3*B*d^2*tan(f*x)^3*tan
(e) - 6*C*c*d*tan(f*x)^2*tan(e)^2 - 3*B*d^2*tan(f*x)^2*tan(e)^2 + 6*C*c*d*t
an(f*x)*tan(e)^3 + 3*B*d^2*tan(f*x)*tan(e)^3 - 2*C*d^2*tan(f*x)^3 - 9*B*c^2
*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + t
an(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) - 18*A*c
*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 +
 tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) + 18*C
*c*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) + 9*
B*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^
2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)*tan(e) + 1
2*C*c^2*tan(f*x)^2*tan(e) + 24*B*c*d*tan(f*x)^2*tan(e) + 12*A*d^2*tan(f*x)^
2*tan(e) - 18*C*d^2*tan(f*x)^2*tan(e) + 12*C*c^2*tan(f*x)*tan(e)^2 + 24*B*c
*d*tan(f*x)*tan(e)^2 + 12*A*d^2*tan(f*x)*tan(e)^2 - 18*C*d^2*tan(f*x)*tan(e
)^2 - 2*C*d^2*tan(e)^3 - 6*A*c^2*f*x + 6*C*c^2*f*x + 12*B*c*d*f*x + 6*A*d^2
*f*x - 6*C*d^2*f*x - 6*C*c*d*tan(f*x)^2 - 3*B*d^2*tan(f*x)^2 + 6*C*c*d*tan(
f*x)*tan(e) + 3*B*d^2*tan(f*x)*tan(e) - 6*C*c*d*tan(e)^2 - 3*B*d^2*tan(e)^2
 + 3*B*c^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*ta
n(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 6*A*c*d*log(
4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*
x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - 6*C*c*d*log(4*(tan(f*x)^4*t
an(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*
x)*tan(e) + 1)/(tan(e)^2 + 1)) - 3*B*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan
(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/
(tan(e)^2 + 1)) - 6*C*c^2*tan(f*x) - 12*B*c*d*tan(f*x) - 6*A*d^2*tan(f*x) +
 6*C*d^2*tan(f*x) - 6*C*c^2*tan(e) - 12*B*c*d*tan(e) - 6*A*d^2*tan(e) + 6*C
*d^2*tan(e) - 6*C*c*d - 3*B*d^2)/(f*tan(f*x)^3*tan(e)^3 - 3*f*tan(f*x)^2*ta
n(e)^2 + 3*f*tan(f*x)*tan(e) - f)
```

maple [B] time = 0.03, size = 262, normalized size = 2.00

$$\frac{C\,d^2\left(\tan^3\left(fx+e\right)\right)}{3f}+\frac{B\left(\tan^2\left(fx+e\right)\right)d^2}{2f}+\frac{C\left(\tan^2\left(fx+e\right)\right)cd}{f}+\frac{A\,d^2\tan\left(fx+e\right)}{f}+\frac{2Bcd\tan\left(fx+e\right)}{f}+\frac{c^2C\,\mathrm{t}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] 1/3/f*C*d^2*tan(f*x+e)^3+1/2/f*B*tan(f*x+e)^2*d^2+1/f*C*tan(f*x+e)^2*c*d+1/
f*A*d^2*tan(f*x+e)+2/f*B*c*d*tan(f*x+e)+1/f*c^2*C*tan(f*x+e)-1/f*C*d^2*tan(
f*x+e)+1/f*ln(1+tan(f*x+e)^2)*A*c*d+1/2/f*ln(1+tan(f*x+e)^2)*B*c^2-1/2/f*ln
(1+tan(f*x+e)^2)*B*d^2-1/f*ln(1+tan(f*x+e)^2)*c*C*d+1/f*A*arctan(tan(f*x+e)
)*c^2-1/f*A*arctan(tan(f*x+e))*d^2-2/f*B*arctan(tan(f*x+e))*c*d-1/f*C*arcta
n(tan(f*x+e))*c^2+1/f*C*arctan(tan(f*x+e))*d^2
```

maxima [A] time = 0.50, size = 135, normalized size = 1.03

$$\frac{2\,Cd^2\tan\left(fx+e\right)^3+3\left(2\,Ccd+Bd^2\right)\tan\left(fx+e\right)^2+6\left((A-C)c^2-2\,Bcd-(A-C)d^2\right)\left(fx+e\right)+3\left(Bc^2+2\,(A\right.}{6\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
maxima")
```

```
[Out] 1/6*(2*C*d^2*tan(f*x + e)^3 + 3*(2*C*c*d + B*d^2)*tan(f*x + e)^2 + 6*((A -
C)*c^2 - 2*B*c*d - (A - C)*d^2)*(f*x + e) + 3*(B*c^2 + 2*(A - C)*c*d - B*d^
2)*log(tan(f*x + e)^2 + 1) + 6*(C*c^2 + 2*B*c*d + (A - C)*d^2)*tan(f*x + e)
)/f
```

mupad [B] time = 8.81, size = 141, normalized size = 1.08

$$\frac{\tan\left(e+f\,x\right)^2\,\left(\frac{B\,d^2}{2}+C\,c\,d\right)}{f}-x\,\left(A\,d^2-A\,c^2+C\,c^2-C\,d^2+2\,B\,c\,d\right)+\frac{\tan\left(e+f\,x\right)\,\left(A\,d^2+C\,c^2-C\,d^2+2\,B\,c\,d\right)}{f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] (tan(e + f*x)^2*((B*d^2)/2 + C*c*d))/f - x*(A*d^2 - A*c^2 + C*c^2 - C*d^2 +
 2*B*c*d) + (tan(e + f*x)*(A*d^2 + C*c^2 - C*d^2 + 2*B*c*d))/f + (log(tan(e
 + f*x)^2 + 1)*((B*c^2)/2 - (B*d^2)/2 + A*c*d - C*c*d))/f + (C*d^2*tan(e +
f*x)^3)/(3*f)
```

sympy [A] time = 0.47, size = 241, normalized size = 1.84

$$\begin{cases} Ac^2x+\frac{Acd\log\left(\tan^2\left(e+fx\right)+1\right)}{f}-Ad^2x+\frac{Ad^2\tan\left(e+fx\right)}{f}+\frac{Bc^2\log\left(\tan^2\left(e+fx\right)+1\right)}{2f}-2Bcdx+\frac{2Bcd\tan\left(e+fx\right)}{f}-\frac{Bd^2\log\left(\tan^2\left(e\right.\right.}{2f} \\ x\left(c+d\tan\left(e\right)\right)^2\left(A+B\tan\left(e\right)+C\tan^2\left(e\right)\right) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Piecewise((A*c**2*x + A*c*d*log(tan(e + f*x)**2 + 1)/f - A*d**2*x + A*d**2*
tan(e + f*x)/f + B*c**2*log(tan(e + f*x)**2 + 1)/(2*f) - 2*B*c*d*x + 2*B*c*
d*tan(e + f*x)/f - B*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + B*d**2*tan(e + f
*x)**2/(2*f) - C*c**2*x + C*c**2*tan(e + f*x)/f - C*c*d*log(tan(e + f*x)**2
 + 1)/f + C*c*d*tan(e + f*x)**2/f + C*d**2*x + C*d**2*tan(e + f*x)**3/(3*f)
 - C*d**2*tan(e + f*x)/f, Ne(f, 0)), (x*(c + d*tan(e))**2*(A + B*tan(e) + C
*tan(e)**2), True))
```

3.61 $\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$

Optimal. Leaf size=254

$$-\frac{\log(\cos(e+fx))\left(A\left(2acd-b\left(c^2-d^2\right)\right)+a\left(Bc^2-Bd^2-2cCd\right)+b\left(2Bcd+c^2C-Cd^2\right)\right)}{f\left(a^2+b^2\right)}-\frac{x\left(a\left(-A\left(c^2-d^2\right)\right.\right.}{}$$

```
[Out] -(a*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-b*(2*c*(A-C)*d+B*(c^2-d^2)))*x/(a^2+b
^2)-(a*(B*c^2-B*d^2-2*C*c*d)+b*(2*B*c*d+C*c^2-C*d^2)+A*(2*a*c*d-b*(c^2-d^2)
))*ln(cos(f*x+e))/(a^2+b^2)/f+(A*b^2-a*(B*b-C*a))*(-a*d+b*c)^2*ln(a+b*tan(f
*x+e))/b^3/(a^2+b^2)/f+d*(B*b*d-C*a*d+C*b*c)*tan(f*x+e)/b^2/f+1/2*C*(c+d*ta
n(f*x+e))^2/b/f
```

Rubi [A] time = 0.83, antiderivative size = 252, normalized size of antiderivative = 0.99, number of steps used = 6, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3647, 3637, 3626, 3617, 31, 3475}

$$-\frac{\log(\cos(e+fx))\left(2aAcd+aB\left(c^2-d^2\right)-2acCd-Ab\left(c^2-d^2\right)+b\left(2Bcd+c^2C-Cd^2\right)\right)}{f\left(a^2+b^2\right)}-\frac{x\left(a\left(-A\left(c^2-d^2\right)\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x]),x]
```

```
[Out] -(((a*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b*(2*c*(A - C)*d + B*(c^2
 - d^2)))*x)/(a^2 + b^2)) - ((2*a*A*c*d - 2*a*c*C*d - A*b*(c^2 - d^2) + a*B
*(c^2 - d^2) + b*(c^2*C + 2*B*c*d - C*d^2))*Log[Cos[e + f*x]])/((a^2 + b^2)
*f) + ((A*b^2 - a*(b*B - a*C))*(b*c - a*d)^2*Log[a + b*Tan[e + f*x]])/(b^3*
(a^2 + b^2)*f) + (d*(b*c*C + b*B*d - a*C*d)*Tan[e + f*x])/(b^2*f) + (C*(c +
 d*Tan[e + f*x])^2)/(2*b*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \frac{C(c+d\tan(e+fx))^2}{2bf} + \frac{\int \frac{(c+d\tan(e+fx))\left(2(Ab\ldots\right.}{\ldots}}{\ldots}$$

$$= \frac{d(bcC+bBd-aCd)\tan(e+fx)}{b^2f} + \frac{C(c+d\ldots}{\ldots}$$

$$= -\frac{\left(a\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2\ldots\right.\right.}{a^2+b^2}$$

$$= -\frac{\left(a\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2\ldots\right.\right.}{a^2+b^2}$$

$$= -\frac{\left(a\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2\ldots\right.\right.}{a^2+b^2}$$

Mathematica [C] time = 3.03, size = 190, normalized size = 0.75

$$\frac{\frac{2(bc-ad)^2\left(a(aC-bB)+Ab^2\right)\log(a+b\tan(e+fx))}{b^2\left(a^2+b^2\right)} + \frac{b(c-id)^2(iA+B-iC)\log(\tan(e+fx)+i)}{a-ib} + \frac{b(c+id)^2(-iA+B+iC)\log(-\tan(e+fx)+i)}{a+ib} + \frac{2d\tan(e+\ldots}{\ldots}}{2bf}$$

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x]),x]
```

```
[Out] ((b*((-I)*A + B + I*C)*(c + I*d)^2*Log[I - Tan[e + f*x]])/(a + I*b) + (b*(I
*A + B - I*C)*(c - I*d)^2*Log[I + Tan[e + f*x]])/(a - I*b) + (2*(A*b^2 + a*
(-(b*B) + a*C))*(b*c - a*d)^2*Log[a + b*Tan[e + f*x]])/(b^2*(a^2 + b^2)) +
(2*d*(b*c*C + b*B*d - a*C*d)*Tan[e + f*x])/b + C*(c + d*Tan[e + f*x])^2)/(2
*b*f)
```

fricas [A] time = 1.65, size = 397, normalized size = 1.56

$$\frac{\left(Ca^2b^2+Cb^4\right)d^2\tan\left(fx+e\right)^2+2\left(\left((A-C)ab^3+Bb^4\right)c^2-2\left(Bab^3-(A-C)b^4\right)cd-\left((A-C)ab^3+Bb^4\right)d^2\right)f\ldots}{\ldots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] 1/2*((C*a^2*b^2 + C*b^4)*d^2*tan(f*x + e)^2 + 2*(((A - C)*a*b^3 + B*b^4)*c^
2 - 2*(B*a*b^3 - (A - C)*b^4)*c*d - ((A - C)*a*b^3 + B*b^4)*d^2)*f*x + ((C*
a^2*b^2 - B*a*b^3 + A*b^4)*c^2 - 2*(C*a^3*b - B*a^2*b^2 + A*a*b^3)*c*d + (C
*a^4 - B*a^3*b + A*a^2*b^2)*d^2)*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x +
e) + a^2)/(tan(f*x + e)^2 + 1)) - ((C*a^2*b^2 + C*b^4)*c^2 - 2*(C*a^3*b - B
*a^2*b^2 + C*a*b^3 - B*b^4)*c*d + (C*a^4 - B*a^3*b + A*a^2*b^2 - B*a*b^3 +
(A - C)*b^4)*d^2)*log(1/(tan(f*x + e)^2 + 1)) + 2*(2*(C*a^2*b^2 + C*b^4)*c*
d - (C*a^3*b - B*a^2*b^2 + C*a*b^3 - B*b^4)*d^2)*tan(f*x + e))/((a^2*b^3 +
b^5)*f)
```

giac [A] time = 3.66, size = 338, normalized size = 1.33

$$\frac{\frac{2\left(Aac^2-Cac^2+Bbc^2-2\,Bacd+2\,Abcd-2\,Cbcd-Aad^2+Cad^2-Bbd^2\right)\left(fx+e\right)}{a^2+b^2}+\frac{\left(Bac^2-Abc^2+Cbc^2+2\,Aacd-2\,Cacd+2\,Bbcd-Bad^2+Abd^2-Cbd^2\right)\log\left(\tan\right.}{a^2+b^2}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c^2 - C*a*c^2 + B*b*c^2 - 2*B*a*c*d + 2*A*b*c*d - 2*C*b*c*d - A
*a*d^2 + C*a*d^2 - B*b*d^2)*(f*x + e)/(a^2 + b^2) + (B*a*c^2 - A*b*c^2 + C*
b*c^2 + 2*A*a*c*d - 2*C*a*c*d + 2*B*b*c*d - B*a*d^2 + A*b*d^2 - C*b*d^2)*lo
g(tan(f*x + e)^2 + 1)/(a^2 + b^2) + 2*(C*a^2*b^2*c^2 - B*a*b^3*c^2 + A*b^4*
c^2 - 2*C*a^3*b*c*d + 2*B*a^2*b^2*c*d - 2*A*a*b^3*c*d + C*a^4*d^2 - B*a^3*b
*d^2 + A*a^2*b^2*d^2)*log(abs(b*tan(f*x + e) + a))/(a^2*b^3 + b^5) + (C*b*d
^2*tan(f*x + e)^2 + 4*C*b*c*d*tan(f*x + e) - 2*C*a*d^2*tan(f*x + e) + 2*B*b
*d^2*tan(f*x + e))/b^2)/f
```

maple [B] time = 0.24, size = 861, normalized size = 3.39

$$\frac{A\arctan\left(\tan\left(fx+e\right)\right)a\,c^2}{f\left(a^2+b^2\right)}-\frac{2\ln\left(a+b\tan\left(fx+e\right)\right)Aacd}{f\left(a^2+b^2\right)}+\frac{\ln\left(a+b\tan\left(fx+e\right)\right)C\,a^4d^2}{f\,b^3\left(a^2+b^2\right)}+\frac{2A\arctan\left(\tan\left(fx\right.\right.}{f\left(a^2+b^2\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e)),x)
```

```
[Out] 1/f/(a^2+b^2)*A*arctan(tan(f*x+e))*a*c^2-2/f/(a^2+b^2)*ln(a+b*tan(f*x+e))*A
*a*c*d+1/f/b^3/(a^2+b^2)*ln(a+b*tan(f*x+e))*C*a^4*d^2+2/f/(a^2+b^2)*A*arcta
n(tan(f*x+e))*b*c*d-2/f/(a^2+b^2)*B*arctan(tan(f*x+e))*a*c*d-2/f/(a^2+b^2)*
C*arctan(tan(f*x+e))*b*c*d+1/f/b/(a^2+b^2)*ln(a+b*tan(f*x+e))*a^2*A*d^2+1/f
```

```
/b/(a^2+b^2)*ln(a+b*tan(f*x+e))*a^2*C*c^2-1/f/b^2/(a^2+b^2)*ln(a+b*tan(f*x+
e))*B*a^3*d^2+1/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*B*b*c*d-1/f/(a^2+b^2)*ln(1+t
an(f*x+e)^2)*C*a*c*d+1/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*A*a*c*d+2/f/b/(a^2+b^
2)*ln(a+b*tan(f*x+e))*a^2*B*c*d-2/f/b^2/(a^2+b^2)*ln(a+b*tan(f*x+e))*C*a^3*
c*d+1/2/f*d^2/b*C*tan(f*x+e)^2+1/f*d^2/b*B*tan(f*x+e)+1/2/f/(a^2+b^2)*ln(1+
tan(f*x+e)^2)*A*b*d^2+1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*B*a*c^2-1/f/(a^2+b
^2)*A*arctan(tan(f*x+e))*a*d^2-1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*C*b*d^2-1
/f/(a^2+b^2)*B*arctan(tan(f*x+e))*b*d^2-1/f*d^2/b^2*C*tan(f*x+e)*a-1/f/(a^2
+b^2)*ln(a+b*tan(f*x+e))*B*a*c^2-1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*A*b*c^2
+1/f*b/(a^2+b^2)*ln(a+b*tan(f*x+e))*A*c^2+1/f/(a^2+b^2)*B*arctan(tan(f*x+e)
)*b*c^2+2/f*d/b*C*c*tan(f*x+e)+1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*C*b*c^2-1
/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*B*a*d^2-1/f/(a^2+b^2)*C*arctan(tan(f*x+e)
)*a*c^2+1/f/(a^2+b^2)*C*arctan(tan(f*x+e))*a*d^2
```

maxima [A] time = 0.54, size = 290, normalized size = 1.14

$$\frac{\frac{2\left(((A-C)a+Bb)c^2-2\,(Ba-(A-C)b)cd-((A-C)a+Bb)d^2\right)(fx+e)}{a^2+b^2}+\frac{2\left((Ca^2b^2-Bab^3+Ab^4)c^2-2\left(Ca^3b-Ba^2b^2+Aab^3\right)cd+\left(Ca^4-Ba^3b+Aa^2b^2\right)d^2\right)\log(\cdots}{a^2b^3+b^5}}{2\cdots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a + B*b)*c^2 - 2*(B*a - (A - C)*b)*c*d - ((A - C)*a + B*b)
*d^2)*(f*x + e)/(a^2 + b^2) + 2*((C*a^2*b^2 - B*a*b^3 + A*b^4)*c^2 - 2*(C*a
^3*b - B*a^2*b^2 + A*a*b^3)*c*d + (C*a^4 - B*a^3*b + A*a^2*b^2)*d^2)*log(b*
tan(f*x + e) + a)/(a^2*b^3 + b^5) + ((B*a - (A - C)*b)*c^2 + 2*((A - C)*a +
 B*b)*c*d - (B*a - (A - C)*b)*d^2)*log(tan(f*x + e)^2 + 1)/(a^2 + b^2) + (C
*b*d^2*tan(f*x + e)^2 + 2*(2*C*b*c*d - (C*a - B*b)*d^2)*tan(f*x + e))/b^2)/
f
```

mupad [B] time = 11.28, size = 325, normalized size = 1.28

$$\frac{\tan\left(e+f\,x\right)\,\left(\frac{B\,d^2+2\,C\,c\,d}{b}-\frac{C\,a\,d^2}{b^2}\right)}{f}+\frac{\ln\left(a+b\,\tan\left(e+f\,x\right)\right)\,\left(b^2\,\left(C\,a^2\,c^2+2\,B\,a^2\,c\,d+A\,a^2\,d^2\right)-b\,\left(B\,a^3\,d^2+2\,C\cdots\right.\right.}{f\,\left(a^2\,b^3+b^5\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x)),x)
```

```
[Out] (tan(e + f*x)*((B*d^2 + 2*C*c*d)/b - (C*a*d^2)/b^2))/f + (log(a + b*tan(e +
 f*x))*(b^2*(A*a^2*d^2 + C*a^2*c^2 + 2*B*a^2*c*d) - b*(B*a^3*d^2 + 2*C*a^3*
```

```
c*d) - b^3*(B*a*c^2 + 2*A*a*c*d) + A*b^4*c^2 + C*a^4*d^2))/(f*(b^5 + a^2*b^
3)) + (log(tan(e + f*x) + 1i)*(A*d^2 - A*c^2 + B*c^2*1i - B*d^2*1i + C*c^2
- C*d^2 + A*c*d*2i + 2*B*c*d - C*c*d*2i))/(2*f*(a*1i + b)) + (log(tan(e + f
*x) - 1i)*(A*d^2*1i - A*c^2*1i + B*c^2 - B*d^2 + C*c^2*1i - C*d^2*1i + 2*A*
c*d + B*c*d*2i - 2*C*c*d))/(2*f*(a + b*1i)) + (C*d^2*tan(e + f*x)^2)/(2*b*f
)
```

sympy [A] time = 8.06, size = 4517, normalized size = 17.78

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e)),x)
```

```
[Out] Piecewise((zoo*x*(c + d*tan(e))**2*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(
a, 0) & Eq(b, 0) & Eq(f, 0)), (-I*A*c**2*f*x*tan(e + f*x)/(-2*b*f*tan(e + f
*x) + 2*I*b*f) - A*c**2*f*x/(-2*b*f*tan(e + f*x) + 2*I*b*f) - I*A*c**2/(-2*
b*f*tan(e + f*x) + 2*I*b*f) - 2*A*c*d*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x)
 + 2*I*b*f) + 2*I*A*c*d*f*x/(-2*b*f*tan(e + f*x) + 2*I*b*f) + 2*A*c*d/(-2*b
*f*tan(e + f*x) + 2*I*b*f) - I*A*d**2*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x)
 + 2*I*b*f) - A*d**2*f*x/(-2*b*f*tan(e + f*x) + 2*I*b*f) - A*d**2*log(tan(e
 + f*x)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) + I*A*d**2*log
(tan(e + f*x)**2 + 1)/(-2*b*f*tan(e + f*x) + 2*I*b*f) + I*A*d**2/(-2*b*f*ta
n(e + f*x) + 2*I*b*f) - B*c**2*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*
b*f) + I*B*c**2*f*x/(-2*b*f*tan(e + f*x) + 2*I*b*f) + B*c**2/(-2*b*f*tan(e
+ f*x) + 2*I*b*f) - 2*I*B*c*d*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b
*f) - 2*B*c*d*f*x/(-2*b*f*tan(e + f*x) + 2*I*b*f) - 2*B*c*d*log(tan(e + f*x
)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) + 2*I*B*c*d*log(tan(
e + f*x)**2 + 1)/(-2*b*f*tan(e + f*x) + 2*I*b*f) + 2*I*B*c*d/(-2*b*f*tan(e
+ f*x) + 2*I*b*f) + 3*B*d**2*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*
f) - 3*I*B*d**2*f*x/(-2*b*f*tan(e + f*x) + 2*I*b*f) - I*B*d**2*log(tan(e +
f*x)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) - B*d**2*log(tan(
e + f*x)**2 + 1)/(-2*b*f*tan(e + f*x) + 2*I*b*f) - 2*B*d**2*tan(e + f*x)**2
/(-2*b*f*tan(e + f*x) + 2*I*b*f) - 3*B*d**2/(-2*b*f*tan(e + f*x) + 2*I*b*f)
 - I*C*c**2*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) - C*c**2*f*x/(
-2*b*f*tan(e + f*x) + 2*I*b*f) - C*c**2*log(tan(e + f*x)**2 + 1)*tan(e + f*
x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) + I*C*c**2*log(tan(e + f*x)**2 + 1)/(-2*
b*f*tan(e + f*x) + 2*I*b*f) + I*C*c**2/(-2*b*f*tan(e + f*x) + 2*I*b*f) + 6*
C*c*d*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) - 6*I*C*c*d*f*x/(-2*
b*f*tan(e + f*x) + 2*I*b*f) - 2*I*C*c*d*log(tan(e + f*x)**2 + 1)*tan(e + f*
x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) - 2*C*c*d*log(tan(e + f*x)**2 + 1)/(-2*b
*f*tan(e + f*x) + 2*I*b*f) - 4*C*c*d*tan(e + f*x)**2/(-2*b*f*tan(e + f*x) +
 2*I*b*f) - 6*C*c*d/(-2*b*f*tan(e + f*x) + 2*I*b*f) + 3*I*C*d**2*f*x*tan(e
+ f*x)/(-2*b*f*tan(e + f*x) + 2*I*b*f) + 3*C*d**2*f*x/(-2*b*f*tan(e + f*x)
```

```
+ 2*I*b*f) + 2*C*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e +
 f*x) + 2*I*b*f) - 2*I*C*d**2*log(tan(e + f*x)**2 + 1)/(-2*b*f*tan(e + f*x)
 + 2*I*b*f) - C*d**2*tan(e + f*x)**3/(-2*b*f*tan(e + f*x) + 2*I*b*f) - I*C*
d**2*tan(e + f*x)**2/(-2*b*f*tan(e + f*x) + 2*I*b*f) - 3*I*C*d**2/(-2*b*f*t
an(e + f*x) + 2*I*b*f), Eq(a, -I*b)), (I*A*c**2*f*x*tan(e + f*x)/(-2*b*f*ta
n(e + f*x) - 2*I*b*f) - A*c**2*f*x/(-2*b*f*tan(e + f*x) - 2*I*b*f) + I*A*c*
*2/(-2*b*f*tan(e + f*x) - 2*I*b*f) - 2*A*c*d*f*x*tan(e + f*x)/(-2*b*f*tan(e
 + f*x) - 2*I*b*f) - 2*I*A*c*d*f*x/(-2*b*f*tan(e + f*x) - 2*I*b*f) + 2*A*c*
d/(-2*b*f*tan(e + f*x) - 2*I*b*f) + I*A*d**2*f*x*tan(e + f*x)/(-2*b*f*tan(e
 + f*x) - 2*I*b*f) - A*d**2*f*x/(-2*b*f*tan(e + f*x) - 2*I*b*f) - A*d**2*lo
g(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - I*A*d
**2*log(tan(e + f*x)**2 + 1)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - I*A*d**2/(-2
*b*f*tan(e + f*x) - 2*I*b*f) - B*c**2*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x)
 - 2*I*b*f) - I*B*c**2*f*x/(-2*b*f*tan(e + f*x) - 2*I*b*f) + B*c**2/(-2*b*f
*tan(e + f*x) - 2*I*b*f) + 2*I*B*c*d*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x)
- 2*I*b*f) - 2*B*c*d*f*x/(-2*b*f*tan(e + f*x) - 2*I*b*f) - 2*B*c*d*log(tan(
e + f*x)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - 2*I*B*c*d*l
og(tan(e + f*x)**2 + 1)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - 2*I*B*c*d/(-2*b*f
*tan(e + f*x) - 2*I*b*f) + 3*B*d**2*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) -
 2*I*b*f) + 3*I*B*d**2*f*x/(-2*b*f*tan(e + f*x) - 2*I*b*f) + I*B*d**2*log(t
an(e + f*x)**2 + 1)*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - B*d**2*l
og(tan(e + f*x)**2 + 1)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - 2*B*d**2*tan(e +
f*x)**2/(-2*b*f*tan(e + f*x) - 2*I*b*f) - 3*B*d**2/(-2*b*f*tan(e + f*x) - 2
*I*b*f) + I*C*c**2*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - C*c**
2*f*x/(-2*b*f*tan(e + f*x) - 2*I*b*f) - C*c**2*log(tan(e + f*x)**2 + 1)*tan
(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - I*C*c**2*log(tan(e + f*x)**2 +
1)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - I*C*c**2/(-2*b*f*tan(e + f*x) - 2*I*b*
f) + 6*C*c*d*f*x*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) + 6*I*C*c*d*f
*x/(-2*b*f*tan(e + f*x) - 2*I*b*f) + 2*I*C*c*d*log(tan(e + f*x)**2 + 1)*tan
(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - 2*C*c*d*log(tan(e + f*x)**2 + 1
)/(-2*b*f*tan(e + f*x) - 2*I*b*f) - 4*C*c*d*tan(e + f*x)**2/(-2*b*f*tan(e +
 f*x) - 2*I*b*f) - 6*C*c*d/(-2*b*f*tan(e + f*x) - 2*I*b*f) - 3*I*C*d**2*f*x
*tan(e + f*x)/(-2*b*f*tan(e + f*x) - 2*I*b*f) + 3*C*d**2*f*x/(-2*b*f*tan(e
+ f*x) - 2*I*b*f) + 2*C*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*b*f*
tan(e + f*x) - 2*I*b*f) + 2*I*C*d**2*log(tan(e + f*x)**2 + 1)/(-2*b*f*tan(e
 + f*x) - 2*I*b*f) - C*d**2*tan(e + f*x)**3/(-2*b*f*tan(e + f*x) - 2*I*b*f)
 + I*C*d**2*tan(e + f*x)**2/(-2*b*f*tan(e + f*x) - 2*I*b*f) + 3*I*C*d**2/(-
2*b*f*tan(e + f*x) - 2*I*b*f), Eq(a, I*b)), ((A*c**2*x + A*c*d*log(tan(e +
f*x)**2 + 1)/f - A*d**2*x + A*d**2*tan(e + f*x)/f + B*c**2*log(tan(e + f*x)
**2 + 1)/(2*f) - 2*B*c*d*x + 2*B*c*d*tan(e + f*x)/f - B*d**2*log(tan(e + f*
x)**2 + 1)/(2*f) + B*d**2*tan(e + f*x)**2/(2*f) - C*c**2*x + C*c**2*tan(e +
 f*x)/f - C*c*d*log(tan(e + f*x)**2 + 1)/f + C*c*d*tan(e + f*x)**2/f + C*d*
*2*x + C*d**2*tan(e + f*x)**3/(3*f) - C*d**2*tan(e + f*x)/f)/a, Eq(b, 0)),
(x*(c + d*tan(e))**2*(A + B*tan(e) + C*tan(e)**2)/(a + b*tan(e)), Eq(f, 0))
, (2*A*a**2*b**2*d**2*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b**5*f) +
```

```
2*A*a*b**3*c**2*f*x/(2*a**2*b**3*f + 2*b**5*f) - 4*A*a*b**3*c*d*log(a/b + t
an(e + f*x))/(2*a**2*b**3*f + 2*b**5*f) + 2*A*a*b**3*c*d*log(tan(e + f*x)**
2 + 1)/(2*a**2*b**3*f + 2*b**5*f) - 2*A*a*b**3*d**2*f*x/(2*a**2*b**3*f + 2*
b**5*f) + 2*A*b**4*c**2*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b**5*f)
- A*b**4*c**2*log(tan(e + f*x)**2 + 1)/(2*a**2*b**3*f + 2*b**5*f) + 4*A*b**
4*c*d*f*x/(2*a**2*b**3*f + 2*b**5*f) + A*b**4*d**2*log(tan(e + f*x)**2 + 1)
/(2*a**2*b**3*f + 2*b**5*f) - 2*B*a**3*b*d**2*log(a/b + tan(e + f*x))/(2*a*
*2*b**3*f + 2*b**5*f) + 4*B*a**2*b**2*c*d*log(a/b + tan(e + f*x))/(2*a**2*b
**3*f + 2*b**5*f) + 2*B*a**2*b**2*d**2*tan(e + f*x)/(2*a**2*b**3*f + 2*b**5
*f) - 2*B*a*b**3*c**2*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b**5*f) +
B*a*b**3*c**2*log(tan(e + f*x)**2 + 1)/(2*a**2*b**3*f + 2*b**5*f) - 4*B*a*b
**3*c*d*f*x/(2*a**2*b**3*f + 2*b**5*f) - B*a*b**3*d**2*log(tan(e + f*x)**2
+ 1)/(2*a**2*b**3*f + 2*b**5*f) + 2*B*b**4*c**2*f*x/(2*a**2*b**3*f + 2*b**5
*f) + 2*B*b**4*c*d*log(tan(e + f*x)**2 + 1)/(2*a**2*b**3*f + 2*b**5*f) - 2*
B*b**4*d**2*f*x/(2*a**2*b**3*f + 2*b**5*f) + 2*B*b**4*d**2*tan(e + f*x)/(2*
a**2*b**3*f + 2*b**5*f) + 2*C*a**4*d**2*log(a/b + tan(e + f*x))/(2*a**2*b**
3*f + 2*b**5*f) - 4*C*a**3*b*c*d*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2
*b**5*f) - 2*C*a**3*b*d**2*tan(e + f*x)/(2*a**2*b**3*f + 2*b**5*f) + 2*C*a*
*2*b**2*c**2*log(a/b + tan(e + f*x))/(2*a**2*b**3*f + 2*b**5*f) + 4*C*a**2*
b**2*c*d*tan(e + f*x)/(2*a**2*b**3*f + 2*b**5*f) + C*a**2*b**2*d**2*tan(e +
 f*x)**2/(2*a**2*b**3*f + 2*b**5*f) - 2*C*a*b**3*c**2*f*x/(2*a**2*b**3*f +
2*b**5*f) - 2*C*a*b**3*c*d*log(tan(e + f*x)**2 + 1)/(2*a**2*b**3*f + 2*b**5
*f) + 2*C*a*b**3*d**2*f*x/(2*a**2*b**3*f + 2*b**5*f) - 2*C*a*b**3*d**2*tan(
e + f*x)/(2*a**2*b**3*f + 2*b**5*f) + C*b**4*c**2*log(tan(e + f*x)**2 + 1)/
(2*a**2*b**3*f + 2*b**5*f) - 4*C*b**4*c*d*f*x/(2*a**2*b**3*f + 2*b**5*f) +
4*C*b**4*c*d*tan(e + f*x)/(2*a**2*b**3*f + 2*b**5*f) - C*b**4*d**2*log(tan(
e + f*x)**2 + 1)/(2*a**2*b**3*f + 2*b**5*f) + C*b**4*d**2*tan(e + f*x)**2/(
2*a**2*b**3*f + 2*b**5*f), True))
```

3.62 $\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$

Optimal. Leaf size=415

$$-\frac{\log(\cos(e+fx))\left(a^2\left(2cd(A-C)+B\left(c^2-d^2\right)\right)+2ab\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)-b^2\left(2cd(A-C)+\right.\right.}{f\left(a^2+b^2\right)^2}$$

```
[Out] -(a^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-b^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2
))-2*a*b*(2*c*(A-C)*d+B*(c^2-d^2)))*x/(a^2+b^2)^2-(2*a*b*(c^2*C+2*B*c*d-C*d
^2-A*(c^2-d^2))+a^2*(2*c*(A-C)*d+B*(c^2-d^2))-b^2*(2*c*(A-C)*d+B*(c^2-d^2))
)*ln(cos(f*x+e))/(a^2+b^2)^2/f-(-a*d+b*c)*(a^3*b*B*d-2*a^4*C*d-b^4*(2*A*d+B
*c)-a*b^3*(2*A*c-3*B*d-2*C*c)+a^2*b^2*(B*c-4*C*d))*ln(a+b*tan(f*x+e))/b^3/(
a^2+b^2)^2/f+(A*b^2-B*a*b+2*C*a^2+C*b^2)*d^2*tan(f*x+e)/b^2/(a^2+b^2)/f-(A*
b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^2/b/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] time = 1.05, antiderivative size = 415, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3645, 3637, 3626, 3617, 31, 3475}

$$-\frac{\log(\cos(e+fx))\left(a^2\left(2cd(A-C)+B\left(c^2-d^2\right)\right)+2ab\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)-b^2\left(2cd(A-C)+\right.\right.}{f\left(a^2+b^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x])^2,x]
```

```
[Out] -(((a^2*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C + 2*B*c*d -
C*d^2 - A*(c^2 - d^2)) - 2*a*b*(2*c*(A - C)*d + B*(c^2 - d^2)))*x)/(a^2 + b
^2)^2) - ((2*a*b*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + a^2*(2*c*(A -
C)*d + B*(c^2 - d^2)) - b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))*Log[Cos[e + f*
x]])/((a^2 + b^2)^2*f) - ((b*c - a*d)*(a^3*b*B*d - 2*a^4*C*d - b^4*(B*c + 2
*A*d) - a*b^3*(2*A*c - 2*c*C - 3*B*d) + a^2*b^2*(B*c - 4*C*d))*Log[a + b*Ta
n[e + f*x]])/(b^3*(a^2 + b^2)^2*f) + ((A*b^2 - a*b*B + 2*a^2*C + b^2*C)*d^2
*Tan[e + f*x])/(b^2*(a^2 + b^2)*f) - ((A*b^2 - a*(b*B - a*C))*(c + d*Tan[e
+ f*x])^2)/(b*(a^2 + b^2)*f*(a + b*Tan[e + f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^2}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))} +$$

$$= \frac{\left(Ab^2-abB+2a^2C+b^2C\right)d^2\tan(e+fx)}{b^2\left(a^2+b^2\right)f} -$$

$$= -\frac{\left(a^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\right.}{}$$

$$= -\frac{\left(a^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\right.}{}$$

$$= -\frac{\left(a^2\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\right.}{}$$

Mathematica [C] time = 8.02, size = 2640, normalized size = 6.36

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x])^2,x]
```

```
[Out] ((-I)*(-2*a^6*A*b^6*c^2 + (2*I)*a^5*A*b^7*c^2 - 2*a^4*A*b^8*c^2 + (2*I)*a^3
*A*b^9*c^2 + a^7*b^5*B*c^2 - I*a^6*b^6*B*c^2 - a^3*b^9*B*c^2 + I*a^2*b^10*B
*c^2 + 2*a^6*b^6*c^2*C - (2*I)*a^5*b^7*c^2*C + 2*a^4*b^8*c^2*C - (2*I)*a^3*
b^9*c^2*C + 2*a^7*A*b^5*c*d - (2*I)*a^6*A*b^6*c*d - 2*a^3*A*b^9*c*d + (2*I)
*a^2*A*b^10*c*d + 4*a^6*b^6*B*c*d - (4*I)*a^5*b^7*B*c*d + 4*a^4*b^8*B*c*d -
 (4*I)*a^3*b^9*B*c*d - 2*a^9*b^3*c*C*d + (2*I)*a^8*b^4*c*C*d - 8*a^7*b^5*c*
C*d + (8*I)*a^6*b^6*c*C*d - 6*a^5*b^7*c*C*d + (6*I)*a^4*b^8*c*C*d + 2*a^6*A
*b^6*d^2 - (2*I)*a^5*A*b^7*d^2 + 2*a^4*A*b^8*d^2 - (2*I)*a^3*A*b^9*d^2 - a^
9*b^3*B*d^2 + I*a^8*b^4*B*d^2 - 4*a^7*b^5*B*d^2 + (4*I)*a^6*b^6*B*d^2 - 3*a
^5*b^7*B*d^2 + (3*I)*a^4*b^8*B*d^2 + 2*a^10*b^2*C*d^2 - (2*I)*a^9*b^3*C*d^2
 + 6*a^8*b^4*C*d^2 - (6*I)*a^7*b^5*C*d^2 + 4*a^6*b^6*C*d^2 - (4*I)*a^5*b^7*
C*d^2)*(e + f*x)*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^2
)/(a^2*(a - I*b)^4*(a + I*b)^3*b^5*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a
 + b*Tan[e + f*x])^2) - (I*(2*a*A*b^4*c^2 - a^2*b^3*B*c^2 + b^5*B*c^2 - 2*a
*b^4*c^2*C - 2*a^2*A*b^3*c*d + 2*A*b^5*c*d - 4*a*b^4*B*c*d + 2*a^4*b*c*C*d
```

```
+ 6*a^2*b^3*c*C*d - 2*a*A*b^4*d^2 + a^4*b*B*d^2 + 3*a^2*b^3*B*d^2 - 2*a^5*C
*d^2 - 4*a^3*b^2*C*d^2)*ArcTan[Tan[e + f*x]]*(a*Cos[e + f*x] + b*Sin[e + f*
x])^2*(c + d*Tan[e + f*x])^2)/(b^3*(a^2 + b^2)^2*f*(c*Cos[e + f*x] + d*Sin[
e + f*x])^2*(a + b*Tan[e + f*x])^2) + ((-2*b*c*C*d - b*B*d^2 + 2*a*C*d^2)*L
og[Cos[e + f*x]]*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^2
)/(b^3*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^2) + ((2*
a*A*b^4*c^2 - a^2*b^3*B*c^2 + b^5*B*c^2 - 2*a*b^4*c^2*C - 2*a^2*A*b^3*c*d +
 2*A*b^5*c*d - 4*a*b^4*B*c*d + 2*a^4*b*c*C*d + 6*a^2*b^3*c*C*d - 2*a*A*b^4*
d^2 + a^4*b*B*d^2 + 3*a^2*b^3*B*d^2 - 2*a^5*C*d^2 - 4*a^3*b^2*C*d^2)*Log[(a
*Cos[e + f*x] + b*Sin[e + f*x])^2]*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c +
 d*Tan[e + f*x])^2)/(2*b^3*(a^2 + b^2)^2*f*(c*Cos[e + f*x] + d*Sin[e + f*x]
)^2*(a + b*Tan[e + f*x])^2) + (Sec[e + f*x]*(a*Cos[e + f*x] + b*Sin[e + f*x
])*(a^5*b*C*d^2 + 2*a^3*b^3*C*d^2 + a*b^5*C*d^2 + a^4*A*b^2*c^2*(e + f*x) -
 a^2*A*b^4*c^2*(e + f*x) + 2*a^3*b^3*B*c^2*(e + f*x) - a^4*b^2*c^2*C*(e + f
*x) + a^2*b^4*c^2*C*(e + f*x) + 4*a^3*A*b^3*c*d*(e + f*x) - 2*a^4*b^2*B*c*d
*(e + f*x) + 2*a^2*b^4*B*c*d*(e + f*x) - 4*a^3*b^3*c*C*d*(e + f*x) - a^4*A*
b^2*d^2*(e + f*x) + a^2*A*b^4*d^2*(e + f*x) - 2*a^3*b^3*B*d^2*(e + f*x) + a
^4*b^2*C*d^2*(e + f*x) - a^2*b^4*C*d^2*(e + f*x) - a^5*b*C*d^2*Cos[2*(e + f
*x)] - 2*a^3*b^3*C*d^2*Cos[2*(e + f*x)] - a*b^5*C*d^2*Cos[2*(e + f*x)] + a^
4*A*b^2*c^2*(e + f*x)*Cos[2*(e + f*x)] - a^2*A*b^4*c^2*(e + f*x)*Cos[2*(e +
 f*x)] + 2*a^3*b^3*B*c^2*(e + f*x)*Cos[2*(e + f*x)] - a^4*b^2*c^2*C*(e + f*
x)*Cos[2*(e + f*x)] + a^2*b^4*c^2*C*(e + f*x)*Cos[2*(e + f*x)] + 4*a^3*A*b^
3*c*d*(e + f*x)*Cos[2*(e + f*x)] - 2*a^4*b^2*B*c*d*(e + f*x)*Cos[2*(e + f*x
)] + 2*a^2*b^4*B*c*d*(e + f*x)*Cos[2*(e + f*x)] - 4*a^3*b^3*c*C*d*(e + f*x)
*Cos[2*(e + f*x)] - a^4*A*b^2*d^2*(e + f*x)*Cos[2*(e + f*x)] + a^2*A*b^4*d^
2*(e + f*x)*Cos[2*(e + f*x)] - 2*a^3*b^3*B*d^2*(e + f*x)*Cos[2*(e + f*x)] +
 a^4*b^2*C*d^2*(e + f*x)*Cos[2*(e + f*x)] - a^2*b^4*C*d^2*(e + f*x)*Cos[2*(
e + f*x)] + a^2*A*b^4*c^2*Sin[2*(e + f*x)] + A*b^6*c^2*Sin[2*(e + f*x)] - a
^3*b^3*B*c^2*Sin[2*(e + f*x)] - a*b^5*B*c^2*Sin[2*(e + f*x)] + a^4*b^2*c^2*
C*Sin[2*(e + f*x)] + a^2*b^4*c^2*C*Sin[2*(e + f*x)] - 2*a^3*A*b^3*c*d*Sin[2
*(e + f*x)] - 2*a*A*b^5*c*d*Sin[2*(e + f*x)] + 2*a^4*b^2*B*c*d*Sin[2*(e + f
*x)] + 2*a^2*b^4*B*c*d*Sin[2*(e + f*x)] - 2*a^5*b*c*C*d*Sin[2*(e + f*x)] -
2*a^3*b^3*c*C*d*Sin[2*(e + f*x)] + a^4*A*b^2*d^2*Sin[2*(e + f*x)] + a^2*A*b
^4*d^2*Sin[2*(e + f*x)] - a^5*b*B*d^2*Sin[2*(e + f*x)] - a^3*b^3*B*d^2*Sin[
2*(e + f*x)] + 2*a^6*C*d^2*Sin[2*(e + f*x)] + 3*a^4*b^2*C*d^2*Sin[2*(e + f*
x)] + a^2*b^4*C*d^2*Sin[2*(e + f*x)] + a^3*A*b^3*c^2*(e + f*x)*Sin[2*(e + f
*x)] - a*A*b^5*c^2*(e + f*x)*Sin[2*(e + f*x)] + 2*a^2*b^4*B*c^2*(e + f*x)*S
in[2*(e + f*x)] - a^3*b^3*c^2*C*(e + f*x)*Sin[2*(e + f*x)] + a*b^5*c^2*C*(e
 + f*x)*Sin[2*(e + f*x)] + 4*a^2*A*b^4*c*d*(e + f*x)*Sin[2*(e + f*x)] - 2*a
^3*b^3*B*c*d*(e + f*x)*Sin[2*(e + f*x)] + 2*a*b^5*B*c*d*(e + f*x)*Sin[2*(e
+ f*x)] - 4*a^2*b^4*c*C*d*(e + f*x)*Sin[2*(e + f*x)] - a^3*A*b^3*d^2*(e + f
*x)*Sin[2*(e + f*x)] + a*A*b^5*d^2*(e + f*x)*Sin[2*(e + f*x)] - 2*a^2*b^4*B
*d^2*(e + f*x)*Sin[2*(e + f*x)] + a^3*b^3*C*d^2*(e + f*x)*Sin[2*(e + f*x)]
- a*b^5*C*d^2*(e + f*x)*Sin[2*(e + f*x)])*(c + d*Tan[e + f*x])^2)/(2*a*(a -
 I*b)^2*(a + I*b)^2*b^2*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e
```

```
+ f*x])^2)
```

fricas [B] time = 3.00, size = 964, normalized size = 2.32

$$2\left(Ca^4b^2 + 2\,Ca^2b^4 + Cb^6\right)d^2\tan\left(fx+e\right)^2 - 2\left(Ca^2b^4 - Bab^5 + Ab^6\right)c^2 + 4\left(Ca^3b^3 - Ba^2b^4 + Aab^5\right)cd - 2\left(Ca\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(C*a^4*b^2 + 2*C*a^2*b^4 + C*b^6)*d^2*tan(f*x + e)^2 - 2*(C*a^2*b^4
- B*a*b^5 + A*b^6)*c^2 + 4*(C*a^3*b^3 - B*a^2*b^4 + A*a*b^5)*c*d - 2*(C*a^4
*b^2 - B*a^3*b^3 + A*a^2*b^4)*d^2 + 2*(((A - C)*a^3*b^3 + 2*B*a^2*b^4 - (A
- C)*a*b^5)*c^2 - 2*(B*a^3*b^3 - 2*(A - C)*a^2*b^4 - B*a*b^5)*c*d - ((A - C
)*a^3*b^3 + 2*B*a^2*b^4 - (A - C)*a*b^5)*d^2)*f*x - ((B*a^3*b^3 - 2*(A - C)
*a^2*b^4 - B*a*b^5)*c^2 - 2*(C*a^5*b - (A - 3*C)*a^3*b^3 - 2*B*a^2*b^4 + A*
a*b^5)*c*d + (2*C*a^6 - B*a^5*b + 4*C*a^4*b^2 - 3*B*a^3*b^3 + 2*A*a^2*b^4)*
d^2 + ((B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^2 - 2*(C*a^4*b^2 - (A - 3*C)
*a^2*b^4 - 2*B*a*b^5 + A*b^6)*c*d + (2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 -
3*B*a^2*b^4 + 2*A*a*b^5)*d^2)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b
*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - (2*(C*a^5*b + 2*C*a^3*b^3 + C*
a*b^5)*c*d - (2*C*a^6 - B*a^5*b + 4*C*a^4*b^2 - 2*B*a^3*b^3 + 2*C*a^2*b^4 -
 B*a*b^5)*d^2 + (2*(C*a^4*b^2 + 2*C*a^2*b^4 + C*b^6)*c*d - (2*C*a^5*b - B*a
^4*b^2 + 4*C*a^3*b^3 - 2*B*a^2*b^4 + 2*C*a*b^5 - B*b^6)*d^2)*tan(f*x + e))*
log(1/(tan(f*x + e)^2 + 1)) + 2*((C*a^3*b^3 - B*a^2*b^4 + A*a*b^5)*c^2 - 2*
(C*a^4*b^2 - B*a^3*b^3 + A*a^2*b^4)*c*d + (2*C*a^5*b - B*a^4*b^2 + (A + 2*C
)*a^3*b^3 + C*a*b^5)*d^2 + (((A - C)*a^2*b^4 + 2*B*a*b^5 - (A - C)*b^6)*c^2
 - 2*(B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c*d - ((A - C)*a^2*b^4 + 2*B*a*b
^5 - (A - C)*b^6)*d^2)*f*x)*tan(f*x + e))/((a^4*b^4 + 2*a^2*b^6 + b^8)*f*ta
n(f*x + e) + (a^5*b^3 + 2*a^3*b^5 + a*b^7)*f)
```

giac [B] time = 3.06, size = 912, normalized size = 2.20

$$\frac{2\,Cd^2\tan\left(fx+e\right)}{b^2} + \frac{2\left(Aa^2c^2 - Ca^2c^2 + 2\,Babc^2 - Ab^2c^2 + Cb^2c^2 - 2\,Ba^2cd + 4\,Aabcd - 4\,Cabcd + 2\,Bb^2cd - Aa^2d^2 + Ca^2d^2 - 2\,Babd^2 + Ab^2d^2 - Cb^2d^2\right)\left(fx+e\right)}{a^4 + 2\,a^2b^2 + b^4}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*d^2*tan(f*x + e)/b^2 + 2*(A*a^2*c^2 - C*a^2*c^2 + 2*B*a*b*c^2 - A*
b^2*c^2 + C*b^2*c^2 - 2*B*a^2*c*d + 4*A*a*b*c*d - 4*C*a*b*c*d + 2*B*b^2*c*d
 - A*a^2*d^2 + C*a^2*d^2 - 2*B*a*b*d^2 + A*b^2*d^2 - C*b^2*d^2)*(f*x + e)/(
a^4 + 2*a^2*b^2 + b^4) + (B*a^2*c^2 - 2*A*a*b*c^2 + 2*C*a*b*c^2 - B*b^2*c^2
 + 2*A*a^2*c*d - 2*C*a^2*c*d + 4*B*a*b*c*d - 2*A*b^2*c*d + 2*C*b^2*c*d - B*
a^2*d^2 + 2*A*a*b*d^2 - 2*C*a*b*d^2 + B*b^2*d^2)*log(tan(f*x + e)^2 + 1)/(a
^4 + 2*a^2*b^2 + b^4) - 2*(B*a^2*b^3*c^2 - 2*A*a*b^4*c^2 + 2*C*a*b^4*c^2 -
B*b^5*c^2 - 2*C*a^4*b*c*d + 2*A*a^2*b^3*c*d - 6*C*a^2*b^3*c*d + 4*B*a*b^4*c
*d - 2*A*b^5*c*d + 2*C*a^5*d^2 - B*a^4*b*d^2 + 4*C*a^3*b^2*d^2 - 3*B*a^2*b^
3*d^2 + 2*A*a*b^4*d^2)*log(abs(b*tan(f*x + e) + a))/(a^4*b^3 + 2*a^2*b^5 +
b^7) + 2*(B*a^2*b^4*c^2*tan(f*x + e) - 2*A*a*b^5*c^2*tan(f*x + e) + 2*C*a*b
^5*c^2*tan(f*x + e) - B*b^6*c^2*tan(f*x + e) - 2*C*a^4*b^2*c*d*tan(f*x + e)
 + 2*A*a^2*b^4*c*d*tan(f*x + e) - 6*C*a^2*b^4*c*d*tan(f*x + e) + 4*B*a*b^5*
c*d*tan(f*x + e) - 2*A*b^6*c*d*tan(f*x + e) + 2*C*a^5*b*d^2*tan(f*x + e) -
B*a^4*b^2*d^2*tan(f*x + e) + 4*C*a^3*b^3*d^2*tan(f*x + e) - 3*B*a^2*b^4*d^2
*tan(f*x + e) + 2*A*a*b^5*d^2*tan(f*x + e) - C*a^4*b^2*c^2 + 2*B*a^3*b^3*c^
2 - 3*A*a^2*b^4*c^2 + C*a^2*b^4*c^2 - A*b^6*c^2 - 2*B*a^4*b^2*c*d + 4*A*a^3
*b^3*c*d - 4*C*a^3*b^3*c*d + 2*B*a^2*b^4*c*d + C*a^6*d^2 - A*a^4*b^2*d^2 +
3*C*a^4*b^2*d^2 - 2*B*a^3*b^3*d^2 + A*a^2*b^4*d^2)/((a^4*b^3 + 2*a^2*b^5 +
b^7)*(b*tan(f*x + e) + a)))/f
```

maple [B] time = 0.29, size = 1554, normalized size = 3.74

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2,x
)
```

```
[Out] 2/f/b^2/(a^2+b^2)/(a+b*tan(f*x+e))*C*a^3*c*d-4/f/(a^2+b^2)^2*C*arctan(tan(f
*x+e))*a*b*c*d-2/f/b/(a^2+b^2)/(a+b*tan(f*x+e))*B*a^2*c*d-4/f*b/(a^2+b^2)^2
*ln(a+b*tan(f*x+e))*B*a*c*d+4/f/(a^2+b^2)^2*A*arctan(tan(f*x+e))*a*b*c*d+2/
f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*B*a*b*c*d+2/f/b^2/(a^2+b^2)^2*ln(a+b*tan(f
*x+e))*C*a^4*c*d-2/f/b^3/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*C*a^5*d^2+2/f*b^2/(
a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*c*d+1/f/b^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*
B*a^4*d^2-2/f/(a^2+b^2)^2*B*arctan(tan(f*x+e))*a^2*c*d+2/f/(a^2+b^2)^2*B*ar
ctan(tan(f*x+e))*a*b*c^2-1/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*A*a*b*c^2+1/f/(
a^2+b^2)^2*ln(1+tan(f*x+e)^2)*A*a*b*d^2+2/f/(a^2+b^2)^2*B*arctan(tan(f*x+e)
)*b^2*c*d-2/f*b/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*a*d^2+2/f/(a^2+b^2)/(a+b*t
an(f*x+e))*A*a*c*d+1/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*C*b^2*c*d-1/f/(a^2+b^
2)^2*ln(1+tan(f*x+e)^2)*C*a*b*d^2+6/f/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*C*a^2*
c*d-2/f/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*a^2*c*d-1/f/b/(a^2+b^2)/(a+b*tan(f
*x+e))*C*a^2*c^2+1/f/b^2/(a^2+b^2)/(a+b*tan(f*x+e))*B*a^3*d^2-1/f/b^3/(a^2+
b^2)/(a+b*tan(f*x+e))*C*a^4*d^2-4/f/b/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*C*a^3*
d^2-2/f*b/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*C*a*c^2+2/f*b/(a^2+b^2)^2*ln(a+b*t
```

```
an(f*x+e))*A*a*c^2+1/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*A*a^2*c*d-1/f/b/(a^2+
b^2)/(a+b*tan(f*x+e))*A*a^2*d^2-1/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*C*a^2*c*
d+1/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*C*a*b*c^2-2/f/(a^2+b^2)^2*B*arctan(tan
(f*x+e))*a*b*d^2-1/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*A*b^2*c*d+1/f*C*d^2/b^2
*tan(f*x+e)-1/f/(a^2+b^2)^2*C*arctan(tan(f*x+e))*b^2*d^2-1/f/(a^2+b^2)^2*A*
arctan(tan(f*x+e))*b^2*c^2+1/f/(a^2+b^2)^2*A*arctan(tan(f*x+e))*b^2*d^2-1/f
/(a^2+b^2)^2*C*arctan(tan(f*x+e))*a^2*c^2+1/f/(a^2+b^2)^2*C*arctan(tan(f*x+
e))*a^2*d^2+1/f/(a^2+b^2)^2*C*arctan(tan(f*x+e))*b^2*c^2+1/2/f/(a^2+b^2)^2*
ln(1+tan(f*x+e)^2)*B*a^2*c^2-1/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*B*a^2*d^2
-1/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*B*b^2*c^2-1/f/(a^2+b^2)^2*ln(a+b*tan(
f*x+e))*B*a^2*c^2+3/f/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*a^2*d^2+1/f/(a^2+b^2
)/(a+b*tan(f*x+e))*B*a*c^2+1/f*b^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*c^2-1/f
*b/(a^2+b^2)/(a+b*tan(f*x+e))*A*c^2+1/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*B*
b^2*d^2+1/f/(a^2+b^2)^2*A*arctan(tan(f*x+e))*a^2*c^2-1/f/(a^2+b^2)^2*A*arct
an(tan(f*x+e))*a^2*d^2
```

maxima [A] time = 0.49, size = 496, normalized size = 1.20

$$\frac{\frac{2\,Cd^2\tan\left(fx+e\right)}{b^2}+\frac{2\left(\left((A-C)a^2+2\,Bab-(A-C)b^2\right)c^2-2\left(Ba^2-2\,(A-C)ab-Bb^2\right)cd-\left((A-C)a^2+2\,Bab-(A-C)b^2\right)d^2\right)\left(fx+e\right)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(\left(Ba^2b^3-2\,(A-C)ab^{4}\right.\right.}{}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*d^2*tan(f*x + e)/b^2 + 2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c^
2 - 2*(B*a^2 - 2*(A - C)*a*b - B*b^2)*c*d - ((A - C)*a^2 + 2*B*a*b - (A - C
)*b^2)*d^2)*(f*x + e)/(a^4 + 2*a^2*b^2 + b^4) - 2*((B*a^2*b^3 - 2*(A - C)*a
*b^4 - B*b^5)*c^2 - 2*(C*a^4*b - (A - 3*C)*a^2*b^3 - 2*B*a*b^4 + A*b^5)*c*d
 + (2*C*a^5 - B*a^4*b + 4*C*a^3*b^2 - 3*B*a^2*b^3 + 2*A*a*b^4)*d^2)*log(b*t
an(f*x + e) + a)/(a^4*b^3 + 2*a^2*b^5 + b^7) + ((B*a^2 - 2*(A - C)*a*b - B*
b^2)*c^2 + 2*((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c*d - (B*a^2 - 2*(A - C)
*a*b - B*b^2)*d^2)*log(tan(f*x + e)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2*((C*
a^2*b^2 - B*a*b^3 + A*b^4)*c^2 - 2*(C*a^3*b - B*a^2*b^2 + A*a*b^3)*c*d + (C
*a^4 - B*a^3*b + A*a^2*b^2)*d^2)/(a^3*b^3 + a*b^5 + (a^2*b^4 + b^6)*tan(f*x
 + e)))/f
```

mupad [B] time = 34.03, size = 3958, normalized size = 9.54

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x))^2,x)
```

```
[Out] (log((2*C^2*a^5*d^4 + 4*C^2*a^3*b^2*d^4 - 2*C^2*a^5*c^2*d^2 - A*B*b^5*c^4 -
 2*A*C*a^5*d^4 + B*C*b^5*c^4 - A^2*a*b^4*c^4 - A^2*a*b^4*d^4 + B^2*a*b^4*c^
4 + B^2*a*b^4*d^4 - C^2*a*b^4*c^4 + 2*A^2*b^5*c*d^3 - 2*A^2*b^5*c^3*d + C^2
*a*b^4*d^4 + 2*B^2*b^5*c^3*d - 4*C^2*a^3*b^2*c^2*d^2 + A*B*a^2*b^3*c^4 + 3*
A*B*a^2*b^3*d^4 - 4*A*C*a^3*b^2*d^4 - B*C*a^2*b^3*c^4 + 5*A*B*b^5*c^2*d^2 +
 2*A*C*a^5*c^2*d^2 - 3*B*C*a^2*b^3*d^4 - B*C*b^5*c^2*d^2 + 2*B^2*a^4*b*c*d^
3 - 2*C^2*a^4*b*c*d^3 + 2*C^2*a^4*b*c^3*d + 6*A^2*a*b^4*c^2*d^2 - 2*A^2*a^2
*b^3*c*d^3 + 2*A^2*a^2*b^3*c^3*d - 6*B^2*a*b^4*c^2*d^2 + 6*B^2*a^2*b^3*c*d^
3 - 2*B^2*a^2*b^3*c^3*d + 4*C^2*a*b^4*c^2*d^2 - 6*C^2*a^2*b^3*c*d^3 + 6*C^2
*a^2*b^3*c^3*d + A*B*a^4*b*d^4 + 2*A*C*a*b^4*c^4 - B*C*a^4*b*d^4 - 2*A*C*b^
5*c*d^3 + 2*A*C*b^5*c^3*d - 4*B*C*a^5*c*d^3 - 8*A*B*a*b^4*c*d^3 + 8*A*B*a*b
^4*c^3*d + 2*A*C*a^4*b*c*d^3 - 2*A*C*a^4*b*c^3*d + 4*B*C*a*b^4*c*d^3 - 8*B*
C*a*b^4*c^3*d - A*B*a^4*b*c^2*d^2 - 10*A*C*a*b^4*c^2*d^2 + 8*A*C*a^2*b^3*c*
d^3 - 8*A*C*a^2*b^3*c^3*d - 8*B*C*a^3*b^2*c*d^3 + 5*B*C*a^4*b*c^2*d^2 - 8*A
*B*a^2*b^3*c^2*d^2 + 4*A*C*a^3*b^2*c^2*d^2 + 16*B*C*a^2*b^3*c^2*d^2)/(b^2*(
a^2 + b^2)^2) + ((c*1i + d)^2*((tan(e + f*x)*(3*B*b^5*c^2 - 5*B*b^5*d^2 - 4
*C*a^5*d^2 + 6*A*b^5*c*d - 10*C*b^5*c*d + 4*A*a*b^4*c^2 - 4*A*a*b^4*d^2 + 2
*B*a^4*b*d^2 - 4*C*a*b^4*c^2 + 8*C*a*b^4*d^2 - B*a^2*b^3*c^2 + B*a^2*b^3*d^
2 - 8*B*a*b^4*c*d + 4*C*a^4*b*c*d - 2*A*a^2*b^3*c*d + 2*C*a^2*b^3*c*d))/(b^
2*(a^2 + b^2)) - (A*b^2*d^2 - A*b^2*c^2 - 8*C*a^2*d^2 + C*b^2*c^2 - C*b^2*d
^2 + 4*B*a*b*d^2 + 2*B*b^2*c*d + 8*C*a*b*c*d)/b + (b*(c*1i + d)^2*(4*a*b -
a^2*tan(e + f*x) + 3*b^2*tan(e + f*x))*(A*1i + B - C*1i))/(a*1i + b)^2)*(A*
1i + B - C*1i))/(2*(a*1i + b)^2) + (tan(e + f*x)*(A^2*b^5*c^4 + A^2*b^5*d^4
 + B^2*b^5*d^4 + C^2*b^5*c^4 + C^2*b^5*d^4 + B^2*a^2*b^3*c^4 + 3*B^2*a^2*b^
3*d^4 - 2*A^2*b^5*c^2*d^2 + 3*B^2*b^5*c^2*d^2 + 2*C^2*b^5*c^2*d^2 - 2*A*C*b
^5*c^4 - 2*A*C*b^5*d^4 - 2*B*C*a^5*d^4 + B^2*a^4*b*d^4 - 4*C^2*a^5*c*d^3 +
4*A^2*a^2*b^3*c^2*d^2 - 4*B^2*a^2*b^3*c^2*d^2 + 12*C^2*a^2*b^3*c^2*d^2 - 4*
B*C*a^3*b^2*d^4 + 2*B*C*a^5*c^2*d^2 + 4*A^2*a*b^4*c*d^3 - 4*A^2*a*b^4*c^3*d
 - 4*B^2*a*b^4*c*d^3 + 4*B^2*a*b^4*c^3*d - 4*C^2*a*b^4*c^3*d - B^2*a^4*b*c^
2*d^2 - 8*C^2*a^3*b^2*c*d^3 + 4*C^2*a^4*b*c^2*d^2 - 2*A*B*a*b^4*c^4 - 2*A*B
*a*b^4*d^4 + 2*B*C*a*b^4*c^4 + 2*A*B*b^5*c*d^3 - 4*A*B*b^5*c^3*d + 4*A*C*a^
5*c*d^3 + 2*B*C*b^5*c^3*d - 2*A*B*a^4*b*c*d^3 - 4*A*C*a*b^4*c*d^3 + 8*A*C*a
*b^4*c^3*d + 4*B*C*a^4*b*c*d^3 - 2*B*C*a^4*b*c^3*d + 12*A*B*a*b^4*c^2*d^2 -
 8*A*B*a^2*b^3*c*d^3 + 4*A*B*a^2*b^3*c^3*d + 8*A*C*a^3*b^2*c*d^3 - 4*A*C*a^
4*b*c^2*d^2 - 10*B*C*a*b^4*c^2*d^2 + 12*B*C*a^2*b^3*c*d^3 - 8*B*C*a^2*b^3*c
^3*d - 16*A*C*a^2*b^3*c^2*d^2 + 4*B*C*a^3*b^2*c^2*d^2))/(b^2*(a^2 + b^2)^2)
)*(A*d^2*1i - A*c^2*1i - B*c^2 + B*d^2 + C*c^2*1i - C*d^2*1i - 2*A*c*d + B*
c*d*2i + 2*C*c*d))/(2*f*(a*b*2i - a^2 + b^2)) - (log(a + b*tan(e + f*x))*(b
^3*(B*a^2*c^2 - 3*B*a^2*d^2 + 2*A*a^2*c*d - 6*C*a^2*c*d) - b^5*(B*c^2 + 2*A
*c*d) - b*(B*a^4*d^2 + 2*C*a^4*c*d) + b^4*(2*A*a*d^2 - 2*A*a*c^2 + 2*C*a*c^
2 + 4*B*a*c*d) + 2*C*a^5*d^2 + 4*C*a^3*b^2*d^2))/(f*(b^7 + 2*a^2*b^5 + a^4*
b^3)) + (log((2*C^2*a^5*d^4 + 4*C^2*a^3*b^2*d^4 - 2*C^2*a^5*c^2*d^2 - A*B*b
^5*c^4 - 2*A*C*a^5*d^4 + B*C*b^5*c^4 - A^2*a*b^4*c^4 - A^2*a*b^4*d^4 + B^2*
a*b^4*c^4 + B^2*a*b^4*d^4 - C^2*a*b^4*c^4 + 2*A^2*b^5*c*d^3 - 2*A^2*b^5*c^3
*d + C^2*a*b^4*d^4 + 2*B^2*b^5*c^3*d - 4*C^2*a^3*b^2*c^2*d^2 + A*B*a^2*b^3*
```

```
c^4 + 3*A*B*a^2*b^3*d^4 - 4*A*C*a^3*b^2*d^4 - B*C*a^2*b^3*c^4 + 5*A*B*b^5*c
^2*d^2 + 2*A*C*a^5*c^2*d^2 - 3*B*C*a^2*b^3*d^4 - B*C*b^5*c^2*d^2 + 2*B^2*a^
4*b*c*d^3 - 2*C^2*a^4*b*c*d^3 + 2*C^2*a^4*b*c^3*d + 6*A^2*a*b^4*c^2*d^2 - 2
*A^2*a^2*b^3*c*d^3 + 2*A^2*a^2*b^3*c^3*d - 6*B^2*a*b^4*c^2*d^2 + 6*B^2*a^2*
b^3*c*d^3 - 2*B^2*a^2*b^3*c^3*d + 4*C^2*a*b^4*c^2*d^2 - 6*C^2*a^2*b^3*c*d^3
 + 6*C^2*a^2*b^3*c^3*d + A*B*a^4*b*d^4 + 2*A*C*a*b^4*c^4 - B*C*a^4*b*d^4 -
2*A*C*b^5*c*d^3 + 2*A*C*b^5*c^3*d - 4*B*C*a^5*c*d^3 - 8*A*B*a*b^4*c*d^3 + 8
*A*B*a*b^4*c^3*d + 2*A*C*a^4*b*c*d^3 - 2*A*C*a^4*b*c^3*d + 4*B*C*a*b^4*c*d^
3 - 8*B*C*a*b^4*c^3*d - A*B*a^4*b*c^2*d^2 - 10*A*C*a*b^4*c^2*d^2 + 8*A*C*a^
2*b^3*c*d^3 - 8*A*C*a^2*b^3*c^3*d - 8*B*C*a^3*b^2*c*d^3 + 5*B*C*a^4*b*c^2*d
^2 - 8*A*B*a^2*b^3*c^2*d^2 + 4*A*C*a^3*b^2*c^2*d^2 + 16*B*C*a^2*b^3*c^2*d^2
)/(b^2*(a^2 + b^2)^2) + ((c*1i - d)^2*((A*b^2*d^2 - A*b^2*c^2 - 8*C*a^2*d^2
 + C*b^2*c^2 - C*b^2*d^2 + 4*B*a*b*d^2 + 2*B*b^2*c*d + 8*C*a*b*c*d)/b - (ta
n(e + f*x)*(3*B*b^5*c^2 - 5*B*b^5*d^2 - 4*C*a^5*d^2 + 6*A*b^5*c*d - 10*C*b^
5*c*d + 4*A*a*b^4*c^2 - 4*A*a*b^4*d^2 + 2*B*a^4*b*d^2 - 4*C*a*b^4*c^2 + 8*C
*a*b^4*d^2 - B*a^2*b^3*c^2 + B*a^2*b^3*d^2 - 8*B*a*b^4*c*d + 4*C*a^4*b*c*d
- 2*A*a^2*b^3*c*d + 2*C*a^2*b^3*c*d))/(b^2*(a^2 + b^2)) + (b*(c*1i - d)^2*(
4*a*b - a^2*tan(e + f*x) + 3*b^2*tan(e + f*x))*(A + B*1i - C)*1i)/(a*1i - b
)^2)*(A + B*1i - C)*1i)/(2*(a*1i - b)^2) + (tan(e + f*x)*(A^2*b^5*c^4 + A^2
*b^5*d^4 + B^2*b^5*d^4 + C^2*b^5*c^4 + C^2*b^5*d^4 + B^2*a^2*b^3*c^4 + 3*B^
2*a^2*b^3*d^4 - 2*A^2*b^5*c^2*d^2 + 3*B^2*b^5*c^2*d^2 + 2*C^2*b^5*c^2*d^2 -
 2*A*C*b^5*c^4 - 2*A*C*b^5*d^4 - 2*B*C*a^5*d^4 + B^2*a^4*b*d^4 - 4*C^2*a^5*
c*d^3 + 4*A^2*a^2*b^3*c^2*d^2 - 4*B^2*a^2*b^3*c^2*d^2 + 12*C^2*a^2*b^3*c^2*
d^2 - 4*B*C*a^3*b^2*d^4 + 2*B*C*a^5*c^2*d^2 + 4*A^2*a*b^4*c*d^3 - 4*A^2*a*b
^4*c^3*d - 4*B^2*a*b^4*c*d^3 + 4*B^2*a*b^4*c^3*d - 4*C^2*a*b^4*c^3*d - B^2*
a^4*b*c^2*d^2 - 8*C^2*a^3*b^2*c*d^3 + 4*C^2*a^4*b*c^2*d^2 - 2*A*B*a*b^4*c^4
 - 2*A*B*a*b^4*d^4 + 2*B*C*a*b^4*c^4 + 2*A*B*b^5*c*d^3 - 4*A*B*b^5*c^3*d +
4*A*C*a^5*c*d^3 + 2*B*C*b^5*c^3*d - 2*A*B*a^4*b*c*d^3 - 4*A*C*a*b^4*c*d^3 +
 8*A*C*a*b^4*c^3*d + 4*B*C*a^4*b*c*d^3 - 2*B*C*a^4*b*c^3*d + 12*A*B*a*b^4*c
^2*d^2 - 8*A*B*a^2*b^3*c*d^3 + 4*A*B*a^2*b^3*c^3*d + 8*A*C*a^3*b^2*c*d^3 -
4*A*C*a^4*b*c^2*d^2 - 10*B*C*a*b^4*c^2*d^2 + 12*B*C*a^2*b^3*c*d^3 - 8*B*C*a
^2*b^3*c^3*d - 16*A*C*a^2*b^3*c^2*d^2 + 4*B*C*a^3*b^2*c^2*d^2))/(b^2*(a^2 +
 b^2)^2))*(A*d^2 - A*c^2 - B*c^2*1i + B*d^2*1i + C*c^2 - C*d^2 - A*c*d*2i +
 2*B*c*d + C*c*d*2i))/(2*f*(2*a*b - a^2*1i + b^2*1i)) + (C*d^2*tan(e + f*x)
)/(b^2*f) - (A*b^4*c^2 + C*a^4*d^2 - B*a*b^3*c^2 - B*a^3*b*d^2 + A*a^2*b^2*
d^2 + C*a^2*b^2*c^2 - 2*A*a*b^3*c*d - 2*C*a^3*b*c*d + 2*B*a^2*b^2*c*d)/(b*f
*(a*b^2 + b^3*tan(e + f*x))*(a^2 + b^2))
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e))**2,x)
```

```
[Out] Timed out
```

3.63 $\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$

Optimal. Leaf size=597

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^2}{2bf\left(a^2+b^2\right)(a+b\tan(e+fx))^2}-\frac{(bc-ad)\left(a^4Cd-a^2b^2(d(A-3C)+Bc)+2ab^3(Ac-Bd-cC)+b^4(\right.}{b^3f\left(a^2+b^2\right)^2(a+b\tan(e+fx))}$$

```
[Out] -(a^3*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-3*a*b^2*(c^2*C+2*B*c*d-C*d^2-A*(c^2
-d^2))-3*a^2*b*(2*c*(A-C)*d+B*(c^2-d^2))+b^3*(2*c*(A-C)*d+B*(c^2-d^2)))*x/(
a^2+b^2)^3-(3*a^2*b*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-b^3*(c^2*C+2*B*c*d-C*
d^2-A*(c^2-d^2))+a^3*(2*c*(A-C)*d+B*(c^2-d^2))-3*a*b^2*(2*c*(A-C)*d+B*(c^2-
d^2)))*ln(cos(f*x+e))/(a^2+b^2)^3/f+(a^6*C*d^2+3*a^4*b^2*C*d^2-3*a^2*b^4*(c
^2*C+2*B*c*d-2*C*d^2-A*(c^2-d^2))+b^6*(c*(2*B*d+C*c)-A*(c^2-d^2))-a^3*b^3*(
2*c*(A-C)*d+B*(c^2-d^2))+3*a*b^5*(2*c*(A-C)*d+B*(c^2-d^2)))*ln(a+b*tan(f*x+
e))/b^3/(a^2+b^2)^3/f-(-a*d+b*c)*(a^4*C*d+b^4*(A*d+B*c)+2*a*b^3*(A*c-B*d-C*
c)-a^2*b^2*(B*c+(A-3*C)*d))/b^3/(a^2+b^2)^2/f/(a+b*tan(f*x+e))-1/2*(A*b^2-a
*(B*b-C*a))*(c+d*tan(f*x+e))^2/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^2
```

Rubi [A] time = 1.29, antiderivative size = 597, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3645, 3635, 3626, 3617, 31, 3475}

$$\frac{\left(-a^3b^3\left(2cd(A-C)+B\left(c^2-d^2\right)\right)-3a^2b^4\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-2Cd^2\right)+3a^4b^2Cd^2+a^6Cd^2+3ab^5\left(2\right.\right.}{b^3f\left(a^2+b^2\right)^3}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x])^3,x]
```

```
[Out] -(((a^3*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - 3*a*b^2*(c^2*C + 2*B*c*
d - C*d^2 - A*(c^2 - d^2)) - 3*a^2*b*(2*c*(A - C)*d + B*(c^2 - d^2)) + b^3*
(2*c*(A - C)*d + B*(c^2 - d^2)))*x)/(a^2 + b^2)^3) - ((3*a^2*b*(c^2*C + 2*B
*c*d - C*d^2 - A*(c^2 - d^2)) - b^3*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2
)) + a^3*(2*c*(A - C)*d + B*(c^2 - d^2)) - 3*a*b^2*(2*c*(A - C)*d + B*(c^2
- d^2)))*Log[Cos[e + f*x]])/((a^2 + b^2)^3*f) + ((a^6*C*d^2 + 3*a^4*b^2*C*d
^2 - 3*a^2*b^4*(c^2*C + 2*B*c*d - 2*C*d^2 - A*(c^2 - d^2)) + b^6*(c*(c*C +
2*B*d) - A*(c^2 - d^2)) - a^3*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)) + 3*a*b^5
*(2*c*(A - C)*d + B*(c^2 - d^2)))*Log[a + b*Tan[e + f*x]])/(b^3*(a^2 + b^2)
^3*f) - ((b*c - a*d)*(a^4*C*d + b^4*(B*c + A*d) + 2*a*b^3*(A*c - c*C - B*d)
 - a^2*b^2*(B*c + (A - 3*C)*d)))/(b^3*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x]))
 - ((A*b^2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^2)/(2*b*(a^2 + b^2)*f*(a +
 b*Tan[e + f*x])^2)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
```

```
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^2}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2} + \cdots$$

$$= -\frac{(bc-ad)\left(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac \cdots\right.}{b^3\left(a^2+b^2\right)^2 f(a+ \cdots}$$

$$= -\frac{\left(a^3\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-3a \cdots\right.}{\cdots}$$

$$= -\frac{\left(a^3\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-3a \cdots\right.}{\cdots}$$

$$= -\frac{\left(a^3\left(c^2C+2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-3a \cdots\right.}{\cdots}$$

Mathematica [C] time = 8.16, size = 2499, normalized size = 4.19

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x])^3,x]
```

```
[Out] ((-(A*b^4*c^2) + a*b^3*B*c^2 - a^2*b^2*c^2*C + 2*a*A*b^3*c*d - 2*a^2*b^2*B*
c*d + 2*a^3*b*c*C*d - a^2*A*b^2*d^2 + a^3*b*B*d^2 - a^4*C*d^2)*Sec[e + f*x]
*(a*Cos[e + f*x] + b*Sin[e + f*x])*(c + d*Tan[e + f*x])^2)/(2*(a - I*b)^2*(
a + I*b)^2*b*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^3)
+ ((a^3*A*c^2 - 3*a*A*b^2*c^2 + 3*a^2*b*B*c^2 - b^3*B*c^2 - a^3*c^2*C + 3*a
*b^2*c^2*C + 6*a^2*A*b*c*d - 2*A*b^3*c*d - 2*a^3*B*c*d + 6*a*b^2*B*c*d - 6*
```

```
a^2*b*c*C*d + 2*b^3*c*C*d - a^3*A*d^2 + 3*a*A*b^2*d^2 - 3*a^2*b*B*d^2 + b^3
*B*d^2 + a^3*C*d^2 - 3*a*b^2*C*d^2)*(e + f*x)*Sec[e + f*x]*(a*Cos[e + f*x]
+ b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^2)/((a - I*b)^3*(a + I*b)^3*f*(c*C
os[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^3) + (((3*I)*a^9*A*b^6
*c^2 + 3*a^8*A*b^7*c^2 + (5*I)*a^7*A*b^8*c^2 + 5*a^6*A*b^9*c^2 + I*a^5*A*b^
10*c^2 + a^4*A*b^11*c^2 - I*a^3*A*b^12*c^2 - a^2*A*b^13*c^2 - I*a^10*b^5*B*
c^2 - a^9*b^6*B*c^2 + I*a^8*b^7*B*c^2 + a^7*b^8*B*c^2 + (5*I)*a^6*b^9*B*c^2
 + 5*a^5*b^10*B*c^2 + (3*I)*a^4*b^11*B*c^2 + 3*a^3*b^12*B*c^2 - (3*I)*a^9*b
^6*c^2*C - 3*a^8*b^7*c^2*C - (5*I)*a^7*b^8*c^2*C - 5*a^6*b^9*c^2*C - I*a^5*
b^10*c^2*C - a^4*b^11*c^2*C + I*a^3*b^12*c^2*C + a^2*b^13*c^2*C - (2*I)*a^1
0*A*b^5*c*d - 2*a^9*A*b^6*c*d + (2*I)*a^8*A*b^7*c*d + 2*a^7*A*b^8*c*d + (10
*I)*a^6*A*b^9*c*d + 10*a^5*A*b^10*c*d + (6*I)*a^4*A*b^11*c*d + 6*a^3*A*b^12
*c*d - (6*I)*a^9*b^6*B*c*d - 6*a^8*b^7*B*c*d - (10*I)*a^7*b^8*B*c*d - 10*a^
6*b^9*B*c*d - (2*I)*a^5*b^10*B*c*d - 2*a^4*b^11*B*c*d + (2*I)*a^3*b^12*B*c*
d + 2*a^2*b^13*B*c*d + (2*I)*a^10*b^5*c*C*d + 2*a^9*b^6*c*C*d - (2*I)*a^8*b
^7*c*C*d - 2*a^7*b^8*c*C*d - (10*I)*a^6*b^9*c*C*d - 10*a^5*b^10*c*C*d - (6*
I)*a^4*b^11*c*C*d - 6*a^3*b^12*c*C*d - (3*I)*a^9*A*b^6*d^2 - 3*a^8*A*b^7*d^
2 - (5*I)*a^7*A*b^8*d^2 - 5*a^6*A*b^9*d^2 - I*a^5*A*b^10*d^2 - a^4*A*b^11*d
^2 + I*a^3*A*b^12*d^2 + a^2*A*b^13*d^2 + I*a^10*b^5*B*d^2 + a^9*b^6*B*d^2 -
 I*a^8*b^7*B*d^2 - a^7*b^8*B*d^2 - (5*I)*a^6*b^9*B*d^2 - 5*a^5*b^10*B*d^2 -
 (3*I)*a^4*b^11*B*d^2 - 3*a^3*b^12*B*d^2 + I*a^13*b^2*C*d^2 + a^12*b^3*C*d^
2 + (5*I)*a^11*b^4*C*d^2 + 5*a^10*b^5*C*d^2 + (13*I)*a^9*b^6*C*d^2 + 13*a^8
*b^7*C*d^2 + (15*I)*a^7*b^8*C*d^2 + 15*a^6*b^9*C*d^2 + (6*I)*a^5*b^10*C*d^2
 + 6*a^4*b^11*C*d^2)*(e + f*x)*Sec[e + f*x]*(a*Cos[e + f*x] + b*Sin[e + f*x
])^3*(c + d*Tan[e + f*x])^2)/(a^2*(a - I*b)^6*(a + I*b)^5*b^5*f*(c*Cos[e +
f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^3) - (I*(3*a^2*A*b^4*c^2 - A*
b^6*c^2 - a^3*b^3*B*c^2 + 3*a*b^5*B*c^2 - 3*a^2*b^4*c^2*C + b^6*c^2*C - 2*a
^3*A*b^3*c*d + 6*a*A*b^5*c*d - 6*a^2*b^4*B*c*d + 2*b^6*B*c*d + 2*a^3*b^3*c*
C*d - 6*a*b^5*c*C*d - 3*a^2*A*b^4*d^2 + A*b^6*d^2 + a^3*b^3*B*d^2 - 3*a*b^5
*B*d^2 + a^6*C*d^2 + 3*a^4*b^2*C*d^2 + 6*a^2*b^4*C*d^2)*ArcTan[Tan[e + f*x]
]*Sec[e + f*x]*(a*Cos[e + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^2)/
(b^3*(a^2 + b^2)^3*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x
])^3) - (C*d^2*Log[Cos[e + f*x]]*Sec[e + f*x]*(a*Cos[e + f*x] + b*Sin[e + f
*x])^3*(c + d*Tan[e + f*x])^2)/(b^3*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(
a + b*Tan[e + f*x])^3) + ((3*a^2*A*b^4*c^2 - A*b^6*c^2 - a^3*b^3*B*c^2 + 3*
a*b^5*B*c^2 - 3*a^2*b^4*c^2*C + b^6*c^2*C - 2*a^3*A*b^3*c*d + 6*a*A*b^5*c*d
 - 6*a^2*b^4*B*c*d + 2*b^6*B*c*d + 2*a^3*b^3*c*C*d - 6*a*b^5*c*C*d - 3*a^2*
A*b^4*d^2 + A*b^6*d^2 + a^3*b^3*B*d^2 - 3*a*b^5*B*d^2 + a^6*C*d^2 + 3*a^4*b
^2*C*d^2 + 6*a^2*b^4*C*d^2)*Log[(a*Cos[e + f*x] + b*Sin[e + f*x])^2]*Sec[e
+ f*x]*(a*Cos[e + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^2)/(2*b^3*(
a^2 + b^2)^3*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^3)
+ (Sec[e + f*x]*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(3*a*A*b^4*c^2*Sin[e +
f*x] - 2*a^2*b^3*B*c^2*Sin[e + f*x] + b^5*B*c^2*Sin[e + f*x] + a^3*b^2*c^2*
C*Sin[e + f*x] - 2*a*b^4*c^2*C*Sin[e + f*x] - 4*a^2*A*b^3*c*d*Sin[e + f*x]
+ 2*A*b^5*c*d*Sin[e + f*x] + 2*a^3*b^2*B*c*d*Sin[e + f*x] - 4*a*b^4*B*c*d*S
```

```
in[e + f*x] + 6*a^2*b^3*c*C*d*Sin[e + f*x] + a^3*A*b^2*d^2*Sin[e + f*x] - 2
*a*A*b^4*d^2*Sin[e + f*x] + 3*a^2*b^3*B*d^2*Sin[e + f*x] - a^5*C*d^2*Sin[e
+ f*x] - 4*a^3*b^2*C*d^2*Sin[e + f*x])*(c + d*Tan[e + f*x])^2)/(a*(a - I*b)
^2*(a + I*b)^2*b^2*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x
])^3)
```

fricas [B] time = 2.65, size = 1699, normalized size = 2.85

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] -1/2*((3*C*a^4*b^4 - 5*B*a^3*b^5 + (7*A - 3*C)*a^2*b^6 + B*a*b^7 + A*b^8)*c
^2 - 2*(C*a^5*b^3 - 3*B*a^4*b^4 + 5*(A - C)*a^3*b^5 + 3*B*a^2*b^6 - A*a*b^7
)*c*d - (C*a^6*b^2 + B*a^5*b^3 - (3*A - 7*C)*a^4*b^4 - 5*B*a^3*b^5 + 3*A*a^
2*b^6)*d^2 - 2*(((A - C)*a^5*b^3 + 3*B*a^4*b^4 - 3*(A - C)*a^3*b^5 - B*a^2*
b^6)*c^2 - 2*(B*a^5*b^3 - 3*(A - C)*a^4*b^4 - 3*B*a^3*b^5 + (A - C)*a^2*b^6
)*c*d - ((A - C)*a^5*b^3 + 3*B*a^4*b^4 - 3*(A - C)*a^3*b^5 - B*a^2*b^6)*d^2
)*f*x - ((C*a^4*b^4 - 3*B*a^3*b^5 + 5*(A - C)*a^2*b^6 + 3*B*a*b^7 - A*b^8)*
c^2 + 2*(C*a^5*b^3 + B*a^4*b^4 - (3*A - 7*C)*a^3*b^5 - 5*B*a^2*b^6 + 3*A*a*
b^7)*c*d - (3*C*a^6*b^2 - B*a^5*b^3 - (A - 9*C)*a^4*b^4 - 7*B*a^3*b^5 + 5*A
*a^2*b^6)*d^2 + 2*(((A - C)*a^3*b^5 + 3*B*a^2*b^6 - 3*(A - C)*a*b^7 - B*b^8
)*c^2 - 2*(B*a^3*b^5 - 3*(A - C)*a^2*b^6 - 3*B*a*b^7 + (A - C)*b^8)*c*d - (
(A - C)*a^3*b^5 + 3*B*a^2*b^6 - 3*(A - C)*a*b^7 - B*b^8)*d^2)*f*x)*tan(f*x
+ e)^2 + ((B*a^5*b^3 - 3*(A - C)*a^4*b^4 - 3*B*a^3*b^5 + (A - C)*a^2*b^6)*c
^2 + 2*((A - C)*a^5*b^3 + 3*B*a^4*b^4 - 3*(A - C)*a^3*b^5 - B*a^2*b^6)*c*d
- (C*a^8 + 3*C*a^6*b^2 + B*a^5*b^3 - 3*(A - 2*C)*a^4*b^4 - 3*B*a^3*b^5 + A*
a^2*b^6)*d^2 + ((B*a^3*b^5 - 3*(A - C)*a^2*b^6 - 3*B*a*b^7 + (A - C)*b^8)*c
^2 + 2*((A - C)*a^3*b^5 + 3*B*a^2*b^6 - 3*(A - C)*a*b^7 - B*b^8)*c*d - (C*a
^6*b^2 + 3*C*a^4*b^4 + B*a^3*b^5 - 3*(A - 2*C)*a^2*b^6 - 3*B*a*b^7 + A*b^8)
*d^2)*tan(f*x + e)^2 + 2*((B*a^4*b^4 - 3*(A - C)*a^3*b^5 - 3*B*a^2*b^6 + (A
 - C)*a*b^7)*c^2 + 2*((A - C)*a^4*b^4 + 3*B*a^3*b^5 - 3*(A - C)*a^2*b^6 - B
*a*b^7)*c*d - (C*a^7*b + 3*C*a^5*b^3 + B*a^4*b^4 - 3*(A - 2*C)*a^3*b^5 - 3*
B*a^2*b^6 + A*a*b^7)*d^2)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan
(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) + ((C*a^6*b^2 + 3*C*a^4*b^4 + 3*C*a^
2*b^6 + C*b^8)*d^2*tan(f*x + e)^2 + 2*(C*a^7*b + 3*C*a^5*b^3 + 3*C*a^3*b^5
+ C*a*b^7)*d^2*tan(f*x + e) + (C*a^8 + 3*C*a^6*b^2 + 3*C*a^4*b^4 + C*a^2*b^
6)*d^2)*log(1/(tan(f*x + e)^2 + 1)) - 2*((C*a^5*b^3 - 2*B*a^4*b^4 + 3*(A -
C)*a^3*b^5 + 3*B*a^2*b^6 - (3*A - 2*C)*a*b^7 - B*b^8)*c^2 + 2*(B*a^5*b^3 -
(2*A - 3*C)*a^4*b^4 - 3*B*a^3*b^5 + 3*(A - C)*a^2*b^6 + 2*B*a*b^7 - A*b^8)*
c*d - (C*a^7*b - (A - 3*C)*a^5*b^3 - 3*B*a^4*b^4 + (3*A - 4*C)*a^3*b^5 + 3*
B*a^2*b^6 - 2*A*a*b^7)*d^2 + 2*(((A - C)*a^4*b^4 + 3*B*a^3*b^5 - 3*(A - C)*
a^2*b^6 - B*a*b^7)*c^2 - 2*(B*a^4*b^4 - 3*(A - C)*a^3*b^5 - 3*B*a^2*b^6 + (
```

```
A - C)*a*b^7)*c*d - ((A - C)*a^4*b^4 + 3*B*a^3*b^5 - 3*(A - C)*a^2*b^6 - B*
a*b^7)*d^2)*f*x)*tan(f*x + e))/((a^6*b^5 + 3*a^4*b^7 + 3*a^2*b^9 + b^11)*f*
tan(f*x + e)^2 + 2*(a^7*b^4 + 3*a^5*b^6 + 3*a^3*b^8 + a*b^10)*f*tan(f*x + e
) + (a^8*b^3 + 3*a^6*b^5 + 3*a^4*b^7 + a^2*b^9)*f)
```

giac [B] time = 2.70, size = 1714, normalized size = 2.87

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^3*c^2 - C*a^3*c^2 + 3*B*a^2*b*c^2 - 3*A*a*b^2*c^2 + 3*C*a*b^2*c
^2 - B*b^3*c^2 - 2*B*a^3*c*d + 6*A*a^2*b*c*d - 6*C*a^2*b*c*d + 6*B*a*b^2*c*
d - 2*A*b^3*c*d + 2*C*b^3*c*d - A*a^3*d^2 + C*a^3*d^2 - 3*B*a^2*b*d^2 + 3*A
*a*b^2*d^2 - 3*C*a*b^2*d^2 + B*b^3*d^2)*(f*x + e)/(a^6 + 3*a^4*b^2 + 3*a^2*
b^4 + b^6) + (B*a^3*c^2 - 3*A*a^2*b*c^2 + 3*C*a^2*b*c^2 - 3*B*a*b^2*c^2 + A
*b^3*c^2 - C*b^3*c^2 + 2*A*a^3*c*d - 2*C*a^3*c*d + 6*B*a^2*b*c*d - 6*A*a*b^
2*c*d + 6*C*a*b^2*c*d - 2*B*b^3*c*d - B*a^3*d^2 + 3*A*a^2*b*d^2 - 3*C*a^2*b
*d^2 + 3*B*a*b^2*d^2 - A*b^3*d^2 + C*b^3*d^2)*log(tan(f*x + e)^2 + 1)/(a^6
+ 3*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(B*a^3*b^3*c^2 - 3*A*a^2*b^4*c^2 + 3*C*a
^2*b^4*c^2 - 3*B*a*b^5*c^2 + A*b^6*c^2 - C*b^6*c^2 + 2*A*a^3*b^3*c*d - 2*C*
a^3*b^3*c*d + 6*B*a^2*b^4*c*d - 6*A*a*b^5*c*d + 6*C*a*b^5*c*d - 2*B*b^6*c*d
 - C*a^6*d^2 - 3*C*a^4*b^2*d^2 - B*a^3*b^3*d^2 + 3*A*a^2*b^4*d^2 - 6*C*a^2*
b^4*d^2 + 3*B*a*b^5*d^2 - A*b^6*d^2)*log(abs(b*tan(f*x + e) + a))/(a^6*b^3
+ 3*a^4*b^5 + 3*a^2*b^7 + b^9) + (3*B*a^3*b^4*c^2*tan(f*x + e)^2 - 9*A*a^2*
b^5*c^2*tan(f*x + e)^2 + 9*C*a^2*b^5*c^2*tan(f*x + e)^2 - 9*B*a*b^6*c^2*tan
(f*x + e)^2 + 3*A*b^7*c^2*tan(f*x + e)^2 - 3*C*b^7*c^2*tan(f*x + e)^2 + 6*A
*a^3*b^4*c*d*tan(f*x + e)^2 - 6*C*a^3*b^4*c*d*tan(f*x + e)^2 + 18*B*a^2*b^5
*c*d*tan(f*x + e)^2 - 18*A*a*b^6*c*d*tan(f*x + e)^2 + 18*C*a*b^6*c*d*tan(f*
x + e)^2 - 6*B*b^7*c*d*tan(f*x + e)^2 - 3*C*a^6*b*d^2*tan(f*x + e)^2 - 9*C*
a^4*b^3*d^2*tan(f*x + e)^2 - 3*B*a^3*b^4*d^2*tan(f*x + e)^2 + 9*A*a^2*b^5*d
^2*tan(f*x + e)^2 - 18*C*a^2*b^5*d^2*tan(f*x + e)^2 + 9*B*a*b^6*d^2*tan(f*x
 + e)^2 - 3*A*b^7*d^2*tan(f*x + e)^2 + 8*B*a^4*b^3*c^2*tan(f*x + e) - 22*A*
a^3*b^4*c^2*tan(f*x + e) + 22*C*a^3*b^4*c^2*tan(f*x + e) - 18*B*a^2*b^5*c^2
*tan(f*x + e) + 2*A*a*b^6*c^2*tan(f*x + e) - 2*C*a*b^6*c^2*tan(f*x + e) - 2
*B*b^7*c^2*tan(f*x + e) - 4*C*a^6*b*c*d*tan(f*x + e) + 16*A*a^4*b^3*c*d*tan
(f*x + e) - 28*C*a^4*b^3*c*d*tan(f*x + e) + 44*B*a^3*b^4*c*d*tan(f*x + e) -
 36*A*a^2*b^5*c*d*tan(f*x + e) + 24*C*a^2*b^5*c*d*tan(f*x + e) - 4*B*a*b^6*
c*d*tan(f*x + e) - 4*A*b^7*c*d*tan(f*x + e) - 2*C*a^7*d^2*tan(f*x + e) - 2*
B*a^6*b*d^2*tan(f*x + e) - 6*C*a^5*b^2*d^2*tan(f*x + e) - 14*B*a^4*b^3*d^2*
tan(f*x + e) + 22*A*a^3*b^4*d^2*tan(f*x + e) - 28*C*a^3*b^4*d^2*tan(f*x + e
) + 12*B*a^2*b^5*d^2*tan(f*x + e) - 2*A*a*b^6*d^2*tan(f*x + e) - C*a^6*b*c^
2 + 6*B*a^5*b^2*c^2 - 14*A*a^4*b^3*c^2 + 11*C*a^4*b^3*c^2 - 7*B*a^3*b^4*c^2
```

```
 - 3*A*a^2*b^5*c^2 - B*a*b^6*c^2 - A*b^7*c^2 - 2*C*a^7*c*d - 2*B*a^6*b*c*d
+ 12*A*a^5*b^2*c*d - 18*C*a^5*b^2*c*d + 22*B*a^4*b^3*c*d - 14*A*a^3*b^4*c*d
 + 8*C*a^3*b^4*c*d - 2*A*a*b^6*c*d - B*a^7*d^2 - A*a^6*b*d^2 + C*a^6*b*d^2
- 9*B*a^5*b^2*d^2 + 11*A*a^4*b^3*d^2 - 11*C*a^4*b^3*d^2 + 4*B*a^3*b^4*d^2)/
((a^6*b^2 + 3*a^4*b^4 + 3*a^2*b^6 + b^8)*(b*tan(f*x + e) + a)^2))/f
```

maple [B] time = 0.33, size = 2465, normalized size = 4.13

Expression too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3,x
)
```

```
[Out] -3/f/(a^2+b^2)^3*b*ln(a+b*tan(f*x+e))*A*a^2*d^2+3/f/(a^2+b^2)^3*b^2*ln(a+b*
tan(f*x+e))*a*B*c^2-3/f/(a^2+b^2)^3*b^2*ln(a+b*tan(f*x+e))*B*a*d^2+2/f/(a^2
+b^2)^3*b^3*ln(a+b*tan(f*x+e))*B*c*d+1/f/(a^2+b^2)^3/b^3*ln(a+b*tan(f*x+e))
*a^6*C*d^2+3/f/(a^2+b^2)^3/b*ln(a+b*tan(f*x+e))*a^4*C*d^2-3/f/(a^2+b^2)^3*b
*ln(a+b*tan(f*x+e))*C*a^2*c^2+6/f/(a^2+b^2)^3*b*ln(a+b*tan(f*x+e))*C*a^2*d^
2-6/f/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a^2*c*d-2/f/(a^2+b^2)^3*ln(a+b*tan(f*x
+e))*A*a^3*c*d+2/f*b/(a^2+b^2)^2/(a+b*tan(f*x+e))*A*a*d^2+2/f/(a^2+b^2)^3*C
*arctan(tan(f*x+e))*b^3*c*d+2/f/(a^2+b^2)^2/(a+b*tan(f*x+e))*A*a^2*c*d+1/f/
(a^2+b^2)/(a+b*tan(f*x+e))^2*A*a*c*d-1/2/f/b/(a^2+b^2)/(a+b*tan(f*x+e))^2*C
*a^2*c^2-2/f*b/(a^2+b^2)^2/(a+b*tan(f*x+e))*A*a*c^2+2/f/(a^2+b^2)^3*ln(a+b*
tan(f*x+e))*C*a^3*c*d-1/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*a^3*c*d+1/f/(a^2
+b^2)^3*ln(1+tan(f*x+e)^2)*A*a^3*c*d-3/f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*a
*b^2*c^2+3/f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*a*b^2*d^2-2/f/(a^2+b^2)^3*A*a
rctan(tan(f*x+e))*b^3*c*d-2/f/(a^2+b^2)^3*B*arctan(tan(f*x+e))*a^3*c*d+3/f/
(a^2+b^2)^3*B*arctan(tan(f*x+e))*a^2*b*c^2-3/f/(a^2+b^2)^3*B*arctan(tan(f*x
+e))*a^2*b*d^2+3/f/(a^2+b^2)^3*C*arctan(tan(f*x+e))*a*b^2*c^2-3/f/(a^2+b^2)
^3*C*arctan(tan(f*x+e))*a*b^2*d^2+3/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*a*b^
2*c*d+6/f/(a^2+b^2)^3*B*arctan(tan(f*x+e))*a*b^2*c*d-6/f/(a^2+b^2)^3*C*arct
an(tan(f*x+e))*a^2*b*c*d+1/f/b^2/(a^2+b^2)/(a+b*tan(f*x+e))^2*C*a^3*c*d+4/f
*b/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a*c*d+6/f/(a^2+b^2)^3*b^2*ln(a+b*tan(f*x+
e))*A*a*c*d-6/f/(a^2+b^2)^3*b*ln(a+b*tan(f*x+e))*B*a^2*c*d-3/f/(a^2+b^2)^3*
ln(1+tan(f*x+e)^2)*A*a*b^2*c*d+3/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*B*a^2*b*c
*d-2/f/b^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a^4*c*d+6/f/(a^2+b^2)^3*A*arctan(
tan(f*x+e))*a^2*b*c*d-1/f/b/(a^2+b^2)/(a+b*tan(f*x+e))^2*B*a^2*c*d-6/f/(a^2
+b^2)^3*b^2*ln(a+b*tan(f*x+e))*C*a*c*d-2/f*b^2/(a^2+b^2)^2/(a+b*tan(f*x+e))
*A*c*d-1/f/b^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a^4*d^2-3/2/f/(a^2+b^2)^3*ln(
1+tan(f*x+e)^2)*B*a*b^2*c^2+3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*B*a*b^2*d^
2-1/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*B*b^3*c*d+1/2/f/b^2/(a^2+b^2)/(a+b*tan
(f*x+e))^2*B*a^3*d^2-1/2/f/b^3/(a^2+b^2)/(a+b*tan(f*x+e))^2*C*a^4*d^2+2/f/b
^3/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a^5*d^2+4/f/b/(a^2+b^2)^2/(a+b*tan(f*x+e)
)*C*a^3*d^2+2/f*b/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a*c^2+3/2/f/(a^2+b^2)^3*ln
```

```
(1+tan(f*x+e)^2)*C*a^2*b*c^2-3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*a^2*b*d
^2-3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A*a^2*b*c^2+3/2/f/(a^2+b^2)^3*ln(1+
tan(f*x+e)^2)*A*a^2*b*d^2-1/2/f/b/(a^2+b^2)/(a+b*tan(f*x+e))^2*A*a^2*d^2+3/
f/(a^2+b^2)^3*b*ln(a+b*tan(f*x+e))*A*a^2*c^2-1/f/(a^2+b^2)^3*B*arctan(tan(f
*x+e))*b^3*c^2+1/f/(a^2+b^2)^3*B*arctan(tan(f*x+e))*b^3*d^2-1/f/(a^2+b^2)^3
*C*arctan(tan(f*x+e))*a^3*c^2+1/f/(a^2+b^2)^3*C*arctan(tan(f*x+e))*a^3*d^2+
1/f/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a^2*c^2+1/2/f/(a^2+b^2)/(a+b*tan(f*x+e))
^2*B*a*c^2-3/f/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a^2*d^2-1/f/(a^2+b^2)^3*ln(a+
b*tan(f*x+e))*B*a^3*c^2+1/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^3*d^2-1/2/f*
b/(a^2+b^2)/(a+b*tan(f*x+e))^2*A*c^2+1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A
*b^3*c^2-1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A*b^3*d^2+1/2/f/(a^2+b^2)^3*l
n(1+tan(f*x+e)^2)*B*a^3*c^2-1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*B*a^3*d^2-
1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*b^3*c^2+1/2/f/(a^2+b^2)^3*ln(1+tan(f
*x+e)^2)*C*b^3*d^2+1/f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*a^3*c^2-1/f/(a^2+b^
2)^3*A*arctan(tan(f*x+e))*a^3*d^2-1/f*b^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*c^
2-1/f/(a^2+b^2)^3*b^3*ln(a+b*tan(f*x+e))*A*c^2+1/f/(a^2+b^2)^3*b^3*ln(a+b*t
an(f*x+e))*A*d^2+1/f/(a^2+b^2)^3*b^3*ln(a+b*tan(f*x+e))*C*c^2
```

maxima [A] time = 0.57, size = 839, normalized size = 1.41

$$\frac{2\left(\left((A-C)a^3+3\,Ba^2b-3\,(A-C)ab^2-Bb^3\right)c^2-2\left(Ba^3-3\,(A-C)a^2b-3\,Bab^2+(A-C)b^3\right)cd-\left((A-C)a^3+3\,Ba^2b-3\,(A-C)ab^2-Bb^3\right)d^2\right)(fx+e)}{a^6+3\,a^4b^2+3\,a^2b^4+b^6}-\frac{2\left(\left(Ba^3b^3-3\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c^2 - 2*(B*a^3
- 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c*d - ((A - C)*a^3 + 3*B*a^2*b
 - 3*(A - C)*a*b^2 - B*b^3)*d^2)*(f*x + e)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b
^6) - 2*((B*a^3*b^3 - 3*(A - C)*a^2*b^4 - 3*B*a*b^5 + (A - C)*b^6)*c^2 + 2*
((A - C)*a^3*b^3 + 3*B*a^2*b^4 - 3*(A - C)*a*b^5 - B*b^6)*c*d - (C*a^6 + 3*
C*a^4*b^2 + B*a^3*b^3 - 3*(A - 2*C)*a^2*b^4 - 3*B*a*b^5 + A*b^6)*d^2)*log(b
*tan(f*x + e) + a)/(a^6*b^3 + 3*a^4*b^5 + 3*a^2*b^7 + b^9) + ((B*a^3 - 3*(A
 - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c^2 + 2*((A - C)*a^3 + 3*B*a^2*b - 3
*(A - C)*a*b^2 - B*b^3)*c*d - (B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C
)*b^3)*d^2)*log(tan(f*x + e)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - (
(C*a^4*b^2 - 3*B*a^3*b^3 + (5*A - 3*C)*a^2*b^4 + B*a*b^5 + A*b^6)*c^2 + 2*(
C*a^5*b + B*a^4*b^2 - (3*A - 5*C)*a^3*b^3 - 3*B*a^2*b^4 + A*a*b^5)*c*d - (3
*C*a^6 - B*a^5*b - (A - 7*C)*a^4*b^2 - 5*B*a^3*b^3 + 3*A*a^2*b^4)*d^2 - 2*(
(B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^2 - 2*(C*a^4*b^2 - (A - 3*C)*a^2*b^
4 - 2*B*a*b^5 + A*b^6)*c*d + (2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 - 3*B*a^2
*b^4 + 2*A*a*b^5)*d^2)*tan(f*x + e))/(a^6*b^3 + 2*a^4*b^5 + a^2*b^7 + (a^4*
```

```
b^5 + 2*a^2*b^7 + b^9)*tan(f*x + e)^2 + 2*(a^5*b^4 + 2*a^3*b^6 + a*b^8)*tan
(f*x + e)))/f
```

mupad [B] time = 29.28, size = 807, normalized size = 1.35

$$-\frac{\ln\left(a+b\tan\left(e+f\,x\right)\right)\left(\frac{a^2\left(b^4\left(3\,A\,d^2-3\,A\,c^2+3\,C\,c^2-6\,C\,d^2+6\,B\,c\,d\right)+3\,C\,b^4\,d^2\right)-b^6\left(A\,d^2-A\,c^2+C\,c^2+2\,B\,c\,d\right)+C\,b^6\,d^2-a\,b^5\left(3\,B\,c^2-3\,B\,d\right.}{a^6\,b^3+3\,a^4\,b^5+3\,a^2\,b^7+b^9}\right.}{f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x))^3,x)
```

```
[Out] - (log(a + b*tan(e + f*x))*((a^2*(b^4*(3*A*d^2 - 3*A*c^2 + 3*C*c^2 - 6*C*d^
2 + 6*B*c*d) + 3*C*b^4*d^2) - b^6*(A*d^2 - A*c^2 + C*c^2 + 2*B*c*d) + C*b^6
*d^2 - a*b^5*(3*B*c^2 - 3*B*d^2 + 6*A*c*d - 6*C*c*d) + a^3*b^3*(B*c^2 - B*d
^2 + 2*A*c*d - 2*C*c*d))/(b^9 + 3*a^2*b^7 + 3*a^4*b^5 + a^6*b^3) - (C*d^2)/
b^3))/f - ((A*b^6*c^2 - 3*C*a^6*d^2 + B*a*b^5*c^2 + B*a^5*b*d^2 + 5*A*a^2*b
^4*c^2 - 3*A*a^2*b^4*d^2 + A*a^4*b^2*d^2 - 3*B*a^3*b^3*c^2 + 5*B*a^3*b^3*d^
2 - 3*C*a^2*b^4*c^2 + C*a^4*b^2*c^2 - 7*C*a^4*b^2*d^2 + 2*A*a*b^5*c*d + 2*C
*a^5*b*c*d - 6*A*a^3*b^3*c*d - 6*B*a^2*b^4*c*d + 2*B*a^4*b^2*c*d + 10*C*a^3
*b^3*c*d)/(2*b^3*(a^4 + b^4 + 2*a^2*b^2)) + (tan(e + f*x)*(B*b^5*c^2 - 2*C*
a^5*d^2 + 2*A*b^5*c*d + 2*A*a*b^4*c^2 - 2*A*a*b^4*d^2 + B*a^4*b*d^2 - 2*C*a
*b^4*c^2 - B*a^2*b^3*c^2 + 3*B*a^2*b^3*d^2 - 4*C*a^3*b^2*d^2 - 4*B*a*b^4*c*
d + 2*C*a^4*b*c*d - 2*A*a^2*b^3*c*d + 6*C*a^2*b^3*c*d))/(b^2*(a^4 + b^4 + 2
*a^2*b^2)))/(f*(a^2 + b^2*tan(e + f*x)^2 + 2*a*b*tan(e + f*x))) - (log(tan(
e + f*x) - 1i)*(A*d^2*1i - A*c^2*1i + B*c^2 - B*d^2 + C*c^2*1i - C*d^2*1i +
 2*A*c*d + B*c*d*2i - 2*C*c*d))/(2*f*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i)) -
 (log(tan(e + f*x) + 1i)*(A*d^2 - A*c^2 + B*c^2*1i - B*d^2*1i + C*c^2 - C*d
^2 + A*c*d*2i + 2*B*c*d - C*c*d*2i))/(2*f*(a*b^2*3i - 3*a^2*b - a^3*1i + b^
3))
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e))**3,x)
```

```
[Out] Exception raised: AttributeError
```

3.64 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3 \left(A + B\tan(e + fx\right.$

Optimal. Leaf size=603

$$-\frac{d\tan(e+fx)\left(-\left(a^2\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\right)+2ab\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)+b^2\left(2cd(A-C)+\right.\right.}{f}$$

```
[Out] (a^2*(A*c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)+b^2*(c^3*C+3*B*c^2*d
-3*c*C*d^2-B*d^3-A*(c^3-3*c*d^2))-2*a*b*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2
)))*x+(2*a*b*(c^3*C+3*B*c^2*d-3*c*C*d^2-B*d^3-A*(c^3-3*c*d^2))-a^2*((A-C)*d
*(3*c^2-d^2)+B*(c^3-3*c*d^2))+b^2*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2)))*ln
(cos(f*x+e))/f-d*(2*a*b*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-a^2*(2*c*(A-C)*d+
B*(c^2-d^2))+b^2*(2*c*(A-C)*d+B*(c^2-d^2)))*tan(f*x+e)/f+1/2*(2*a*b*(A*c-B*
d-C*c)+a^2*(B*c+(A-C)*d)-b^2*(B*c+(A-C)*d))*(c+d*tan(f*x+e))^2/f+1/3*(a^2*B
-b^2*B+2*a*b*(A-C))*(c+d*tan(f*x+e))^3/f+1/60*(5*a^2*C*d^2-6*a*b*d*(-5*B*d+
C*c)+b^2*(c^2*C-3*B*c*d+15*(A-C)*d^2))*(c+d*tan(f*x+e))^4/d^3/f-1/15*b*(-3*
B*b*d-C*a*d+C*b*c)*tan(f*x+e)*(c+d*tan(f*x+e))^4/d^2/f+1/6*C*(a+b*tan(f*x+e
))^2*(c+d*tan(f*x+e))^4/d/f
```

Rubi [A] time = 1.53, antiderivative size = 603, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3647, 3637, 3630, 3528, 3525, 3475}

$$\frac{(c+d\tan(e+fx))^4\left(5a^2Cd^2-6abd(cC-5Bd)+b^2\left(15d^2(A-C)-3Bcd+c^2C\right)\right)}{60d^3f}-\frac{d\tan(e+fx)\left(a^2\left(-\left(2cd(A\right.\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] (a^2*(A*c^3 - c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3) + b^2*(c^3
*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3 - A*(c^3 - 3*c*d^2)) - 2*a*b*((A - C)*d*
(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)))*x + ((2*a*b*(c^3*C + 3*B*c^2*d - 3*c*C*
d^2 - B*d^3 - A*(c^3 - 3*c*d^2)) - a^2*((A - C)*d*(3*c^2 - d^2) + B*(c^3 -
3*c*d^2)) + b^2*((A - C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)))*Log[Cos[e +
f*x]])/f - (d*(2*a*b*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - a^2*(2*c*(
A - C)*d + B*(c^2 - d^2)) + b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))*Tan[e + f*
x])/f + ((2*a*b*(A*c - c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A -
 C)*d))*(c + d*Tan[e + f*x])^2)/(2*f) + ((a^2*B - b^2*B + 2*a*b*(A - C))*(c
 + d*Tan[e + f*x])^3)/(3*f) + ((5*a^2*C*d^2 - 6*a*b*d*(c*C - 5*B*d) + b^2*(
c^2*C - 3*B*c*d + 15*(A - C)*d^2))*(c + d*Tan[e + f*x])^4)/(60*d^3*f) - (b*
(b*c*C - 3*b*B*d - a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^4)/(15*d^2*f) +
 (C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^4)/(6*d*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
```

```
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{C(a+b\tan(e+fx))^2(c+}{6df}$$

$$= -\frac{b(bcC-3bBd-aCd)\tan}{1}$$

$$= \frac{\left(5a^2Cd^2-6abd(cC-5Bd\right.}{}$$

$$= \frac{\left(a^2B-b^2B+2ab(A-C)\right)}{3f}$$

$$= \frac{\left(2ab(Ac-cC-Bd)+a^2(\right.}{}$$

$$= \left(a^2\left(Ac^3-c^3C-3Bc^2d-\right.\right.$$

$$= \left(a^2\left(Ac^3-c^3C-3Bc^2d-\right.\right.$$

Mathematica [C] time = 6.57, size = 419, normalized size = 0.69

$$\frac{C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^4}{6df}+\frac{-\frac{2b\tan(e+fx)(-aCd-3bBd+bcC)(c+d\tan(e+fx))^4}{5df}-\frac{-\frac{(c+d\tan(e+fx))^4\left(5a^2Cd^2-6abd(cC-5Bd)\right.}{2df}}{}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^4)/(6*d*f) + ((-2*b*(b*c*C -
 3*b*B*d - a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^4)/(5*d*f) - (-1/2*((5*
```

```
a^2*C*d^2 - 6*a*b*d*(c*C - 5*B*d) + b^2*(c^2*C - 3*B*c*d + 15*(A - C)*d^2))
*(c + d*Tan[e + f*x])^4)/(d*f) + (5*(3*d*(2*a*b*(A*c - c*C + B*d) + a^2*(B*
c - (A - C)*d) - b^2*(B*c - (A - C)*d))*((I*c - d)^3*Log[I - Tan[e + f*x]]
- (I*c + d)^3*Log[I + Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3*Tan[e + f*
x]^2) + (a^2*B - b^2*B + 2*a*b*(A - C))*d*((3*I)*(c + I*d)^4*Log[I - Tan[e
+ f*x]] - (3*I)*(c - I*d)^4*Log[I + Tan[e + f*x]] - 6*d^2*(6*c^2 - d^2)*Tan
[e + f*x] - 12*c*d^3*Tan[e + f*x]^2 - 2*d^4*Tan[e + f*x]^3)))/f)/(5*d))/(6*
d)
```

fricas [A] time = 0.64, size = 679, normalized size = 1.13

$$10\,Cb^2d^3\tan\left(fx+e\right)^6+12\left(3\,Cb^2cd^2+\left(2\,Cab+Bb^2\right)d^3\right)\tan\left(fx+e\right)^5+15\left(3\,Cb^2c^2d+3\left(2\,Cab+Bb^2\right)cd^2+\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] 1/60*(10*C*b^2*d^3*tan(f*x + e)^6 + 12*(3*C*b^2*c*d^2 + (2*C*a*b + B*b^2)*d
^3)*tan(f*x + e)^5 + 15*(3*C*b^2*c^2*d + 3*(2*C*a*b + B*b^2)*c*d^2 + (C*a^2
 + 2*B*a*b + (A - C)*b^2)*d^3)*tan(f*x + e)^4 + 20*(C*b^2*c^3 + 3*(2*C*a*b
+ B*b^2)*c^2*d + 3*(C*a^2 + 2*B*a*b + (A - C)*b^2)*c*d^2 + (B*a^2 + 2*(A -
C)*a*b - B*b^2)*d^3)*tan(f*x + e)^3 + 60*(((A - C)*a^2 - 2*B*a*b - (A - C)*
b^2)*c^3 - 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2*d - 3*((A - C)*a^2 - 2*B*a
*b - (A - C)*b^2)*c*d^2 + (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^3)*f*x + 30*((2
*C*a*b + B*b^2)*c^3 + 3*(C*a^2 + 2*B*a*b + (A - C)*b^2)*c^2*d + 3*(B*a^2 +
2*(A - C)*a*b - B*b^2)*c*d^2 + ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^3)*t
an(f*x + e)^2 - 30*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c^3 + 3*((A - C)*a^2 -
2*B*a*b - (A - C)*b^2)*c^2*d - 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d^2 - ((
A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^3)*log(1/(tan(f*x + e)^2 + 1)) + 60*(
(C*a^2 + 2*B*a*b + (A - C)*b^2)*c^3 + 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2
*d + 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^2 - (B*a^2 + 2*(A - C)*a*b
 - B*b^2)*d^3)*tan(f*x + e))/f
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.03, size = 1807, normalized size = 3.00

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

```
[Out] -1/4/f*C*tan(f*x+e)^4*b^2*d^3+1/f*A*b^2*c^3*tan(f*x+e)-1/f*B*a^2*d^3*tan(f*
x+e)-1/f*B*arctan(tan(f*x+e))*b^2*d^3+1/5/f*B*tan(f*x+e)^5*b^2*d^3+1/4/f*A*
tan(f*x+e)^4*b^2*d^3-1/2/f*C*tan(f*x+e)^2*a^2*d^3+1/2/f*C*tan(f*x+e)^2*b^2*
d^3+1/3/f*B*tan(f*x+e)^3*a^2*d^3-1/f*C*b^2*c^3*tan(f*x+e)-3/2/f*ln(1+tan(f*
x+e)^2)*C*a^2*c^2*d-3/f*C*tan(f*x+e)^2*a*b*c*d^2+3/2/f*C*tan(f*x+e)^4*a*b*c
*d^2-3/f*ln(1+tan(f*x+e)^2)*A*a*b*c*d^2-3/f*ln(1+tan(f*x+e)^2)*B*a*b*c^2*d+
3/f*ln(1+tan(f*x+e)^2)*C*a*b*c*d^2-6/f*A*arctan(tan(f*x+e))*a*b*c^2*d+6/f*B
*arctan(tan(f*x+e))*a*b*c*d^2+6/f*C*arctan(tan(f*x+e))*a*b*c^2*d+6/f*A*a*b*
c^2*d*tan(f*x+e)+1/3/f*C*tan(f*x+e)^3*b^2*c^3-1/f*C*arctan(tan(f*x+e))*a^2*
c^3+3/f*B*tan(f*x+e)^2*a*b*c^2*d+2/f*B*tan(f*x+e)^3*a*b*c*d^2+2/f*C*tan(f*x
+e)^3*a*b*c^2*d+3/f*A*tan(f*x+e)^2*a*b*c*d^2-6/f*B*a*b*c*d^2*tan(f*x+e)-6/f
*C*a*b*c^2*d*tan(f*x+e)+1/f*A*arctan(tan(f*x+e))*a^2*c^3+1/2/f*B*tan(f*x+e)
^2*b^2*c^3-1/3/f*B*tan(f*x+e)^3*b^2*d^3+1/f*B*b^2*d^3*tan(f*x+e)+1/f*C*a^2*
c^3*tan(f*x+e)-1/f*A*arctan(tan(f*x+e))*b^2*c^3+1/f*B*arctan(tan(f*x+e))*a^
2*d^3+1/4/f*C*tan(f*x+e)^4*a^2*d^3-3/f*A*arctan(tan(f*x+e))*a^2*c*d^2+1/2/f
*A*tan(f*x+e)^2*a^2*d^3-1/2/f*A*tan(f*x+e)^2*b^2*d^3-1/2/f*ln(1+tan(f*x+e)^
2)*A*a^2*d^3+1/2/f*ln(1+tan(f*x+e)^2)*A*b^2*d^3+1/2/f*ln(1+tan(f*x+e)^2)*B*
a^2*c^3-1/2/f*ln(1+tan(f*x+e)^2)*B*b^2*c^3+1/6/f*C*b^2*d^3*tan(f*x+e)^6+1/f
*C*arctan(tan(f*x+e))*b^2*c^3+1/2/f*ln(1+tan(f*x+e)^2)*C*a^2*d^3-1/2/f*ln(1
+tan(f*x+e)^2)*C*b^2*d^3+3/2/f*A*tan(f*x+e)^2*b^2*c^2*d-3/f*A*b^2*c*d^2*tan
(f*x+e)+2/5/f*C*tan(f*x+e)^5*a*b*d^3+3/2/f*B*tan(f*x+e)^2*a^2*c*d^2-2/3/f*C
*tan(f*x+e)^3*a*b*d^3-1/f*C*tan(f*x+e)^3*b^2*c*d^2+3/2/f*ln(1+tan(f*x+e)^2)
*C*b^2*c^2*d+1/f*ln(1+tan(f*x+e)^2)*B*a*b*d^3+1/f*A*tan(f*x+e)^3*b^2*c*d^2-
3/f*C*arctan(tan(f*x+e))*b^2*c*d^2+3/2/f*ln(1+tan(f*x+e)^2)*B*b^2*c*d^2+1/f
*C*tan(f*x+e)^3*a^2*c*d^2+2/f*B*a*b*c^3*tan(f*x+e)+3/f*A*a^2*c*d^2*tan(f*x+
e)-1/f*B*tan(f*x+e)^2*a*b*d^3+3/4/f*B*tan(f*x+e)^4*b^2*c*d^2-1/f*ln(1+tan(f
*x+e)^2)*C*a*b*c^3-2/f*B*arctan(tan(f*x+e))*a*b*c^3+3/f*B*arctan(tan(f*x+e)
)*b^2*c^2*d+3/2/f*ln(1+tan(f*x+e)^2)*A*a^2*c^2*d+1/2/f*B*tan(f*x+e)^4*a*b*d
^3-2/f*A*a*b*d^3*tan(f*x+e)-3/f*C*a^2*c*d^2*tan(f*x+e)+2/f*C*a*b*d^3*tan(f*
x+e)+1/f*C*tan(f*x+e)^2*a*b*c^3-3/2/f*ln(1+tan(f*x+e)^2)*B*a^2*c*d^2+3/f*C*
b^2*c*d^2*tan(f*x+e)+1/f*B*tan(f*x+e)^3*b^2*c^2*d-3/2/f*ln(1+tan(f*x+e)^2)*
A*b^2*c^2*d+1/f*ln(1+tan(f*x+e)^2)*A*a*b*c^3-3/f*B*arctan(tan(f*x+e))*a^2*c
^2*d+3/f*C*arctan(tan(f*x+e))*a^2*c*d^2-2/f*C*arctan(tan(f*x+e))*a*b*d^3+3/
5/f*C*tan(f*x+e)^5*b^2*c*d^2+2/3/f*A*tan(f*x+e)^3*a*b*d^3-3/2/f*C*tan(f*x+e
)^2*b^2*c^2*d-3/f*B*b^2*c^2*d*tan(f*x+e)+3/f*A*arctan(tan(f*x+e))*b^2*c*d^2
+2/f*A*arctan(tan(f*x+e))*a*b*d^3-3/2/f*B*tan(f*x+e)^2*b^2*c*d^2+3/2/f*C*ta
```

```
n(f*x+e)^2*a^2*c^2*d+3/f*B*a^2*c^2*d*tan(f*x+e)+3/4/f*C*tan(f*x+e)^4*b^2*c^
2*d
```

maxima [A] time = 0.48, size = 680, normalized size = 1.13

$$\frac{10\,Cb^2d^3\tan\left(fx+e\right)^6+12\left(3\,Cb^2cd^2+\left(2\,Cab+Bb^2\right)d^3\right)\tan\left(fx+e\right)^5+15\left(3\,Cb^2c^2d+3\left(2\,Cab+Bb^2\right)cd^2+\cdots\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
```

```
[Out] 1/60*(10*C*b^2*d^3*tan(f*x + e)^6 + 12*(3*C*b^2*c*d^2 + (2*C*a*b + B*b^2)*d
^3)*tan(f*x + e)^5 + 15*(3*C*b^2*c^2*d + 3*(2*C*a*b + B*b^2)*c*d^2 + (C*a^2
 + 2*B*a*b + (A - C)*b^2)*d^3)*tan(f*x + e)^4 + 20*(C*b^2*c^3 + 3*(2*C*a*b
+ B*b^2)*c^2*d + 3*(C*a^2 + 2*B*a*b + (A - C)*b^2)*c*d^2 + (B*a^2 + 2*(A -
C)*a*b - B*b^2)*d^3)*tan(f*x + e)^3 + 30*((2*C*a*b + B*b^2)*c^3 + 3*(C*a^2
+ 2*B*a*b + (A - C)*b^2)*c^2*d + 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d^2 +
((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^3)*tan(f*x + e)^2 + 60*(((A - C)*a^
2 - 2*B*a*b - (A - C)*b^2)*c^3 - 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2*d -
3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^2 + (B*a^2 + 2*(A - C)*a*b - B*
b^2)*d^3)*(f*x + e) + 30*((B*a^2 + 2*(A - C)*a*b - B*b^2)*c^3 + 3*((A - C)*
a^2 - 2*B*a*b - (A - C)*b^2)*c^2*d - 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d^
2 - ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^3)*log(tan(f*x + e)^2 + 1) + 60
*((C*a^2 + 2*B*a*b + (A - C)*b^2)*c^3 + 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c
^2*d + 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^2 - (B*a^2 + 2*(A - C)*a
*b - B*b^2)*d^3)*tan(f*x + e))/f
```

mupad [B] time = 9.31, size = 891, normalized size = 1.48

$$x\left(A\,a^2\,c^3-A\,b^2\,c^3+B\,a^2\,d^3-C\,a^2\,c^3-B\,b^2\,d^3+C\,b^2\,c^3+2\,A\,a\,b\,d^3-2\,B\,a\,b\,c^3-2\,C\,a\,b\,d^3-3\,A\,a^2\,c\,d^2+3\,A\cdots\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] x*(A*a^2*c^3 - A*b^2*c^3 + B*a^2*d^3 - C*a^2*c^3 - B*b^2*d^3 + C*b^2*c^3 +
2*A*a*b*d^3 - 2*B*a*b*c^3 - 2*C*a*b*d^3 - 3*A*a^2*c*d^2 + 3*A*b^2*c*d^2 - 3
*B*a^2*c^2*d + 3*B*b^2*c^2*d + 3*C*a^2*c*d^2 - 3*C*b^2*c*d^2 - 6*A*a*b*c^2*
d + 6*B*a*b*c*d^2 + 6*C*a*b*c^2*d) - (tan(e + f*x)*(B*a^2*d^3 - A*b^2*c^3 -
 b*d^2*(B*b*d + 2*C*a*d + 3*C*b*c) - C*a^2*c^3 + C*b^2*c^3 + 2*A*a*b*d^3 -
2*B*a*b*c^3 - 3*A*a^2*c*d^2 + 3*A*b^2*c*d^2 - 3*B*a^2*c^2*d + 3*B*b^2*c^2*d
```

```
 + 3*C*a^2*c*d^2 - 6*A*a*b*c^2*d + 6*B*a*b*c*d^2 + 6*C*a*b*c^2*d))/f - (log
(tan(e + f*x)^2 + 1)*((A*a^2*d^3)/2 - (B*a^2*c^3)/2 - (A*b^2*d^3)/2 + (B*b^
2*c^3)/2 - (C*a^2*d^3)/2 + (C*b^2*d^3)/2 - A*a*b*c^3 - B*a*b*d^3 + C*a*b*c^
3 - (3*A*a^2*c^2*d)/2 + (3*A*b^2*c^2*d)/2 + (3*B*a^2*c*d^2)/2 - (3*B*b^2*c*
d^2)/2 + (3*C*a^2*c^2*d)/2 - (3*C*b^2*c^2*d)/2 + 3*A*a*b*c*d^2 + 3*B*a*b*c^
2*d - 3*C*a*b*c*d^2))/f + (tan(e + f*x)^4*((A*b^2*d^3)/4 + (C*a^2*d^3)/4 -
(C*b^2*d^3)/4 + (B*a*b*d^3)/2 + (3*B*b^2*c*d^2)/4 + (3*C*b^2*c^2*d)/4 + (3*
C*a*b*c*d^2)/2))/f + (tan(e + f*x)^3*((B*a^2*d^3)/3 - (b*d^2*(B*b*d + 2*C*a
*d + 3*C*b*c))/3 + (C*b^2*c^3)/3 + (2*A*a*b*d^3)/3 + A*b^2*c*d^2 + B*b^2*c^
2*d + C*a^2*c*d^2 + 2*B*a*b*c*d^2 + 2*C*a*b*c^2*d))/f + (tan(e + f*x)^2*((A
*a^2*d^3)/2 - (A*b^2*d^3)/2 + (B*b^2*c^3)/2 - (C*a^2*d^3)/2 + (C*b^2*d^3)/2
 - B*a*b*d^3 + C*a*b*c^3 + (3*A*b^2*c^2*d)/2 + (3*B*a^2*c*d^2)/2 - (3*B*b^2
*c*d^2)/2 + (3*C*a^2*c^2*d)/2 - (3*C*b^2*c^2*d)/2 + 3*A*a*b*c*d^2 + 3*B*a*b
*c^2*d - 3*C*a*b*c*d^2))/f + (b*d^2*tan(e + f*x)^5*(B*b*d + 2*C*a*d + 3*C*b
*c))/(5*f) + (C*b^2*d^3*tan(e + f*x)^6)/(6*f)
```

sympy [A] time = 3.22, size = 1819, normalized size = 3.02

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
```

```
[Out] Piecewise((A*a**2*c**3*x + 3*A*a**2*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) -
 3*A*a**2*c*d**2*x + 3*A*a**2*c*d**2*tan(e + f*x)/f - A*a**2*d**3*log(tan(e
 + f*x)**2 + 1)/(2*f) + A*a**2*d**3*tan(e + f*x)**2/(2*f) + A*a*b*c**3*log(
tan(e + f*x)**2 + 1)/f - 6*A*a*b*c**2*d*x + 6*A*a*b*c**2*d*tan(e + f*x)/f -
 3*A*a*b*c*d**2*log(tan(e + f*x)**2 + 1)/f + 3*A*a*b*c*d**2*tan(e + f*x)**2
/f + 2*A*a*b*d**3*x + 2*A*a*b*d**3*tan(e + f*x)**3/(3*f) - 2*A*a*b*d**3*tan
(e + f*x)/f - A*b**2*c**3*x + A*b**2*c**3*tan(e + f*x)/f - 3*A*b**2*c**2*d*
log(tan(e + f*x)**2 + 1)/(2*f) + 3*A*b**2*c**2*d*tan(e + f*x)**2/(2*f) + 3*
A*b**2*c*d**2*x + A*b**2*c*d**2*tan(e + f*x)**3/f - 3*A*b**2*c*d**2*tan(e +
 f*x)/f + A*b**2*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + A*b**2*d**3*tan(e +
f*x)**4/(4*f) - A*b**2*d**3*tan(e + f*x)**2/(2*f) + B*a**2*c**3*log(tan(e +
 f*x)**2 + 1)/(2*f) - 3*B*a**2*c**2*d*x + 3*B*a**2*c**2*d*tan(e + f*x)/f -
3*B*a**2*c*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*a**2*c*d**2*tan(e + f*
x)**2/(2*f) + B*a**2*d**3*x + B*a**2*d**3*tan(e + f*x)**3/(3*f) - B*a**2*d*
*3*tan(e + f*x)/f - 2*B*a*b*c**3*x + 2*B*a*b*c**3*tan(e + f*x)/f - 3*B*a*b*
c**2*d*log(tan(e + f*x)**2 + 1)/f + 3*B*a*b*c**2*d*tan(e + f*x)**2/f + 6*B*
a*b*c*d**2*x + 2*B*a*b*c*d**2*tan(e + f*x)**3/f - 6*B*a*b*c*d**2*tan(e + f*
x)/f + B*a*b*d**3*log(tan(e + f*x)**2 + 1)/f + B*a*b*d**3*tan(e + f*x)**4/(
2*f) - B*a*b*d**3*tan(e + f*x)**2/f - B*b**2*c**3*log(tan(e + f*x)**2 + 1)/
(2*f) + B*b**2*c**3*tan(e + f*x)**2/(2*f) + 3*B*b**2*c**2*d*x + B*b**2*c**2
*d*tan(e + f*x)**3/f - 3*B*b**2*c**2*d*tan(e + f*x)/f + 3*B*b**2*c*d**2*log
```

```
(tan(e + f*x)**2 + 1)/(2*f) + 3*B*b**2*c*d**2*tan(e + f*x)**4/(4*f) - 3*B*b
**2*c*d**2*tan(e + f*x)**2/(2*f) - B*b**2*d**3*x + B*b**2*d**3*tan(e + f*x)
**5/(5*f) - B*b**2*d**3*tan(e + f*x)**3/(3*f) + B*b**2*d**3*tan(e + f*x)/f
- C*a**2*c**3*x + C*a**2*c**3*tan(e + f*x)/f - 3*C*a**2*c**2*d*log(tan(e +
f*x)**2 + 1)/(2*f) + 3*C*a**2*c**2*d*tan(e + f*x)**2/(2*f) + 3*C*a**2*c*d**
2*x + C*a**2*c*d**2*tan(e + f*x)**3/f - 3*C*a**2*c*d**2*tan(e + f*x)/f + C*
a**2*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + C*a**2*d**3*tan(e + f*x)**4/(4*f
) - C*a**2*d**3*tan(e + f*x)**2/(2*f) - C*a*b*c**3*log(tan(e + f*x)**2 + 1)
/f + C*a*b*c**3*tan(e + f*x)**2/f + 6*C*a*b*c**2*d*x + 2*C*a*b*c**2*d*tan(e
 + f*x)**3/f - 6*C*a*b*c**2*d*tan(e + f*x)/f + 3*C*a*b*c*d**2*log(tan(e + f
*x)**2 + 1)/f + 3*C*a*b*c*d**2*tan(e + f*x)**4/(2*f) - 3*C*a*b*c*d**2*tan(e
 + f*x)**2/f - 2*C*a*b*d**3*x + 2*C*a*b*d**3*tan(e + f*x)**5/(5*f) - 2*C*a*
b*d**3*tan(e + f*x)**3/(3*f) + 2*C*a*b*d**3*tan(e + f*x)/f + C*b**2*c**3*x
+ C*b**2*c**3*tan(e + f*x)**3/(3*f) - C*b**2*c**3*tan(e + f*x)/f + 3*C*b**2
*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*b**2*c**2*d*tan(e + f*x)**4/(4
*f) - 3*C*b**2*c**2*d*tan(e + f*x)**2/(2*f) - 3*C*b**2*c*d**2*x + 3*C*b**2*
c*d**2*tan(e + f*x)**5/(5*f) - C*b**2*c*d**2*tan(e + f*x)**3/f + 3*C*b**2*c
*d**2*tan(e + f*x)/f - C*b**2*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + C*b**2*
d**3*tan(e + f*x)**6/(6*f) - C*b**2*d**3*tan(e + f*x)**4/(4*f) + C*b**2*d**
3*tan(e + f*x)**2/(2*f), Ne(f, 0)), (x*(a + b*tan(e))**2*(c + d*tan(e))**3*
(A + B*tan(e) + C*tan(e)**2), True))
```

3.65 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)\right.$

Optimal. Leaf size=389

$$\frac{d\tan(e+fx)\left(A\left(2acd+b\left(c^2-d^2\right)\right)+a\left(Bc^2-Bd^2-2cCd\right)-b\left(2Bcd+c^2C-Cd^2\right)\right)}{f}-\frac{\log(\cos(e+fx))\left(A\left(3ac\right.\right.}{}$$

```
[Out] (a*(A*c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)-b*((A-C)*d*(3*c^2-d^2)
+B*(c^3-3*c*d^2)))*x-(A*(3*a*c^2*d-a*d^3+b*c^3-3*b*c*d^2)-b*(3*B*c^2*d-B*d^
3+C*c^3-3*C*c*d^2)+a*(B*c^3-3*B*c*d^2-3*C*c^2*d+C*d^3))*ln(cos(f*x+e))/f+d*
(a*(B*c^2-B*d^2-2*C*c*d)-b*(2*B*c*d+C*c^2-C*d^2)+A*(2*a*c*d+b*(c^2-d^2)))*t
an(f*x+e)/f+1/2*(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b*c)*(c+d*tan(f*x+e))^2/f+
1/3*(A*b+B*a-C*b)*(c+d*tan(f*x+e))^3/f-1/20*(-5*B*b*d-5*C*a*d+C*b*c)*(c+d*t
an(f*x+e))^4/d^2/f+1/5*b*C*tan(f*x+e)*(c+d*tan(f*x+e))^4/d/f
```

Rubi [A] time = 0.71, antiderivative size = 387, normalized size of antiderivative = 0.99, number of steps used = 6, number of rules used = 5, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3637, 3630, 3528, 3525, 3475}

$$\frac{d\tan(e+fx)\left(2aAcd+aB\left(c^2-d^2\right)-2acCd+Ab\left(c^2-d^2\right)-b\left(2Bcd+c^2C-Cd^2\right)\right)}{f}-\frac{\log(\cos(e+fx))\left(A\left(3ac\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan
[e + f*x]^2),x]
```

```
[Out] -((b*(A - C)*d*(3*c^2 - d^2) + b*B*(c^3 - 3*c*d^2) - a*(A*c^3 - c^3*C - 3*B
*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3))*x) - ((A*(b*c^3 + 3*a*c^2*d - 3*b*
c*d^2 - a*d^3) - b*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3) + a*(B*c^3 - 3*c
^2*C*d - 3*B*c*d^2 + C*d^3))*Log[Cos[e + f*x]])/f + (d*(2*a*A*c*d - 2*a*c*C
*d + A*b*(c^2 - d^2) + a*B*(c^2 - d^2) - b*(c^2*C + 2*B*c*d - C*d^2))*Tan[e
 + f*x])/f + ((A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d)*(c + d*Tan[e
+ f*x])^2)/(2*f) + ((A*b + a*B - b*C)*(c + d*Tan[e + f*x])^3)/(3*f) - ((b*c
*C - 5*b*B*d - 5*a*C*d)*(c + d*Tan[e + f*x])^4)/(20*d^2*f) + (b*C*Tan[e + f
*x]*(c + d*Tan[e + f*x])^4)/(5*d*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
```

```
f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{bC\tan(e+fx)(c+d\tan(e}{5df}$$
$$= -\frac{(bcC-5bBd-5aCd)(c+}{20d^2f}$$
$$= \frac{(Ab+aB-bC)(c+d\tan(e}{3f}$$
$$= \frac{(Abc+aBc-bcC+aAd-}{}$$
$$= -\left(b(A-C)d\left(3c^2-d^2\right)+b\right.$$
$$= -\left(b(A-C)d\left(3c^2-d^2\right)+b\right.$$

Mathematica [C] time = 6.33, size = 297, normalized size = 0.76

$$\frac{bC\tan(e+fx)(c+d\tan(e+fx))^4}{5df} - \frac{\frac{(-5aCd-5bBd+bcC)(c+d\tan(e+fx))^4}{4df} + \frac{5\left((aB+Ab-bC)\left(-6d^2\left(6c^2-d^2\right)\tan(e+fx)-12cd^3\tan^2(e+fx\right.\right.}{}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] (b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^4)/(5*d*f) - (((b*c*C - 5*b*B*d - 5*
a*C*d)*(c + d*Tan[e + f*x])^4)/(4*d*f) + (5*(3*(A*b*c + a*B*c - b*c*C - a*A
*d + b*B*d + a*C*d)*((I*c - d)^3*Log[I - Tan[e + f*x]] - (I*c + d)^3*Log[I
+ Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3*Tan[e + f*x]^2) + (A*b + a*B -
 b*C)*((3*I)*(c + I*d)^4*Log[I - Tan[e + f*x]] - (3*I)*(c - I*d)^4*Log[I +
Tan[e + f*x]] - 6*d^2*(6*c^2 - d^2)*Tan[e + f*x] - 12*c*d^3*Tan[e + f*x]^2
- 2*d^4*Tan[e + f*x]^3)))/(6*f))/(5*d)
```

fricas [A] time = 1.48, size = 386, normalized size = 0.99

$$\frac{12\,Cbd^3\tan\left(fx+e\right)^5+15\left(3\,Cbcd^2+(Ca+Bb)d^3\right)\tan\left(fx+e\right)^4+20\left(3\,Cbc^2d+3\,(Ca+Bb)cd^2+(Ba+(A-\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="fricas")
```

```
[Out] 1/60*(12*C*b*d^3*tan(f*x + e)^5 + 15*(3*C*b*c*d^2 + (C*a + B*b)*d^3)*tan(f*
x + e)^4 + 20*(3*C*b*c^2*d + 3*(C*a + B*b)*c*d^2 + (B*a + (A - C)*b)*d^3)*t
an(f*x + e)^3 + 60*(((A - C)*a - B*b)*c^3 - 3*(B*a + (A - C)*b)*c^2*d - 3*(
(A - C)*a - B*b)*c*d^2 + (B*a + (A - C)*b)*d^3)*f*x + 30*(C*b*c^3 + 3*(C*a
+ B*b)*c^2*d + 3*(B*a + (A - C)*b)*c*d^2 + ((A - C)*a - B*b)*d^3)*tan(f*x +
 e)^2 - 30*((B*a + (A - C)*b)*c^3 + 3*((A - C)*a - B*b)*c^2*d - 3*(B*a + (A
 - C)*b)*c*d^2 - ((A - C)*a - B*b)*d^3)*log(1/(tan(f*x + e)^2 + 1)) + 60*((
C*a + B*b)*c^3 + 3*(B*a + (A - C)*b)*c^2*d + 3*((A - C)*a - B*b)*c*d^2 - (B
*a + (A - C)*b)*d^3)*tan(f*x + e))/f
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.04, size = 994, normalized size = 2.56

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] -3/f*A*arctan(tan(f*x+e))*a*c*d^2-1/2/f*C*tan(f*x+e)^2*a*d^3+1/f*B*arctan(t
an(f*x+e))*a*d^3-1/2/f*ln(1+tan(f*x+e)^2)*A*a*d^3-1/f*C*arctan(tan(f*x+e))*
a*c^3-3/2/f*ln(1+tan(f*x+e)^2)*B*b*c^2*d-3/2/f*ln(1+tan(f*x+e)^2)*C*a*c^2*d
+3/2/f*A*tan(f*x+e)^2*b*c*d^2-1/f*B*arctan(tan(f*x+e))*b*c^3-1/f*A*b*d^3*ta
n(f*x+e)+1/3/f*B*tan(f*x+e)^3*a*d^3+1/2/f*ln(1+tan(f*x+e)^2)*A*b*c^3+1/2/f*
C*tan(f*x+e)^2*b*c^3+1/2/f*ln(1+tan(f*x+e)^2)*B*a*c^3+1/f*C*a*c^3*tan(f*x+e
)+1/2/f*ln(1+tan(f*x+e)^2)*C*a*d^3+1/3/f*A*tan(f*x+e)^3*b*d^3+1/2/f*A*tan(f
*x+e)^2*a*d^3+1/5/f*C*b*d^3*tan(f*x+e)^5+1/4/f*C*tan(f*x+e)^4*a*d^3+1/f*C*b
*d^3*tan(f*x+e)-1/3/f*C*tan(f*x+e)^3*b*d^3+1/4/f*B*tan(f*x+e)^4*b*d^3+1/f*A
*arctan(tan(f*x+e))*a*c^3+1/f*A*arctan(tan(f*x+e))*b*d^3-1/f*B*a*d^3*tan(f*
x+e)+1/f*B*b*c^3*tan(f*x+e)+1/2/f*ln(1+tan(f*x+e)^2)*B*b*d^3-1/f*C*arctan(t
an(f*x+e))*b*d^3-1/2/f*B*tan(f*x+e)^2*b*d^3-1/2/f*ln(1+tan(f*x+e)^2)*C*b*c^
3-3/f*A*arctan(tan(f*x+e))*b*c^2*d+3/f*C*arctan(tan(f*x+e))*a*c*d^2+3/f*C*a
rctan(tan(f*x+e))*b*c^2*d+3/2/f*ln(1+tan(f*x+e)^2)*A*a*c^2*d-3/2/f*ln(1+tan
(f*x+e)^2)*A*b*c*d^2+3/2/f*B*tan(f*x+e)^2*b*c^2*d+3/2/f*B*tan(f*x+e)^2*a*c*
```

```
d^2-3/2/f*ln(1+tan(f*x+e)^2)*B*a*c*d^2-3/f*C*a*c*d^2*tan(f*x+e)+3/2/f*ln(1+
tan(f*x+e)^2)*C*b*c*d^2+3/f*B*a*c^2*d*tan(f*x+e)-3/f*B*arctan(tan(f*x+e))*a
*c^2*d+3/f*B*arctan(tan(f*x+e))*b*c*d^2-3/2/f*C*tan(f*x+e)^2*b*c*d^2+3/f*A*
a*c*d^2*tan(f*x+e)-3/f*B*b*c*d^2*tan(f*x+e)-3/f*C*b*c^2*d*tan(f*x+e)+1/f*C*
tan(f*x+e)^3*a*c*d^2+1/f*B*tan(f*x+e)^3*b*c*d^2+1/f*C*tan(f*x+e)^3*b*c^2*d+
3/2/f*C*tan(f*x+e)^2*a*c^2*d+3/4/f*C*tan(f*x+e)^4*b*c*d^2+3/f*A*b*c^2*d*tan
(f*x+e)
```

maxima [A] time = 0.45, size = 387, normalized size = 0.99

$$12\,Cbd^3\tan\left(fx+e\right)^5+15\left(3\,Cbcd^2+(Ca+Bb)d^3\right)\tan\left(fx+e\right)^4+20\left(3\,Cbc^2d+3\,(Ca+Bb)cd^2+(Ba+(A-\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="maxima")
```

```
[Out] 1/60*(12*C*b*d^3*tan(f*x + e)^5 + 15*(3*C*b*c*d^2 + (C*a + B*b)*d^3)*tan(f*
x + e)^4 + 20*(3*C*b*c^2*d + 3*(C*a + B*b)*c*d^2 + (B*a + (A - C)*b)*d^3)*t
an(f*x + e)^3 + 30*(C*b*c^3 + 3*(C*a + B*b)*c^2*d + 3*(B*a + (A - C)*b)*c*d
^2 + ((A - C)*a - B*b)*d^3)*tan(f*x + e)^2 + 60*(((A - C)*a - B*b)*c^3 - 3*
(B*a + (A - C)*b)*c^2*d - 3*((A - C)*a - B*b)*c*d^2 + (B*a + (A - C)*b)*d^3
)*(f*x + e) + 30*((B*a + (A - C)*b)*c^3 + 3*((A - C)*a - B*b)*c^2*d - 3*(B*
a + (A - C)*b)*c*d^2 - ((A - C)*a - B*b)*d^3)*log(tan(f*x + e)^2 + 1) + 60*
((C*a + B*b)*c^3 + 3*(B*a + (A - C)*b)*c^2*d + 3*((A - C)*a - B*b)*c*d^2 -
(B*a + (A - C)*b)*d^3)*tan(f*x + e))/f
```

mupad [B] time = 9.04, size = 478, normalized size = 1.23

$$x\left(A\,a\,c^3+A\,b\,d^3+B\,a\,d^3-B\,b\,c^3-C\,a\,c^3-C\,b\,d^3-3\,A\,a\,c\,d^2-3\,A\,b\,c^2\,d-3\,B\,a\,c^2\,d+3\,B\,b\,c\,d^2+3\,C\,a\,c\,d^2+\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2),x)
```

```
[Out] x*(A*a*c^3 + A*b*d^3 + B*a*d^3 - B*b*c^3 - C*a*c^3 - C*b*d^3 - 3*A*a*c*d^2
- 3*A*b*c^2*d - 3*B*a*c^2*d + 3*B*b*c*d^2 + 3*C*a*c*d^2 + 3*C*b*c^2*d) + (t
an(e + f*x)^4*((B*b*d^3)/4 + (C*a*d^3)/4 + (3*C*b*c*d^2)/4))/f + (tan(e + f
*x)^3*((A*b*d^3)/3 + (B*a*d^3)/3 - (C*b*d^3)/3 + B*b*c*d^2 + C*a*c*d^2 + C*
b*c^2*d))/f + (tan(e + f*x)^2*((A*a*d^3)/2 - (B*b*d^3)/2 - (C*a*d^3)/2 + (C
*b*c^3)/2 + (3*A*b*c*d^2)/2 + (3*B*a*c*d^2)/2 + (3*B*b*c^2*d)/2 + (3*C*a*c^
2*d)/2 - (3*C*b*c*d^2)/2))/f - (log(tan(e + f*x)^2 + 1)*((A*a*d^3)/2 - (A*b
```

```
*c^3)/2 - (B*a*c^3)/2 - (B*b*d^3)/2 - (C*a*d^3)/2 + (C*b*c^3)/2 - (3*A*a*c^
2*d)/2 + (3*A*b*c*d^2)/2 + (3*B*a*c*d^2)/2 + (3*B*b*c^2*d)/2 + (3*C*a*c^2*d
)/2 - (3*C*b*c*d^2)/2))/f + (tan(e + f*x)*(B*b*c^3 - B*a*d^3 - A*b*d^3 + C*
a*c^3 + C*b*d^3 + 3*A*a*c*d^2 + 3*A*b*c^2*d + 3*B*a*c^2*d - 3*B*b*c*d^2 - 3
*C*a*c*d^2 - 3*C*b*c^2*d))/f + (C*b*d^3*tan(e + f*x)^5)/(5*f)
```

sympy [A] time = 1.65, size = 1001, normalized size = 2.57

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)
**2),x)
```

```
[Out] Piecewise((A*a*c**3*x + 3*A*a*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) - 3*A*a
*c*d**2*x + 3*A*a*c*d**2*tan(e + f*x)/f - A*a*d**3*log(tan(e + f*x)**2 + 1)
/(2*f) + A*a*d**3*tan(e + f*x)**2/(2*f) + A*b*c**3*log(tan(e + f*x)**2 + 1)
/(2*f) - 3*A*b*c**2*d*x + 3*A*b*c**2*d*tan(e + f*x)/f - 3*A*b*c*d**2*log(ta
n(e + f*x)**2 + 1)/(2*f) + 3*A*b*c*d**2*tan(e + f*x)**2/(2*f) + A*b*d**3*x
+ A*b*d**3*tan(e + f*x)**3/(3*f) - A*b*d**3*tan(e + f*x)/f + B*a*c**3*log(t
an(e + f*x)**2 + 1)/(2*f) - 3*B*a*c**2*d*x + 3*B*a*c**2*d*tan(e + f*x)/f -
3*B*a*c*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*a*c*d**2*tan(e + f*x)**2/
(2*f) + B*a*d**3*x + B*a*d**3*tan(e + f*x)**3/(3*f) - B*a*d**3*tan(e + f*x)
/f - B*b*c**3*x + B*b*c**3*tan(e + f*x)/f - 3*B*b*c**2*d*log(tan(e + f*x)**
2 + 1)/(2*f) + 3*B*b*c**2*d*tan(e + f*x)**2/(2*f) + 3*B*b*c*d**2*x + B*b*c*
d**2*tan(e + f*x)**3/f - 3*B*b*c*d**2*tan(e + f*x)/f + B*b*d**3*log(tan(e +
 f*x)**2 + 1)/(2*f) + B*b*d**3*tan(e + f*x)**4/(4*f) - B*b*d**3*tan(e + f*x
)**2/(2*f) - C*a*c**3*x + C*a*c**3*tan(e + f*x)/f - 3*C*a*c**2*d*log(tan(e
+ f*x)**2 + 1)/(2*f) + 3*C*a*c**2*d*tan(e + f*x)**2/(2*f) + 3*C*a*c*d**2*x
+ C*a*c*d**2*tan(e + f*x)**3/f - 3*C*a*c*d**2*tan(e + f*x)/f + C*a*d**3*log
(tan(e + f*x)**2 + 1)/(2*f) + C*a*d**3*tan(e + f*x)**4/(4*f) - C*a*d**3*tan
(e + f*x)**2/(2*f) - C*b*c**3*log(tan(e + f*x)**2 + 1)/(2*f) + C*b*c**3*tan
(e + f*x)**2/(2*f) + 3*C*b*c**2*d*x + C*b*c**2*d*tan(e + f*x)**3/f - 3*C*b*
c**2*d*tan(e + f*x)/f + 3*C*b*c*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*b
*c*d**2*tan(e + f*x)**4/(4*f) - 3*C*b*c*d**2*tan(e + f*x)**2/(2*f) - C*b*d*
*3*x + C*b*d**3*tan(e + f*x)**5/(5*f) - C*b*d**3*tan(e + f*x)**3/(3*f) + C*
b*d**3*tan(e + f*x)/f, Ne(f, 0)), (x*(a + b*tan(e))*(c + d*tan(e))**3*(A +
B*tan(e) + C*tan(e)**2), True))
```

3.66 $\int (c+d\tan(e+fx))^3 \left(A+B\tan(e+fx)+C\tan^2(e+fx\right.$

Optimal. Leaf size=191

$$\frac{d\tan(e+fx)\left(2cd(A-C)+B\left(c^2-d^2\right)\right)}{f}-\frac{\left(d(A-C)\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)\log(\cos(e+fx))}{f}-x\left(-A\left(c^3-3\right.\right.$$

```
[Out] -(c^3*C+3*B*c^2*d-3*c*C*d^2-B*d^3-A*(c^3-3*c*d^2))*x-((A-C)*d*(3*c^2-d^2)+B
*(c^3-3*c*d^2))*ln(cos(f*x+e))/f+d*(2*c*(A-C)*d+B*(c^2-d^2))*tan(f*x+e)/f+1
/2*(B*c+(A-C)*d)*(c+d*tan(f*x+e))^2/f+1/3*B*(c+d*tan(f*x+e))^3/f+1/4*C*(c+d
*tan(f*x+e))^4/d/f
```

Rubi [A] time = 0.24, antiderivative size = 191, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 4, integrand size = 33, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.121, Rules used = {3630, 3528, 3525, 3475}

$$\frac{d\tan(e+fx)\left(2cd(A-C)+B\left(c^2-d^2\right)\right)}{f}-\frac{\left(d(A-C)\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)\log(\cos(e+fx))}{f}-x\left(-A\left(c^3-3\right.\right.$$

Antiderivative was successfully verified.

```
[In] Int[(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -((c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3 - A*(c^3 - 3*c*d^2))*x) - (((A - C
)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2))*Log[Cos[e + f*x]])/f + (d*(2*c*(A -
C)*d + B*(c^2 - d^2))*Tan[e + f*x])/f + ((B*c + (A - C)*d)*(c + d*Tan[e + f
*x])^2)/(2*f) + (B*(c + d*Tan[e + f*x])^3)/(3*f) + (C*(c + d*Tan[e + f*x])^
4)/(4*d*f)
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3525

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)]), x_Symbol] :> Simp[(a*c - b*d)*x, x] + (Dist[b*c + a*d, Int[Tan[e +
 f*x], x], x] + Simp[(b*d*Tan[e + f*x])/f, x]) /; FreeQ[{a, b, c, d, e, f},
 x] && NeQ[b*c - a*d, 0] && NeQ[b*c + a*d, 0]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
```

```
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int (c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx=\frac{C(c+d\tan(e+fx))^4}{4df}+\int (A-C+B\tan(e$$

$$=\frac{B(c+d\tan(e+fx))^3}{3f}+\frac{C(c+d\tan(e+fx)}{4df}$$

$$=\frac{(Bc+(A-C)d)(c+d\tan(e+fx))^2}{2f}+\frac{B(c+}{}$$

$$=-\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd\right.\right.$$

$$=-\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-3cd\right.\right.$$

Mathematica [C] time = 2.44, size = 212, normalized size = 1.11

$$\frac{-6(d(C-A)+Bc)\left(6cd^2\tan(e+fx)+(-d+ic)^3\log(-\tan(e+fx)+i)-(d+ic)^3\log(\tan(e+fx)+i)+d^3\text{ta}\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (3*C*(c + d*Tan[e + f*x])^4 - 6*(B*c + (-A + C)*d)*((I*c - d)^3*Log[I - Tan
[e + f*x]] - (I*c + d)^3*Log[I + Tan[e + f*x]] + 6*c*d^2*Tan[e + f*x] + d^3
*Tan[e + f*x]^2) + 2*B*((-3*I)*(c + I*d)^4*Log[I - Tan[e + f*x]] + (3*I)*(c
 - I*d)^4*Log[I + Tan[e + f*x]] - 6*d^2*(-6*c^2 + d^2)*Tan[e + f*x] + 12*c*
d^3*Tan[e + f*x]^2 + 2*d^4*Tan[e + f*x]^3))/(12*d*f)
```

fricas [A] time = 1.57, size = 201, normalized size = 1.05

$$3\,Cd^3\tan\left(fx+e\right)^4+4\left(3\,Ccd^2+Bd^3\right)\tan\left(fx+e\right)^3+12\left((A-C)c^3-3\,Bc^2d-3\,(A-C)cd^2+Bd^3\right)fx+6\left(3\,C\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
fricas")
```

```
[Out] 1/12*(3*C*d^3*tan(f*x + e)^4 + 4*(3*C*c*d^2 + B*d^3)*tan(f*x + e)^3 + 12*((
A - C)*c^3 - 3*B*c^2*d - 3*(A - C)*c*d^2 + B*d^3)*f*x + 6*(3*C*c^2*d + 3*B*
c*d^2 + (A - C)*d^3)*tan(f*x + e)^2 - 6*(B*c^3 + 3*(A - C)*c^2*d - 3*B*c*d^
2 - (A - C)*d^3)*log(1/(tan(f*x + e)^2 + 1)) + 12*(C*c^3 + 3*B*c^2*d + 3*(A
 - C)*c*d^2 - B*d^3)*tan(f*x + e))/f
```

giac [B] time = 24.81, size = 4300, normalized size = 22.51

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
giac")
```

```
[Out] 1/12*(12*A*c^3*f*x*tan(f*x)^4*tan(e)^4 - 12*C*c^3*f*x*tan(f*x)^4*tan(e)^4 -
 36*B*c^2*d*f*x*tan(f*x)^4*tan(e)^4 - 36*A*c*d^2*f*x*tan(f*x)^4*tan(e)^4 +
36*C*c*d^2*f*x*tan(f*x)^4*tan(e)^4 + 12*B*d^3*f*x*tan(f*x)^4*tan(e)^4 - 6*B
*c^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*tan(e)^4
- 18*A*c^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^4*t
an(e)^4 + 18*C*c^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(
f*x)^4*tan(e)^4 + 18*B*c*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(
e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 +
1))*tan(f*x)^4*tan(e)^4 + 6*A*d^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3
*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)
^2 + 1))*tan(f*x)^4*tan(e)^4 - 6*C*d^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f
*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(t
an(e)^2 + 1))*tan(f*x)^4*tan(e)^4 - 48*A*c^3*f*x*tan(f*x)^3*tan(e)^3 + 48*C
*c^3*f*x*tan(f*x)^3*tan(e)^3 + 144*B*c^2*d*f*x*tan(f*x)^3*tan(e)^3 + 144*A*
c*d^2*f*x*tan(f*x)^3*tan(e)^3 - 144*C*c*d^2*f*x*tan(f*x)^3*tan(e)^3 - 48*B*
d^3*f*x*tan(f*x)^3*tan(e)^3 + 18*C*c^2*d*tan(f*x)^4*tan(e)^4 + 18*B*c*d^2*t
an(f*x)^4*tan(e)^4 + 6*A*d^3*tan(f*x)^4*tan(e)^4 - 9*C*d^3*tan(f*x)^4*tan(e
```

```
)^4 + 24*B*c^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^
2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3
*tan(e)^3 + 72*A*c^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + t
an(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*ta
n(f*x)^3*tan(e)^3 - 72*C*c^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*ta
n(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2
+ 1))*tan(f*x)^3*tan(e)^3 - 72*B*c*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f
*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(t
an(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 24*A*d^3*log(4*(tan(f*x)^4*tan(e)^2 - 2
*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) +
 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 + 24*C*d^3*log(4*(tan(f*x)^4*tan(e)
^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*ta
n(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^3*tan(e)^3 - 12*C*c^3*tan(f*x)^4*tan(e)^
3 - 36*B*c^2*d*tan(f*x)^4*tan(e)^3 - 36*A*c*d^2*tan(f*x)^4*tan(e)^3 + 36*C*
c*d^2*tan(f*x)^4*tan(e)^3 + 12*B*d^3*tan(f*x)^4*tan(e)^3 - 12*C*c^3*tan(f*x
)^3*tan(e)^4 - 36*B*c^2*d*tan(f*x)^3*tan(e)^4 - 36*A*c*d^2*tan(f*x)^3*tan(e
)^4 + 36*C*c*d^2*tan(f*x)^3*tan(e)^4 + 12*B*d^3*tan(f*x)^3*tan(e)^4 + 72*A*
c^3*f*x*tan(f*x)^2*tan(e)^2 - 72*C*c^3*f*x*tan(f*x)^2*tan(e)^2 - 216*B*c^2*
d*f*x*tan(f*x)^2*tan(e)^2 - 216*A*c*d^2*f*x*tan(f*x)^2*tan(e)^2 + 216*C*c*d
^2*f*x*tan(f*x)^2*tan(e)^2 + 72*B*d^3*f*x*tan(f*x)^2*tan(e)^2 + 18*C*c^2*d*
tan(f*x)^4*tan(e)^2 + 18*B*c*d^2*tan(f*x)^4*tan(e)^2 + 6*A*d^3*tan(f*x)^4*t
an(e)^2 - 6*C*d^3*tan(f*x)^4*tan(e)^2 - 36*C*c^2*d*tan(f*x)^3*tan(e)^3 - 36
*B*c*d^2*tan(f*x)^3*tan(e)^3 - 12*A*d^3*tan(f*x)^3*tan(e)^3 + 24*C*d^3*tan(
f*x)^3*tan(e)^3 + 18*C*c^2*d*tan(f*x)^2*tan(e)^4 + 18*B*c*d^2*tan(f*x)^2*ta
n(e)^4 + 6*A*d^3*tan(f*x)^2*tan(e)^4 - 6*C*d^3*tan(f*x)^2*tan(e)^4 - 12*C*c
*d^2*tan(f*x)^4*tan(e) - 4*B*d^3*tan(f*x)^4*tan(e) - 36*B*c^3*log(4*(tan(f*
x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*
tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 - 108*A*c^2*d*log(
4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*
x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2 + 108*C*c
^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2
 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*tan(e)^2
+ 108*B*c*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2
*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)^2*
tan(e)^2 + 36*A*d^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(
f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f
*x)^2*tan(e)^2 - 36*C*d^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e)
+ tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))
*tan(f*x)^2*tan(e)^2 + 36*C*c^3*tan(f*x)^3*tan(e)^2 + 108*B*c^2*d*tan(f*x)^
3*tan(e)^2 + 108*A*c*d^2*tan(f*x)^3*tan(e)^2 - 144*C*c*d^2*tan(f*x)^3*tan(e
)^2 - 48*B*d^3*tan(f*x)^3*tan(e)^2 + 36*C*c^3*tan(f*x)^2*tan(e)^3 + 108*B*c
^2*d*tan(f*x)^2*tan(e)^3 + 108*A*c*d^2*tan(f*x)^2*tan(e)^3 - 144*C*c*d^2*ta
n(f*x)^2*tan(e)^3 - 48*B*d^3*tan(f*x)^2*tan(e)^3 - 12*C*c*d^2*tan(f*x)*tan(
e)^4 - 4*B*d^3*tan(f*x)*tan(e)^4 + 3*C*d^3*tan(f*x)^4 - 48*A*c^3*f*x*tan(f*
x)*tan(e) + 48*C*c^3*f*x*tan(f*x)*tan(e) + 144*B*c^2*d*f*x*tan(f*x)*tan(e)
```

```
+ 144*A*c*d^2*f*x*tan(f*x)*tan(e) - 144*C*c*d^2*f*x*tan(f*x)*tan(e) - 48*B*
d^3*f*x*tan(f*x)*tan(e) - 36*C*c^2*d*tan(f*x)^3*tan(e) - 36*B*c*d^2*tan(f*x
)^3*tan(e) - 12*A*d^3*tan(f*x)^3*tan(e) + 24*C*d^3*tan(f*x)^3*tan(e) + 36*C
*c^2*d*tan(f*x)^2*tan(e)^2 + 36*B*c*d^2*tan(f*x)^2*tan(e)^2 + 12*A*d^3*tan(
f*x)^2*tan(e)^2 - 12*C*d^3*tan(f*x)^2*tan(e)^2 - 36*C*c^2*d*tan(f*x)*tan(e)
^3 - 36*B*c*d^2*tan(f*x)*tan(e)^3 - 12*A*d^3*tan(f*x)*tan(e)^3 + 24*C*d^3*t
an(f*x)*tan(e)^3 + 3*C*d^3*tan(e)^4 + 12*C*c*d^2*tan(f*x)^3 + 4*B*d^3*tan(f
*x)^3 + 24*B*c^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x
)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(f*x)
*tan(e) + 72*A*c^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan
(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*tan(
f*x)*tan(e) - 72*C*c^2*d*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) +
 tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1))*
tan(f*x)*tan(e) - 72*B*c*d^2*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(
e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 +
1))*tan(f*x)*tan(e) - 24*A*d^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*ta
n(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2
+ 1))*tan(f*x)*tan(e) + 24*C*d^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*
tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^
2 + 1))*tan(f*x)*tan(e) - 36*C*c^3*tan(f*x)^2*tan(e) - 108*B*c^2*d*tan(f*x)
^2*tan(e) - 108*A*c*d^2*tan(f*x)^2*tan(e) + 144*C*c*d^2*tan(f*x)^2*tan(e) +
 48*B*d^3*tan(f*x)^2*tan(e) - 36*C*c^3*tan(f*x)*tan(e)^2 - 108*B*c^2*d*tan(
f*x)*tan(e)^2 - 108*A*c*d^2*tan(f*x)*tan(e)^2 + 144*C*c*d^2*tan(f*x)*tan(e)
^2 + 48*B*d^3*tan(f*x)*tan(e)^2 + 12*C*c*d^2*tan(e)^3 + 4*B*d^3*tan(e)^3 +
12*A*c^3*f*x - 12*C*c^3*f*x - 36*B*c^2*d*f*x - 36*A*c*d^2*f*x + 36*C*c*d^2*
f*x + 12*B*d^3*f*x + 18*C*c^2*d*tan(f*x)^2 + 18*B*c*d^2*tan(f*x)^2 + 6*A*d^
3*tan(f*x)^2 - 6*C*d^3*tan(f*x)^2 - 36*C*c^2*d*tan(f*x)*tan(e) - 36*B*c*d^2
*tan(f*x)*tan(e) - 12*A*d^3*tan(f*x)*tan(e) + 24*C*d^3*tan(f*x)*tan(e) + 18
*C*c^2*d*tan(e)^2 + 18*B*c*d^2*tan(e)^2 + 6*A*d^3*tan(e)^2 - 6*C*d^3*tan(e)
^2 - 6*B*c^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*
tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) - 18*A*c^2*d
*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + t
an(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 18*C*c^2*d*log(4*(tan(
f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 -
2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 18*B*c*d^2*log(4*(tan(f*x)^4*tan(e
)^2 - 2*tan(f*x)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*t
an(e) + 1)/(tan(e)^2 + 1)) + 6*A*d^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x
)^3*tan(e) + tan(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan
(e)^2 + 1)) - 6*C*d^3*log(4*(tan(f*x)^4*tan(e)^2 - 2*tan(f*x)^3*tan(e) + ta
n(f*x)^2*tan(e)^2 + tan(f*x)^2 - 2*tan(f*x)*tan(e) + 1)/(tan(e)^2 + 1)) + 1
2*C*c^3*tan(f*x) + 36*B*c^2*d*tan(f*x) + 36*A*c*d^2*tan(f*x) - 36*C*c*d^2*t
an(f*x) - 12*B*d^3*tan(f*x) + 12*C*c^3*tan(e) + 36*B*c^2*d*tan(e) + 36*A*c*
d^2*tan(e) - 36*C*c*d^2*tan(e) - 12*B*d^3*tan(e) + 18*C*c^2*d + 18*B*c*d^2
+ 6*A*d^3 - 9*C*d^3)/(f*tan(f*x)^4*tan(e)^4 - 4*f*tan(f*x)^3*tan(e)^3 + 6*f
*tan(f*x)^2*tan(e)^2 - 4*f*tan(f*x)*tan(e) + f)
```

maple [B] time = 0.03, size = 420, normalized size = 2.20

$$\frac{C\,d^3\left(\tan^4\left(fx+e\right)\right)}{4f}+\frac{B\left(\tan^3\left(fx+e\right)\right)d^3}{3f}+\frac{C\left(\tan^3\left(fx+e\right)\right)c\,d^2}{f}+\frac{A\left(\tan^2\left(fx+e\right)\right)d^3}{2f}+\frac{3B\left(\tan^2\left(fx+e\right)\right)c}{2f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] 1/4/f*C*d^3*tan(f*x+e)^4+1/3/f*B*tan(f*x+e)^3*d^3+1/f*C*tan(f*x+e)^3*c*d^2+
1/2/f*A*tan(f*x+e)^2*d^3+3/2/f*B*tan(f*x+e)^2*c*d^2+3/2/f*C*tan(f*x+e)^2*c^
2*d-1/2/f*C*tan(f*x+e)^2*d^3+3/f*A*c*d^2*tan(f*x+e)+3/f*B*c^2*d*tan(f*x+e)-
1/f*B*d^3*tan(f*x+e)+1/f*c^3*C*tan(f*x+e)-3/f*c*C*d^2*tan(f*x+e)+3/2/f*ln(1
+tan(f*x+e)^2)*A*c^2*d-1/2/f*ln(1+tan(f*x+e)^2)*A*d^3+1/2/f*ln(1+tan(f*x+e)
^2)*B*c^3-3/2/f*ln(1+tan(f*x+e)^2)*B*c*d^2-3/2/f*ln(1+tan(f*x+e)^2)*C*c^2*d
+1/2/f*ln(1+tan(f*x+e)^2)*C*d^3+1/f*A*arctan(tan(f*x+e))*c^3-3/f*A*arctan(t
an(f*x+e))*c*d^2-3/f*B*arctan(tan(f*x+e))*c^2*d+1/f*B*arctan(tan(f*x+e))*d^
3-1/f*C*arctan(tan(f*x+e))*c^3+3/f*C*arctan(tan(f*x+e))*c*d^2
```

maxima [A] time = 0.58, size = 202, normalized size = 1.06

$$3\,Cd^3\tan\left(fx+e\right)^4+4\left(3\,Ccd^2+Bd^3\right)\tan\left(fx+e\right)^3+6\left(3\,Cc^2d+3\,Bcd^2+(A-C)d^3\right)\tan\left(fx+e\right)^2+12\left((A\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
maxima")
```

```
[Out] 1/12*(3*C*d^3*tan(f*x + e)^4 + 4*(3*C*c*d^2 + B*d^3)*tan(f*x + e)^3 + 6*(3*
C*c^2*d + 3*B*c*d^2 + (A - C)*d^3)*tan(f*x + e)^2 + 12*((A - C)*c^3 - 3*B*c
^2*d - 3*(A - C)*c*d^2 + B*d^3)*(f*x + e) + 6*(B*c^3 + 3*(A - C)*c^2*d - 3*
B*c*d^2 - (A - C)*d^3)*log(tan(f*x + e)^2 + 1) + 12*(C*c^3 + 3*B*c^2*d + 3*
(A - C)*c*d^2 - B*d^3)*tan(f*x + e))/f
```

mupad [B] time = 8.79, size = 221, normalized size = 1.16

$$x\,\left(A\,c^3+B\,d^3-C\,c^3-3\,A\,c\,d^2-3\,B\,c^2\,d+3\,C\,c\,d^2\right)+\frac{\tan\left(e+f\,x\right)\,\left(C\,c^3-B\,d^3+3\,A\,c\,d^2+3\,B\,c^2\,d-3\,C\,c\,d^2\right)}{f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] x*(A*c^3 + B*d^3 - C*c^3 - 3*A*c*d^2 - 3*B*c^2*d + 3*C*c*d^2) + (tan(e + f*
x)*(C*c^3 - B*d^3 + 3*A*c*d^2 + 3*B*c^2*d - 3*C*c*d^2))/f + (tan(e + f*x)^3
*((B*d^3)/3 + C*c*d^2))/f - (log(tan(e + f*x)^2 + 1)*((A*d^3)/2 - (B*c^3)/2
 - (C*d^3)/2 - (3*A*c^2*d)/2 + (3*B*c*d^2)/2 + (3*C*c^2*d)/2))/f + (tan(e +
 f*x)^2*((A*d^3)/2 - (C*d^3)/2 + (3*B*c*d^2)/2 + (3*C*c^2*d)/2))/f + (C*d^3
*tan(e + f*x)^4)/(4*f)
```

sympy [A] time = 0.76, size = 410, normalized size = 2.15

$$\begin{cases} Ac^3x + \frac{3Ac^2d\log\left(\tan^2\left(e+fx\right)+1\right)}{2f} - 3Acd^2x + \frac{3Acd^2\tan\left(e+fx\right)}{f} - \frac{Ad^3\log\left(\tan^2\left(e+fx\right)+1\right)}{2f} + \frac{Ad^3\tan^2\left(e+fx\right)}{2f} + \frac{Bc^3\log\left(\tan^2\left(e+fx\right.\right.}{2f} \\ x\left(c+d\tan\left(e\right)\right)^3\left(A+B\tan\left(e\right)+C\tan^2\left(e\right)\right) \end{cases}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Piecewise((A*c**3*x + 3*A*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) - 3*A*c*d**
2*x + 3*A*c*d**2*tan(e + f*x)/f - A*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + A
*d**3*tan(e + f*x)**2/(2*f) + B*c**3*log(tan(e + f*x)**2 + 1)/(2*f) - 3*B*c
**2*d*x + 3*B*c**2*d*tan(e + f*x)/f - 3*B*c*d**2*log(tan(e + f*x)**2 + 1)/(
2*f) + 3*B*c*d**2*tan(e + f*x)**2/(2*f) + B*d**3*x + B*d**3*tan(e + f*x)**3
/(3*f) - B*d**3*tan(e + f*x)/f - C*c**3*x + C*c**3*tan(e + f*x)/f - 3*C*c**
2*d*log(tan(e + f*x)**2 + 1)/(2*f) + 3*C*c**2*d*tan(e + f*x)**2/(2*f) + 3*C
*c*d**2*x + C*c*d**2*tan(e + f*x)**3/f - 3*C*c*d**2*tan(e + f*x)/f + C*d**3
*log(tan(e + f*x)**2 + 1)/(2*f) + C*d**3*tan(e + f*x)**4/(4*f) - C*d**3*tan
(e + f*x)**2/(2*f), Ne(f, 0)), (x*(c + d*tan(e))**3*(A + B*tan(e) + C*tan(e
)**2), True))
```

3.67 $\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$

Optimal. Leaf size=363

$$-\frac{\log(\cos(e+fx))\left(A\left(ad\left(3c^2-d^2\right)-b\left(c^3-3cd^2\right)\right)+a\left(Bc^3-3Bcd^2-3c^2Cd+Cd^3\right)+b\left(3Bc^2d-Bd^3+c^3C\right.\right.}{f\left(a^2+b^2\right)}$$

```
[Out] -(a*(c^3*C+3*B*c^2*d-3*c*C*d^2-B*d^3-A*(c^3-3*c*d^2))-b*((A-C)*d*(3*c^2-d^2
)+B*(c^3-3*c*d^2)))*x/(a^2+b^2)-(b*(3*B*c^2*d-B*d^3+C*c^3-3*C*c*d^2)+a*(B*c
^3-3*B*c*d^2-3*C*c^2*d+C*d^3)+A*(a*d*(3*c^2-d^2)-b*(c^3-3*c*d^2)))*ln(cos(f
*x+e))/(a^2+b^2)/f+(A*b^2-a*(B*b-C*a))*(-a*d+b*c)^3*ln(a+b*tan(f*x+e))/b^4/
(a^2+b^2)/f+d*(b^2*d*(B*c+(A-C)*d)+(-a*d+b*c)*(B*b*d-C*a*d+C*b*c))*tan(f*x+
e)/b^3/f+1/2*(B*b*d-C*a*d+C*b*c)*(c+d*tan(f*x+e))^2/b^2/f+1/3*C*(c+d*tan(f*
x+e))^3/b/f
```

Rubi [A] time = 1.51, antiderivative size = 363, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3647, 3637, 3626, 3617, 31, 3475}

$$-\frac{\log(\cos(e+fx))\left(A\left(ad\left(3c^2-d^2\right)-b\left(c^3-3cd^2\right)\right)+a\left(Bc^3-3Bcd^2-3c^2Cd+Cd^3\right)+b\left(3Bc^2d-Bd^3+c^3C\right.\right.}{f\left(a^2+b^2\right)}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x]),x]
```

```
[Out] -(((a*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3 - A*(c^3 - 3*c*d^2)) - b*((A -
 C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)))*x)/(a^2 + b^2)) - ((b*(c^3*C + 3*
B*c^2*d - 3*c*C*d^2 - B*d^3) + a*(B*c^3 - 3*c^2*C*d - 3*B*c*d^2 + C*d^3) +
A*(a*d*(3*c^2 - d^2) - b*(c^3 - 3*c*d^2)))*Log[Cos[e + f*x]])/((a^2 + b^2)*
f) + ((A*b^2 - a*(b*B - a*C))*(b*c - a*d)^3*Log[a + b*Tan[e + f*x]])/(b^4*(
a^2 + b^2)*f) + (d*(b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(b*c*C + b*B*d -
a*C*d))*Tan[e + f*x])/(b^3*f) + ((b*c*C + b*B*d - a*C*d)*(c + d*Tan[e + f*x
])^2)/(2*b^2*f) + (C*(c + d*Tan[e + f*x])^3)/(3*b*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \frac{C(c+d\tan(e+fx))^3}{3bf} + \frac{\int \frac{(c+d\tan(e+fx))^2\left(3(A\right.}{}}{}$$

$$= \frac{(bcC+bBd-aCd)(c+d\tan(e+fx))^2}{2b^2f} + \frac{C}{}$$

$$= \frac{d\left(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+bB\right.}{b^3f}$$

$$= -\frac{\left(a\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-\right.\right.\right.}{}$$

$$= -\frac{\left(a\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-\right.\right.\right.}{}$$

$$= -\frac{\left(a\left(c^3C+3Bc^2d-3cCd^2-Bd^3-A\left(c^3-\right.\right.\right.}{}$$

Mathematica [C] time = 4.83, size = 255, normalized size = 0.70

$$\frac{\frac{6(bc-ad)^3\left(a(aC-bB)+Ab^2\right)\log(a+b\tan(e+fx))}{b^2\left(a^2+b^2\right)} + \frac{3b^2(c+id)^3(-iA+B+iC)\log(-\tan(e+fx)+i)}{a+ib} - \frac{3b^2(d+ic)^3(A-iB-C)\log(\tan(e+fx)+i)}{a-ib} + 3(-a}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x]),x]
```

```
[Out] ((3*b^2*((-I)*A + B + I*C)*(c + I*d)^3*Log[I - Tan[e + f*x]])/(a + I*b) - (
3*b^2*(A - I*B - C)*(I*c + d)^3*Log[I + Tan[e + f*x]])/(a - I*b) + (6*(A*b^
2 + a*(-(b*B) + a*C))*(b*c - a*d)^3*Log[a + b*Tan[e + f*x]])/(b^2*(a^2 + b^
2)) + 6*b*d^2*(B*c + (A - C)*d)*Tan[e + f*x] + (6*d*(b*c - a*d)*(b*c*C + b*
B*d - a*C*d)*Tan[e + f*x])/b + 3*(b*c*C + b*B*d - a*C*d)*(c + d*Tan[e + f*x
])^2 + 2*b*C*(c + d*Tan[e + f*x])^3)/(6*b^2*f)
```

fricas [A] time = 3.14, size = 623, normalized size = 1.72

$$\frac{2\left(Ca^2b^3+Cb^5\right)d^3\tan\left(fx+e\right)^3+6\left(\left(\left((A-C)ab^4+Bb^5\right)c^3-3\left(Bab^4-(A-C)b^5\right)c^2d-3\left((A-C)ab^4+Bb^5\right)c\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] 1/6*(2*(C*a^2*b^3 + C*b^5)*d^3*tan(f*x + e)^3 + 6*(((A - C)*a*b^4 + B*b^5)*
c^3 - 3*(B*a*b^4 - (A - C)*b^5)*c^2*d - 3*((A - C)*a*b^4 + B*b^5)*c*d^2 + (
B*a*b^4 - (A - C)*b^5)*d^3)*f*x + 3*(3*(C*a^2*b^3 + C*b^5)*c*d^2 - (C*a^3*b
^2 - B*a^2*b^3 + C*a*b^4 - B*b^5)*d^3)*tan(f*x + e)^2 + 3*((C*a^2*b^3 - B*a
*b^4 + A*b^5)*c^3 - 3*(C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c^2*d + 3*(C*a^4*b
- B*a^3*b^2 + A*a^2*b^3)*c*d^2 - (C*a^5 - B*a^4*b + A*a^3*b^2)*d^3)*log((b^
2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - 3*((C*
a^2*b^3 + C*b^5)*c^3 - 3*(C*a^3*b^2 - B*a^2*b^3 + C*a*b^4 - B*b^5)*c^2*d +
3*(C*a^4*b - B*a^3*b^2 + A*a^2*b^3 - B*a*b^4 + (A - C)*b^5)*c*d^2 - (C*a^5
- B*a^4*b + A*a^3*b^2 + (A - C)*a*b^4 + B*b^5)*d^3)*log(1/(tan(f*x + e)^2 +
 1)) + 6*(3*(C*a^2*b^3 + C*b^5)*c^2*d - 3*(C*a^3*b^2 - B*a^2*b^3 + C*a*b^4
- B*b^5)*c*d^2 + (C*a^4*b - B*a^3*b^2 + A*a^2*b^3 - B*a*b^4 + (A - C)*b^5)*
d^3)*tan(f*x + e))/((a^2*b^4 + b^6)*f)
```

giac [A] time = 3.41, size = 573, normalized size = 1.58

$$\frac{\frac{6\left(Aac^3-Cac^3+Bbc^3-3\,Bac^2d+3\,Abc^2d-3\,Cbc^2d-3\,Aacd^2+3\,Cacd^2-3\,Bbcd^2+Bad^3-Abd^3+Cbd^3\right)(fx+e)}{a^2+b^2}+\frac{3\left(Bac^3-Abc^3+Cbc^3+3\,Aac^2d-3\,Cac^2d+3\right.}{}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/6*(6*(A*a*c^3 - C*a*c^3 + B*b*c^3 - 3*B*a*c^2*d + 3*A*b*c^2*d - 3*C*b*c^2
*d - 3*A*a*c*d^2 + 3*C*a*c*d^2 - 3*B*b*c*d^2 + B*a*d^3 - A*b*d^3 + C*b*d^3)
*(f*x + e)/(a^2 + b^2) + 3*(B*a*c^3 - A*b*c^3 + C*b*c^3 + 3*A*a*c^2*d - 3*C
*a*c^2*d + 3*B*b*c^2*d - 3*B*a*c*d^2 + 3*A*b*c*d^2 - 3*C*b*c*d^2 - A*a*d^3
+ C*a*d^3 - B*b*d^3)*log(tan(f*x + e)^2 + 1)/(a^2 + b^2) + 6*(C*a^2*b^3*c^3
 - B*a*b^4*c^3 + A*b^5*c^3 - 3*C*a^3*b^2*c^2*d + 3*B*a^2*b^3*c^2*d - 3*A*a*
b^4*c^2*d + 3*C*a^4*b*c*d^2 - 3*B*a^3*b^2*c*d^2 + 3*A*a^2*b^3*c*d^2 - C*a^5
*d^3 + B*a^4*b*d^3 - A*a^3*b^2*d^3)*log(abs(b*tan(f*x + e) + a))/(a^2*b^4 +
 b^6) + (2*C*b^2*d^3*tan(f*x + e)^3 + 9*C*b^2*c*d^2*tan(f*x + e)^2 - 3*C*a*
b*d^3*tan(f*x + e)^2 + 3*B*b^2*d^3*tan(f*x + e)^2 + 18*C*b^2*c^2*d*tan(f*x
+ e) - 18*C*a*b*c*d^2*tan(f*x + e) + 18*B*b^2*c*d^2*tan(f*x + e) + 6*C*a^2*
d^3*tan(f*x + e) - 6*B*a*b*d^3*tan(f*x + e) + 6*A*b^2*d^3*tan(f*x + e) - 6*
C*b^2*d^3*tan(f*x + e))/b^3)/f
```

maple [B] time = 0.25, size = 1304, normalized size = 3.59

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

[In] int((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e)),x)

[Out] -1/f*d^3/b^2*B*a*tan(f*x+e)+3/f*d^2/b*B*c*tan(f*x+e)+1/f/(a^2+b^2)*B*arctan(tan(f*x+e))*a*d^3+1/f*d^3/b^3*a^2*C*tan(f*x+e)+1/f/b^3/(a^2+b^2)*ln(a+b*tan(f*x+e))*B*a^4*d^3+3/f/b/(a^2+b^2)*ln(a+b*tan(f*x+e))*a^2*B*c^2*d+3/f/b^3/(a^2+b^2)*ln(a+b*tan(f*x+e))*C*a^4*c*d^2+3/f/b/(a^2+b^2)*ln(a+b*tan(f*x+e))*A*a^2*c*d^2-3/f/b^2/(a^2+b^2)*ln(a+b*tan(f*x+e))*C*a^3*c^2*d-3/f/b^2/(a^2+b^2)*ln(a+b*tan(f*x+e))*a^3*B*c*d^2-1/f/b^2/(a^2+b^2)*ln(a+b*tan(f*x+e))*A*a^3*d^3+3/f/(a^2+b^2)*A*arctan(tan(f*x+e))*b*c^2*d+3/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*A*b*c*d^2-3/f/(a^2+b^2)*B*arctan(tan(f*x+e))*b*c*d^2-3/f/(a^2+b^2)*C*arctan(tan(f*x+e))*b*c^2*d-3/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*C*a*c^2*d-3/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*B*a*c*d^2+3/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*B*b*c^2*d-3/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*C*b*c*d^2+1/f/b/(a^2+b^2)*ln(a+b*tan(f*x+e))*C*a^2*c^3-1/f/b^4/(a^2+b^2)*ln(a+b*tan(f*x+e))*C*a^5*d^3-3/f/(a^2+b^2)*A*arctan(tan(f*x+e))*a*c*d^2-3/f*d^2/b^2*C*a*c*tan(f*x+e)+3/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*A*a*c^2*d-3/f/(a^2+b^2)*ln(a+b*tan(f*x+e))*A*a*c^2*d-3/f/(a^2+b^2)*B*arctan(tan(f*x+e))*a*c^2*d+3/f/(a^2+b^2)*C*arctan(tan(f*x+e))*a*c*d^2+3/f*d/b*C*c^2*tan(f*x+e)+1/f*b/(a^2+b^2)*ln(a+b*tan(f*x+e))*A*c^3+1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*B*a*c^3-1/f/(a^2+b^2)*A*arctan(tan(f*x+e))*b*d^3+1/f/(a^2+b^2)*B*arctan(tan(f*x+e))*b*c^3-1/f/(a^2+b^2)*C*arctan(tan(f*x+e))*a*c^3+1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*C*a*d^3-1/f/(a^2+b^2)*ln(a+b*tan(f*x+e))*B*a*c^3+1/f/(a^2+b^2)*A*arctan(tan(f*x+e))*a*c^3-1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*A*a*d^3-1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*A*b*c^3-1/2/f*d^3/b^2*C*tan(f*x+e)^2*a+3/2/f*d^2/b*C*tan(f*x+e)^2*c+1/f/(a^2+b^2)*C*arctan(tan(f*x+e))*b*d^3+1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*C*b*c^3-1/2/f/(a^2+b^2)*ln(1+tan(f*x+e)^2)*B*b*d^3+1/3/f*d^3/b*C*tan(f*x+e)^3+1/2/f*d^3/b*B*tan(f*x+e)^2+1/f*d^3/b*A*tan(f*x+e)-1/f*d^3/b*C*tan(f*x+e)

maxima [A] time = 0.54, size = 436, normalized size = 1.20

$$\frac{\frac{6\left(((A-C)a+Bb)c^3-3\,(Ba-(A-C)b)c^2d-3\,((A-C)a+Bb)cd^2+(Ba-(A-C)b)d^3\right)\left(fx+e\right)}{a^2+b^2}+\frac{6\left(\left(Ca^2b^3-Bab^4+Ab^5\right)c^3-3\left(Ca^3b^2-Ba^2b^3+Aab^4\right)c^2d+3\left(C\right.\right.}{}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e)),x, algorithm="maxima")

[Out] 1/6*(6*(((A - C)*a + B*b)*c^3 - 3*(B*a - (A - C)*b)*c^2*d - 3*((A - C)*a + B*b)*c*d^2 + (B*a - (A - C)*b)*d^3)*(f*x + e)/(a^2 + b^2) + 6*((C*a^2*b^3 - B*a*b^4 + A*b^5)*c^3 - 3*(C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c^2*d + 3*(C*a^4*b - B*a^3*b^2 + A*a^2*b^3)*c*d^2 - (C*a^5 - B*a^4*b + A*a^3*b^2)*d^3)*log

```
(b*tan(f*x + e) + a)/(a^2*b^4 + b^6) + 3*((B*a - (A - C)*b)*c^3 + 3*((A - C
)*a + B*b)*c^2*d - 3*(B*a - (A - C)*b)*c*d^2 - ((A - C)*a + B*b)*d^3)*log(t
an(f*x + e)^2 + 1)/(a^2 + b^2) + (2*C*b^2*d^3*tan(f*x + e)^3 + 3*(3*C*b^2*c
*d^2 - (C*a*b - B*b^2)*d^3)*tan(f*x + e)^2 + 6*(3*C*b^2*c^2*d - 3*(C*a*b -
B*b^2)*c*d^2 + (C*a^2 - B*a*b + (A - C)*b^2)*d^3)*tan(f*x + e))/b^3)/f
```

mupad [B] time = 13.00, size = 508, normalized size = 1.40

$$\frac{\tan\left(e+f\,x\right)^2\,\left(\frac{B\,d^3+3\,C\,c\,d^2}{2\,b}-\frac{C\,a\,d^3}{2\,b^2}\right)}{f}-\frac{\tan\left(e+f\,x\right)\,\left(\frac{a\,\left(\frac{B\,d^3+3\,C\,c\,d^2}{b}-\frac{C\,a\,d^3}{b^2}\right)}{b}-\frac{3\,C\,c^2\,d+3\,B\,c\,d^2+A\,d^3}{b}+\frac{C\,d^3}{b}\right)}{f}-\ln\left(a+b\tan\left(\right.\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x)),x)
```

```
[Out] (tan(e + f*x)^2*((B*d^3 + 3*C*c*d^2)/(2*b) - (C*a*d^3)/(2*b^2)))/f - (tan(e
 + f*x)*((a*((B*d^3 + 3*C*c*d^2)/b - (C*a*d^3)/b^2))/b - (A*d^3 + 3*B*c*d^2
 + 3*C*c^2*d)/b + (C*d^3)/b))/f - (log(a + b*tan(e + f*x))*(b^4*(B*a*c^3 +
3*A*a*c^2*d) - b^3*(C*a^2*c^3 + 3*A*a^2*c*d^2 + 3*B*a^2*c^2*d) + b^2*(A*a^3
*d^3 + 3*B*a^3*c*d^2 + 3*C*a^3*c^2*d) - b*(B*a^4*d^3 + 3*C*a^4*c*d^2) - A*b
^5*c^3 + C*a^5*d^3))/(f*(b^6 + a^2*b^4)) - (log(tan(e + f*x) + 1i)*(A*c^3 +
 A*d^3*1i - B*c^3*1i + B*d^3 - C*c^3 - C*d^3*1i - 3*A*c*d^2 - A*c^2*d*3i +
B*c*d^2*3i - 3*B*c^2*d + 3*C*c*d^2 + C*c^2*d*3i))/(2*f*(a*1i + b)) - (log(t
an(e + f*x) - 1i)*(A*c^3*1i + A*d^3 - B*c^3 + B*d^3*1i - C*c^3*1i - C*d^3 -
 A*c*d^2*3i - 3*A*c^2*d + 3*B*c*d^2 - B*c^2*d*3i + C*c*d^2*3i + 3*C*c^2*d))
/(2*f*(a + b*1i)) + (C*d^3*tan(e + f*x)^3)/(3*b*f)
```

sympy [A] time = 113.33, size = 7205, normalized size = 19.85

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e)),x)
```

```
[Out] Piecewise((zoo*x*(c + d*tan(e))**3*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(
a, 0) & Eq(b, 0) & Eq(f, 0)), (-3*I*A*c**3*f*x*tan(e + f*x)/(-6*b*f*tan(e +
 f*x) + 6*I*b*f) - 3*A*c**3*f*x/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 3*I*A*c**
3/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 9*A*c**2*d*f*x*tan(e + f*x)/(-6*b*f*tan
(e + f*x) + 6*I*b*f) + 9*I*A*c**2*d*f*x/(-6*b*f*tan(e + f*x) + 6*I*b*f) + 9
*A*c**2*d/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 9*I*A*c*d**2*f*x*tan(e + f*x)/(
-6*b*f*tan(e + f*x) + 6*I*b*f) - 9*A*c*d**2*f*x/(-6*b*f*tan(e + f*x) + 6*I*
```

```
b*f) - 9*A*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x
) + 6*I*b*f) + 9*I*A*c*d**2*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) +
 6*I*b*f) + 9*I*A*c*d**2/(-6*b*f*tan(e + f*x) + 6*I*b*f) + 9*A*d**3*f*x*tan
(e + f*x)/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 9*I*A*d**3*f*x/(-6*b*f*tan(e +
f*x) + 6*I*b*f) - 3*I*A*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*
tan(e + f*x) + 6*I*b*f) - 3*A*d**3*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e +
 f*x) + 6*I*b*f) - 6*A*d**3*tan(e + f*x)**2/(-6*b*f*tan(e + f*x) + 6*I*b*f)
 - 9*A*d**3/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 3*B*c**3*f*x*tan(e + f*x)/(-6
*b*f*tan(e + f*x) + 6*I*b*f) + 3*I*B*c**3*f*x/(-6*b*f*tan(e + f*x) + 6*I*b*
f) + 3*B*c**3/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 9*I*B*c**2*d*f*x*tan(e + f*
x)/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 9*B*c**2*d*f*x/(-6*b*f*tan(e + f*x) +
6*I*b*f) - 9*B*c**2*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e +
 f*x) + 6*I*b*f) + 9*I*B*c**2*d*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*
x) + 6*I*b*f) + 9*I*B*c**2*d/(-6*b*f*tan(e + f*x) + 6*I*b*f) + 27*B*c*d**2*
f*x*tan(e + f*x)/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 27*I*B*c*d**2*f*x/(-6*b*
f*tan(e + f*x) + 6*I*b*f) - 9*I*B*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f
*x)/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 9*B*c*d**2*log(tan(e + f*x)**2 + 1)/(
-6*b*f*tan(e + f*x) + 6*I*b*f) - 18*B*c*d**2*tan(e + f*x)**2/(-6*b*f*tan(e
+ f*x) + 6*I*b*f) - 27*B*c*d**2/(-6*b*f*tan(e + f*x) + 6*I*b*f) + 9*I*B*d**
3*f*x*tan(e + f*x)/(-6*b*f*tan(e + f*x) + 6*I*b*f) + 9*B*d**3*f*x/(-6*b*f*t
an(e + f*x) + 6*I*b*f) + 6*B*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6
*b*f*tan(e + f*x) + 6*I*b*f) - 6*I*B*d**3*log(tan(e + f*x)**2 + 1)/(-6*b*f*
tan(e + f*x) + 6*I*b*f) - 3*B*d**3*tan(e + f*x)**3/(-6*b*f*tan(e + f*x) + 6
*I*b*f) - 3*I*B*d**3*tan(e + f*x)**2/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 9*I*
B*d**3/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 3*I*C*c**3*f*x*tan(e + f*x)/(-6*b*
f*tan(e + f*x) + 6*I*b*f) - 3*C*c**3*f*x/(-6*b*f*tan(e + f*x) + 6*I*b*f) -
3*C*c**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x) + 6*I*b
*f) + 3*I*C*c**3*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) + 6*I*b*f) +
 3*I*C*c**3/(-6*b*f*tan(e + f*x) + 6*I*b*f) + 27*C*c**2*d*f*x*tan(e + f*x)/
(-6*b*f*tan(e + f*x) + 6*I*b*f) - 27*I*C*c**2*d*f*x/(-6*b*f*tan(e + f*x) +
6*I*b*f) - 9*I*C*c**2*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e
 + f*x) + 6*I*b*f) - 9*C*c**2*d*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*
x) + 6*I*b*f) - 18*C*c**2*d*tan(e + f*x)**2/(-6*b*f*tan(e + f*x) + 6*I*b*f)
 - 27*C*c**2*d/(-6*b*f*tan(e + f*x) + 6*I*b*f) + 27*I*C*c*d**2*f*x*tan(e +
f*x)/(-6*b*f*tan(e + f*x) + 6*I*b*f) + 27*C*c*d**2*f*x/(-6*b*f*tan(e + f*x)
 + 6*I*b*f) + 18*C*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan
(e + f*x) + 6*I*b*f) - 18*I*C*c*d**2*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e
 + f*x) + 6*I*b*f) - 9*C*c*d**2*tan(e + f*x)**3/(-6*b*f*tan(e + f*x) + 6*I*
b*f) - 9*I*C*c*d**2*tan(e + f*x)**2/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 27*I*
C*c*d**2/(-6*b*f*tan(e + f*x) + 6*I*b*f) - 15*C*d**3*f*x*tan(e + f*x)/(-6*b
*f*tan(e + f*x) + 6*I*b*f) + 15*I*C*d**3*f*x/(-6*b*f*tan(e + f*x) + 6*I*b*f
) + 6*I*C*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x) +
 6*I*b*f) + 6*C*d**3*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) + 6*I*b*
f) - 2*C*d**3*tan(e + f*x)**4/(-6*b*f*tan(e + f*x) + 6*I*b*f) - I*C*d**3*ta
n(e + f*x)**3/(-6*b*f*tan(e + f*x) + 6*I*b*f) + 9*C*d**3*tan(e + f*x)**2/(-
```

```
6*b*f*tan(e + f*x) + 6*I*b*f) + 15*C*d**3/(-6*b*f*tan(e + f*x) + 6*I*b*f),
Eq(a, -I*b)), (3*I*A*c**3*f*x*tan(e + f*x)/(-6*b*f*tan(e + f*x) - 6*I*b*f)
- 3*A*c**3*f*x/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 3*I*A*c**3/(-6*b*f*tan(e +
 f*x) - 6*I*b*f) - 9*A*c**2*d*f*x*tan(e + f*x)/(-6*b*f*tan(e + f*x) - 6*I*b
*f) - 9*I*A*c**2*d*f*x/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 9*A*c**2*d/(-6*b*f
*tan(e + f*x) - 6*I*b*f) + 9*I*A*c*d**2*f*x*tan(e + f*x)/(-6*b*f*tan(e + f*
x) - 6*I*b*f) - 9*A*c*d**2*f*x/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 9*A*c*d**2
*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 9*
I*A*c*d**2*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 9*I*A
*c*d**2/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 9*A*d**3*f*x*tan(e + f*x)/(-6*b*f
*tan(e + f*x) - 6*I*b*f) + 9*I*A*d**3*f*x/(-6*b*f*tan(e + f*x) - 6*I*b*f) +
 3*I*A*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x) - 6*
I*b*f) - 3*A*d**3*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) - 6*I*b*f)
- 6*A*d**3*tan(e + f*x)**2/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 9*A*d**3/(-6*b
*f*tan(e + f*x) - 6*I*b*f) - 3*B*c**3*f*x*tan(e + f*x)/(-6*b*f*tan(e + f*x)
 - 6*I*b*f) - 3*I*B*c**3*f*x/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 3*B*c**3/(-6
*b*f*tan(e + f*x) - 6*I*b*f) + 9*I*B*c**2*d*f*x*tan(e + f*x)/(-6*b*f*tan(e
+ f*x) - 6*I*b*f) - 9*B*c**2*d*f*x/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 9*B*c*
*2*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x) - 6*I*b*f)
- 9*I*B*c**2*d*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 9
*I*B*c**2*d/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 27*B*c*d**2*f*x*tan(e + f*x)/
(-6*b*f*tan(e + f*x) - 6*I*b*f) + 27*I*B*c*d**2*f*x/(-6*b*f*tan(e + f*x) -
6*I*b*f) + 9*I*B*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e
 + f*x) - 6*I*b*f) - 9*B*c*d**2*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*
x) - 6*I*b*f) - 18*B*c*d**2*tan(e + f*x)**2/(-6*b*f*tan(e + f*x) - 6*I*b*f)
 - 27*B*c*d**2/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 9*I*B*d**3*f*x*tan(e + f*x
)/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 9*B*d**3*f*x/(-6*b*f*tan(e + f*x) - 6*I
*b*f) + 6*B*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x)
 - 6*I*b*f) + 6*I*B*d**3*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) - 6*
I*b*f) - 3*B*d**3*tan(e + f*x)**3/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 3*I*B*d
**3*tan(e + f*x)**2/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 9*I*B*d**3/(-6*b*f*ta
n(e + f*x) - 6*I*b*f) + 3*I*C*c**3*f*x*tan(e + f*x)/(-6*b*f*tan(e + f*x) -
6*I*b*f) - 3*C*c**3*f*x/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 3*C*c**3*log(tan(
e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 3*I*C*c**3*
log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 3*I*C*c**3/(-6*b
*f*tan(e + f*x) - 6*I*b*f) + 27*C*c**2*d*f*x*tan(e + f*x)/(-6*b*f*tan(e + f
*x) - 6*I*b*f) + 27*I*C*c**2*d*f*x/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 9*I*C*
c**2*d*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x) - 6*I*b*f
) - 9*C*c**2*d*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 1
8*C*c**2*d*tan(e + f*x)**2/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 27*C*c**2*d/(-
6*b*f*tan(e + f*x) - 6*I*b*f) - 27*I*C*c*d**2*f*x*tan(e + f*x)/(-6*b*f*tan(
e + f*x) - 6*I*b*f) + 27*C*c*d**2*f*x/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 18*
C*c*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x) - 6*I*b
*f) + 18*I*C*c*d**2*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) - 6*I*b*f
) - 9*C*c*d**2*tan(e + f*x)**3/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 9*I*C*c*d*
```

```
*2*tan(e + f*x)**2/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 27*I*C*c*d**2/(-6*b*f*
tan(e + f*x) - 6*I*b*f) - 15*C*d**3*f*x*tan(e + f*x)/(-6*b*f*tan(e + f*x) -
 6*I*b*f) - 15*I*C*d**3*f*x/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 6*I*C*d**3*lo
g(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*b*f*tan(e + f*x) - 6*I*b*f) + 6*C*d
**3*log(tan(e + f*x)**2 + 1)/(-6*b*f*tan(e + f*x) - 6*I*b*f) - 2*C*d**3*tan
(e + f*x)**4/(-6*b*f*tan(e + f*x) - 6*I*b*f) + I*C*d**3*tan(e + f*x)**3/(-6
*b*f*tan(e + f*x) - 6*I*b*f) + 9*C*d**3*tan(e + f*x)**2/(-6*b*f*tan(e + f*x
) - 6*I*b*f) + 15*C*d**3/(-6*b*f*tan(e + f*x) - 6*I*b*f), Eq(a, I*b)), ((A*
c**3*x + 3*A*c**2*d*log(tan(e + f*x)**2 + 1)/(2*f) - 3*A*c*d**2*x + 3*A*c*d
**2*tan(e + f*x)/f - A*d**3*log(tan(e + f*x)**2 + 1)/(2*f) + A*d**3*tan(e +
 f*x)**2/(2*f) + B*c**3*log(tan(e + f*x)**2 + 1)/(2*f) - 3*B*c**2*d*x + 3*B
*c**2*d*tan(e + f*x)/f - 3*B*c*d**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*c*
d**2*tan(e + f*x)**2/(2*f) + B*d**3*x + B*d**3*tan(e + f*x)**3/(3*f) - B*d*
*3*tan(e + f*x)/f - C*c**3*x + C*c**3*tan(e + f*x)/f - 3*C*c**2*d*log(tan(e
 + f*x)**2 + 1)/(2*f) + 3*C*c**2*d*tan(e + f*x)**2/(2*f) + 3*C*c*d**2*x + C
*c*d**2*tan(e + f*x)**3/f - 3*C*c*d**2*tan(e + f*x)/f + C*d**3*log(tan(e +
f*x)**2 + 1)/(2*f) + C*d**3*tan(e + f*x)**4/(4*f) - C*d**3*tan(e + f*x)**2/
(2*f))/a, Eq(b, 0)), (x*(c + d*tan(e))**3*(A + B*tan(e) + C*tan(e)**2)/(a +
 b*tan(e)), Eq(f, 0)), (-6*A*a**3*b**2*d**3*log(a/b + tan(e + f*x))/(6*a**2
*b**4*f + 6*b**6*f) + 18*A*a**2*b**3*c*d**2*log(a/b + tan(e + f*x))/(6*a**2
*b**4*f + 6*b**6*f) + 6*A*a**2*b**3*d**3*tan(e + f*x)/(6*a**2*b**4*f + 6*b*
*6*f) + 6*A*a*b**4*c**3*f*x/(6*a**2*b**4*f + 6*b**6*f) - 18*A*a*b**4*c**2*d
*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) + 9*A*a*b**4*c**2*d*log
(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) - 18*A*a*b**4*c*d**2*f*x/(
6*a**2*b**4*f + 6*b**6*f) - 3*A*a*b**4*d**3*log(tan(e + f*x)**2 + 1)/(6*a**
2*b**4*f + 6*b**6*f) + 6*A*b**5*c**3*log(a/b + tan(e + f*x))/(6*a**2*b**4*f
 + 6*b**6*f) - 3*A*b**5*c**3*log(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b*
*6*f) + 18*A*b**5*c**2*d*f*x/(6*a**2*b**4*f + 6*b**6*f) + 9*A*b**5*c*d**2*l
og(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) - 6*A*b**5*d**3*f*x/(6*a
**2*b**4*f + 6*b**6*f) + 6*A*b**5*d**3*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6
*f) + 6*B*a**4*b*d**3*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) -
18*B*a**3*b**2*c*d**2*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) -
6*B*a**3*b**2*d**3*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6*f) + 18*B*a**2*b**3
*c**2*d*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) + 18*B*a**2*b**3
*c*d**2*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6*f) + 3*B*a**2*b**3*d**3*tan(e
+ f*x)**2/(6*a**2*b**4*f + 6*b**6*f) - 6*B*a*b**4*c**3*log(a/b + tan(e + f*
x))/(6*a**2*b**4*f + 6*b**6*f) + 3*B*a*b**4*c**3*log(tan(e + f*x)**2 + 1)/(
6*a**2*b**4*f + 6*b**6*f) - 18*B*a*b**4*c**2*d*f*x/(6*a**2*b**4*f + 6*b**6*
f) - 9*B*a*b**4*c*d**2*log(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f)
+ 6*B*a*b**4*d**3*f*x/(6*a**2*b**4*f + 6*b**6*f) - 6*B*a*b**4*d**3*tan(e +
f*x)/(6*a**2*b**4*f + 6*b**6*f) + 6*B*b**5*c**3*f*x/(6*a**2*b**4*f + 6*b**6
*f) + 9*B*b**5*c**2*d*log(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) -
 18*B*b**5*c*d**2*f*x/(6*a**2*b**4*f + 6*b**6*f) + 18*B*b**5*c*d**2*tan(e +
 f*x)/(6*a**2*b**4*f + 6*b**6*f) - 3*B*b**5*d**3*log(tan(e + f*x)**2 + 1)/(
6*a**2*b**4*f + 6*b**6*f) + 3*B*b**5*d**3*tan(e + f*x)**2/(6*a**2*b**4*f +
```

```
6*b**6*f) - 6*C*a**5*d**3*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f
) + 18*C*a**4*b*c*d**2*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) +
 6*C*a**4*b*d**3*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6*f) - 18*C*a**3*b**2*c
**2*d*log(a/b + tan(e + f*x))/(6*a**2*b**4*f + 6*b**6*f) - 18*C*a**3*b**2*c
*d**2*tan(e + f*x)/(6*a**2*b**4*f + 6*b**6*f) - 3*C*a**3*b**2*d**3*tan(e +
f*x)**2/(6*a**2*b**4*f + 6*b**6*f) + 6*C*a**2*b**3*c**3*log(a/b + tan(e + f
*x))/(6*a**2*b**4*f + 6*b**6*f) + 18*C*a**2*b**3*c**2*d*tan(e + f*x)/(6*a**
2*b**4*f + 6*b**6*f) + 9*C*a**2*b**3*c*d**2*tan(e + f*x)**2/(6*a**2*b**4*f
+ 6*b**6*f) + 2*C*a**2*b**3*d**3*tan(e + f*x)**3/(6*a**2*b**4*f + 6*b**6*f)
 - 6*C*a*b**4*c**3*f*x/(6*a**2*b**4*f + 6*b**6*f) - 9*C*a*b**4*c**2*d*log(t
an(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) + 18*C*a*b**4*c*d**2*f*x/(6*
a**2*b**4*f + 6*b**6*f) - 18*C*a*b**4*c*d**2*tan(e + f*x)/(6*a**2*b**4*f +
6*b**6*f) + 3*C*a*b**4*d**3*log(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**
6*f) - 3*C*a*b**4*d**3*tan(e + f*x)**2/(6*a**2*b**4*f + 6*b**6*f) + 3*C*b**
5*c**3*log(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f + 6*b**6*f) - 18*C*b**5*c**2
*d*f*x/(6*a**2*b**4*f + 6*b**6*f) + 18*C*b**5*c**2*d*tan(e + f*x)/(6*a**2*b
**4*f + 6*b**6*f) - 9*C*b**5*c*d**2*log(tan(e + f*x)**2 + 1)/(6*a**2*b**4*f
 + 6*b**6*f) + 9*C*b**5*c*d**2*tan(e + f*x)**2/(6*a**2*b**4*f + 6*b**6*f) +
 6*C*b**5*d**3*f*x/(6*a**2*b**4*f + 6*b**6*f) + 2*C*b**5*d**3*tan(e + f*x)*
*3/(6*a**2*b**4*f + 6*b**6*f) - 6*C*b**5*d**3*tan(e + f*x)/(6*a**2*b**4*f +
 6*b**6*f), True))
```

3.68 $\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$

Optimal. Leaf size=574

$$\frac{\log(\cos(e+fx))\left(-\left(a^2\left(d(A-C)\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)\right)+2ab\left(Ac^3-3Acd^2-3Bc^2d+Bd^3-c^3C+3cC\right.}{f\left(a^2+b^2\right)^2}$$

```
[Out] -(b^2*(A*c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)+a^2*(c^3*C+3*B*c^2*
d-3*c*C*d^2-B*d^3-A*(c^3-3*c*d^2))-2*a*b*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^
2)))*x/(a^2+b^2)^2+(2*a*b*(A*c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)
-a^2*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2))+b^2*((A-C)*d*(3*c^2-d^2)+B*(c^3-
3*c*d^2)))*ln(cos(f*x+e))/(a^2+b^2)^2/f-(-a*d+b*c)^2*(2*a^3*b*B*d-3*a^4*C*d
-b^4*(3*A*d+B*c)-2*a*b^3*(A*c-2*B*d-C*c)+a^2*b^2*(B*c-(A+5*C)*d))*ln(a+b*ta
n(f*x+e))/b^4/(a^2+b^2)^2/f-d^2*(3*a^3*C*d-A*b^2*(-a*d+b*c)-b^3*(B*d+2*C*c)
-a^2*b*(2*B*d+3*C*c)+a*b^2*(B*c+2*C*d))*tan(f*x+e)/b^3/(a^2+b^2)/f+1/2*(2*A
*b^2-2*B*a*b+3*C*a^2+C*b^2)*d*(c+d*tan(f*x+e))^2/b^2/(a^2+b^2)/f-(A*b^2-a*(
B*b-C*a))*(c+d*tan(f*x+e))^3/b/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] time = 2.32, antiderivative size = 574, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.156, Rules used = {3645, 3647, 3637, 3626, 3617, 31, 3475}

$$\frac{\log(\cos(e+fx))\left(a^2\left(-\left(d(A-C)\left(3c^2-d^2\right)+B\left(c^3-3cd^2\right)\right)\right)+2ab\left(Ac^3-3Acd^2-3Bc^2d+Bd^3-c^3C+3cC\right.}{f\left(a^2+b^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x])^2,x]
```

```
[Out] -(((b^2*(A*c^3 - c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3) + a^2*(
c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3 - A*(c^3 - 3*c*d^2)) - 2*a*b*((A - C)
*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)))*x)/(a^2 + b^2)^2) + ((2*a*b*(A*c^3 -
 c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3) - a^2*((A - C)*d*(3*c^2
 - d^2) + B*(c^3 - 3*c*d^2)) + b^2*((A - C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*
d^2)))*Log[Cos[e + f*x]])/((a^2 + b^2)^2*f) - ((b*c - a*d)^2*(2*a^3*b*B*d -
 3*a^4*C*d - b^4*(B*c + 3*A*d) - 2*a*b^3*(A*c - c*C - 2*B*d) + a^2*b^2*(B*c
 - (A + 5*C)*d))*Log[a + b*Tan[e + f*x]])/(b^4*(a^2 + b^2)^2*f) - (d^2*(3*a
^3*C*d - A*b^2*(b*c - a*d) - b^3*(2*c*C + B*d) - a^2*b*(3*c*C + 2*B*d) + a*
b^2*(B*c + 2*C*d))*Tan[e + f*x])/(b^3*(a^2 + b^2)*f) + ((2*A*b^2 - 2*a*b*B
+ 3*a^2*C + b^2*C)*d*(c + d*Tan[e + f*x])^2)/(2*b^2*(a^2 + b^2)*f) - ((A*b^
2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^3)/(b*(a^2 + b^2)*f*(a + b*Tan[e +
f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
```

```
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^3}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))} + \cdots$$

$$= \frac{\left(2Ab^2-2abB+3a^2C+b^2C\right)d(c+d\tan(e \cdots}{2b^2\left(a^2+b^2\right)f} \cdots$$

$$= -\frac{d^2\left(3a^3Cd-Ab^2(bc-ad)-b^3(2cC+Bd) \cdots\right.}{b^3\left(\cdots\right.}$$

$$= -\frac{\left(b^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2 \cdots\right.\right.}{\cdots}$$

$$= -\frac{\left(b^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2 \cdots\right.\right.}{\cdots}$$

$$= -\frac{\left(b^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2 \cdots\right.\right.}{\cdots}$$

Mathematica [C] time = 8.58, size = 2467, normalized size = 4.30

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x])^2,x]
```

```
[Out] ((a^2*A*c^3 - A*b^2*c^3 + 2*a*b*B*c^3 - a^2*c^3*C + b^2*c^3*C + 6*a*A*b*c^2
*d - 3*a^2*B*c^2*d + 3*b^2*B*c^2*d - 6*a*b*c^2*C*d - 3*a^2*A*c*d^2 + 3*A*b^
2*c*d^2 - 6*a*b*B*c*d^2 + 3*a^2*c*C*d^2 - 3*b^2*c*C*d^2 - 2*a*A*b*d^3 + a^2
*B*d^3 - b^2*B*d^3 + 2*a*b*C*d^3)*(e + f*x)*Cos[e + f*x]*(a*Cos[e + f*x] +
b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^3)/((a - I*b)^2*(a + I*b)^2*f*(c*Cos
[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^2) - (I*(-2*a^6*A*b^8*c^
3 + (2*I)*a^5*A*b^9*c^3 - 2*a^4*A*b^10*c^3 + (2*I)*a^3*A*b^11*c^3 + a^7*b^7
*B*c^3 - I*a^6*b^8*B*c^3 - a^3*b^11*B*c^3 + I*a^2*b^12*B*c^3 + 2*a^6*b^8*c^
3*C - (2*I)*a^5*b^9*c^3*C + 2*a^4*b^10*c^3*C - (2*I)*a^3*b^11*c^3*C + 3*a^7
*A*b^7*c^2*d - (3*I)*a^6*A*b^8*c^2*d - 3*a^3*A*b^11*c^2*d + (3*I)*a^2*A*b^1
2*c^2*d + 6*a^6*b^8*B*c^2*d - (6*I)*a^5*b^9*B*c^2*d + 6*a^4*b^10*B*c^2*d -
(6*I)*a^3*b^11*B*c^2*d - 3*a^9*b^5*c^2*C*d + (3*I)*a^8*b^6*c^2*C*d - 12*a^7
*b^7*c^2*C*d + (12*I)*a^6*b^8*c^2*C*d - 9*a^5*b^9*c^2*C*d + (9*I)*a^4*b^10*
c^2*C*d + 6*a^6*A*b^8*c*d^2 - (6*I)*a^5*A*b^9*c*d^2 + 6*a^4*A*b^10*c*d^2 -
(6*I)*a^3*A*b^11*c*d^2 - 3*a^9*b^5*B*c*d^2 + (3*I)*a^8*b^6*B*c*d^2 - 12*a^7
*b^7*B*c*d^2 + (12*I)*a^6*b^8*B*c*d^2 - 9*a^5*b^9*B*c*d^2 + (9*I)*a^4*b^10*
B*c*d^2 + 6*a^10*b^4*c*C*d^2 - (6*I)*a^9*b^5*c*C*d^2 + 18*a^8*b^6*c*C*d^2 -
 (18*I)*a^7*b^7*c*C*d^2 + 12*a^6*b^8*c*C*d^2 - (12*I)*a^5*b^9*c*C*d^2 - a^9
*A*b^5*d^3 + I*a^8*A*b^6*d^3 - 4*a^7*A*b^7*d^3 + (4*I)*a^6*A*b^8*d^3 - 3*a^
5*A*b^9*d^3 + (3*I)*a^4*A*b^10*d^3 + 2*a^10*b^4*B*d^3 - (2*I)*a^9*b^5*B*d^3
 + 6*a^8*b^6*B*d^3 - (6*I)*a^7*b^7*B*d^3 + 4*a^6*b^8*B*d^3 - (4*I)*a^5*b^9*
B*d^3 - 3*a^11*b^3*C*d^3 + (3*I)*a^10*b^4*C*d^3 - 8*a^9*b^5*C*d^3 + (8*I)*a
^8*b^6*C*d^3 - 5*a^7*b^7*C*d^3 + (5*I)*a^6*b^8*C*d^3)*(e + f*x)*Cos[e + f*x
]*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^3)/(a^2*(a - I*b
)^4*(a + I*b)^3*b^7*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*
x])^2) - (I*(2*a*A*b^5*c^3 - a^2*b^4*B*c^3 + b^6*B*c^3 - 2*a*b^5*c^3*C - 3*
a^2*A*b^4*c^2*d + 3*A*b^6*c^2*d - 6*a*b^5*B*c^2*d + 3*a^4*b^2*c^2*C*d + 9*a
^2*b^4*c^2*C*d - 6*a*A*b^5*c*d^2 + 3*a^4*b^2*B*c*d^2 + 9*a^2*b^4*B*c*d^2 -
6*a^5*b*c*C*d^2 - 12*a^3*b^3*c*C*d^2 + a^4*A*b^2*d^3 + 3*a^2*A*b^4*d^3 - 2*
a^5*b*B*d^3 - 4*a^3*b^3*B*d^3 + 3*a^6*C*d^3 + 5*a^4*b^2*C*d^3)*ArcTan[Tan[e
 + f*x]]*Cos[e + f*x]*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*
x])^3)/(b^4*(a^2 + b^2)^2*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[
e + f*x])^2) + ((-3*b^2*c^2*C*d - 3*b^2*B*c*d^2 + 6*a*b*c*C*d^2 - A*b^2*d^3
 + 2*a*b*B*d^3 - 3*a^2*C*d^3 + b^2*C*d^3)*Cos[e + f*x]*Log[Cos[e + f*x]]*(a
*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^3)/(b^4*f*(c*Cos[e +
 f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^2) + ((2*a*A*b^5*c^3 - a^2*b
```

```
^4*B*c^3 + b^6*B*c^3 - 2*a*b^5*c^3*C - 3*a^2*A*b^4*c^2*d + 3*A*b^6*c^2*d -
6*a*b^5*B*c^2*d + 3*a^4*b^2*c^2*C*d + 9*a^2*b^4*c^2*C*d - 6*a*A*b^5*c*d^2 +
 3*a^4*b^2*B*c*d^2 + 9*a^2*b^4*B*c*d^2 - 6*a^5*b*c*C*d^2 - 12*a^3*b^3*c*C*d
^2 + a^4*A*b^2*d^3 + 3*a^2*A*b^4*d^3 - 2*a^5*b*B*d^3 - 4*a^3*b^3*B*d^3 + 3*
a^6*C*d^3 + 5*a^4*b^2*C*d^3)*Cos[e + f*x]*Log[(a*Cos[e + f*x] + b*Sin[e + f
*x])^2]*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^3)/(2*b^4*
(a^2 + b^2)^2*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^2)
 + (C*d^3*Sec[e + f*x]*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f
*x])^3)/(2*b^2*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^2
) + ((a*Cos[e + f*x] + b*Sin[e + f*x])^2*(3*b*c*C*d^2*Sin[e + f*x] + b*B*d^
3*Sin[e + f*x] - 2*a*C*d^3*Sin[e + f*x])*(c + d*Tan[e + f*x])^3)/(b^3*f*(c*
Cos[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^2) + (Cos[e + f*x]*(a
*Cos[e + f*x] + b*Sin[e + f*x])*(A*b^5*c^3*Sin[e + f*x] - a*b^4*B*c^3*Sin[e
 + f*x] + a^2*b^3*c^3*C*Sin[e + f*x] - 3*a*A*b^4*c^2*d*Sin[e + f*x] + 3*a^2
*b^3*B*c^2*d*Sin[e + f*x] - 3*a^3*b^2*c^2*C*d*Sin[e + f*x] + 3*a^2*A*b^3*c*
d^2*Sin[e + f*x] - 3*a^3*b^2*B*c*d^2*Sin[e + f*x] + 3*a^4*b*c*C*d^2*Sin[e +
 f*x] - a^3*A*b^2*d^3*Sin[e + f*x] + a^4*b*B*d^3*Sin[e + f*x] - a^5*C*d^3*S
in[e + f*x])*(c + d*Tan[e + f*x])^3)/(a*(a - I*b)*(a + I*b)*b^3*f*(c*Cos[e
+ f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^2)
```

fricas [B] time = 3.09, size = 1512, normalized size = 2.63

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] 1/2*((C*a^4*b^3 + 2*C*a^2*b^5 + C*b^7)*d^3*tan(f*x + e)^3 - 2*(C*a^2*b^5 -
B*a*b^6 + A*b^7)*c^3 + 6*(C*a^3*b^4 - B*a^2*b^5 + A*a*b^6)*c^2*d - 6*(C*a^4
*b^3 - B*a^3*b^4 + A*a^2*b^5)*c*d^2 + (3*C*a^5*b^2 - 2*B*a^4*b^3 + 2*(A + C
)*a^3*b^4 + C*a*b^6)*d^3 + 2*(((A - C)*a^3*b^4 + 2*B*a^2*b^5 - (A - C)*a*b^
6)*c^3 - 3*(B*a^3*b^4 - 2*(A - C)*a^2*b^5 - B*a*b^6)*c^2*d - 3*((A - C)*a^3
*b^4 + 2*B*a^2*b^5 - (A - C)*a*b^6)*c*d^2 + (B*a^3*b^4 - 2*(A - C)*a^2*b^5
- B*a*b^6)*d^3)*f*x + (6*(C*a^4*b^3 + 2*C*a^2*b^5 + C*b^7)*c*d^2 - (3*C*a^5
*b^2 - 2*B*a^4*b^3 + 6*C*a^3*b^4 - 4*B*a^2*b^5 + 3*C*a*b^6 - 2*B*b^7)*d^3)*
tan(f*x + e)^2 - ((B*a^3*b^4 - 2*(A - C)*a^2*b^5 - B*a*b^6)*c^3 - 3*(C*a^5*
b^2 - (A - 3*C)*a^3*b^4 - 2*B*a^2*b^5 + A*a*b^6)*c^2*d + 3*(2*C*a^6*b - B*a
^5*b^2 + 4*C*a^4*b^3 - 3*B*a^3*b^4 + 2*A*a^2*b^5)*c*d^2 - (3*C*a^7 - 2*B*a^
6*b + (A + 5*C)*a^5*b^2 - 4*B*a^4*b^3 + 3*A*a^3*b^4)*d^3 + ((B*a^2*b^5 - 2*
(A - C)*a*b^6 - B*b^7)*c^3 - 3*(C*a^4*b^3 - (A - 3*C)*a^2*b^5 - 2*B*a*b^6 +
 A*b^7)*c^2*d + 3*(2*C*a^5*b^2 - B*a^4*b^3 + 4*C*a^3*b^4 - 3*B*a^2*b^5 + 2*
A*a*b^6)*c*d^2 - (3*C*a^6*b - 2*B*a^5*b^2 + (A + 5*C)*a^4*b^3 - 4*B*a^3*b^4
 + 3*A*a^2*b^5)*d^3)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x
+ e) + a^2)/(tan(f*x + e)^2 + 1)) - (3*(C*a^5*b^2 + 2*C*a^3*b^4 + C*a*b^6)*
```

```
c^2*d - 3*(2*C*a^6*b - B*a^5*b^2 + 4*C*a^4*b^3 - 2*B*a^3*b^4 + 2*C*a^2*b^5
- B*a*b^6)*c*d^2 + (3*C*a^7 - 2*B*a^6*b + (A + 5*C)*a^5*b^2 - 4*B*a^4*b^3 +
 (2*A + C)*a^3*b^4 - 2*B*a^2*b^5 + (A - C)*a*b^6)*d^3 + (3*(C*a^4*b^3 + 2*C
*a^2*b^5 + C*b^7)*c^2*d - 3*(2*C*a^5*b^2 - B*a^4*b^3 + 4*C*a^3*b^4 - 2*B*a^
2*b^5 + 2*C*a*b^6 - B*b^7)*c*d^2 + (3*C*a^6*b - 2*B*a^5*b^2 + (A + 5*C)*a^4
*b^3 - 4*B*a^3*b^4 + (2*A + C)*a^2*b^5 - 2*B*a*b^6 + (A - C)*b^7)*d^3)*tan(
f*x + e))*log(1/(tan(f*x + e)^2 + 1)) + (2*(C*a^3*b^4 - B*a^2*b^5 + A*a*b^6
)*c^3 - 6*(C*a^4*b^3 - B*a^3*b^4 + A*a^2*b^5)*c^2*d + 6*(2*C*a^5*b^2 - B*a^
4*b^3 + (A + 2*C)*a^3*b^4 + C*a*b^6)*c*d^2 - (6*C*a^6*b - 4*B*a^5*b^2 + (2*
A + 7*C)*a^4*b^3 - 4*B*a^3*b^4 + 2*C*a^2*b^5 - 2*B*a*b^6 - C*b^7)*d^3 + 2*(
((A - C)*a^2*b^5 + 2*B*a*b^6 - (A - C)*b^7)*c^3 - 3*(B*a^2*b^5 - 2*(A - C)*
a*b^6 - B*b^7)*c^2*d - 3*((A - C)*a^2*b^5 + 2*B*a*b^6 - (A - C)*b^7)*c*d^2
+ (B*a^2*b^5 - 2*(A - C)*a*b^6 - B*b^7)*d^3)*f*x)*tan(f*x + e))/((a^4*b^5 +
 2*a^2*b^7 + b^9)*f*tan(f*x + e) + (a^5*b^4 + 2*a^3*b^6 + a*b^8)*f)
```

giac [B] time = 7.28, size = 1357, normalized size = 2.36

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c^3 - C*a^2*c^3 + 2*B*a*b*c^3 - A*b^2*c^3 + C*b^2*c^3 - 3*B*a
^2*c^2*d + 6*A*a*b*c^2*d - 6*C*a*b*c^2*d + 3*B*b^2*c^2*d - 3*A*a^2*c*d^2 +
3*C*a^2*c*d^2 - 6*B*a*b*c*d^2 + 3*A*b^2*c*d^2 - 3*C*b^2*c*d^2 + B*a^2*d^3 -
 2*A*a*b*d^3 + 2*C*a*b*d^3 - B*b^2*d^3)*(f*x + e)/(a^4 + 2*a^2*b^2 + b^4) +
 (B*a^2*c^3 - 2*A*a*b*c^3 + 2*C*a*b*c^3 - B*b^2*c^3 + 3*A*a^2*c^2*d - 3*C*a
^2*c^2*d + 6*B*a*b*c^2*d - 3*A*b^2*c^2*d + 3*C*b^2*c^2*d - 3*B*a^2*c*d^2 +
6*A*a*b*c*d^2 - 6*C*a*b*c*d^2 + 3*B*b^2*c*d^2 - A*a^2*d^3 + C*a^2*d^3 - 2*B
*a*b*d^3 + A*b^2*d^3 - C*b^2*d^3)*log(tan(f*x + e)^2 + 1)/(a^4 + 2*a^2*b^2
+ b^4) - 2*(B*a^2*b^4*c^3 - 2*A*a*b^5*c^3 + 2*C*a*b^5*c^3 - B*b^6*c^3 - 3*C
*a^4*b^2*c^2*d + 3*A*a^2*b^4*c^2*d - 9*C*a^2*b^4*c^2*d + 6*B*a*b^5*c^2*d -
3*A*b^6*c^2*d + 6*C*a^5*b*c*d^2 - 3*B*a^4*b^2*c*d^2 + 12*C*a^3*b^3*c*d^2 -
9*B*a^2*b^4*c*d^2 + 6*A*a*b^5*c*d^2 - 3*C*a^6*d^3 + 2*B*a^5*b*d^3 - A*a^4*b
^2*d^3 - 5*C*a^4*b^2*d^3 + 4*B*a^3*b^3*d^3 - 3*A*a^2*b^4*d^3)*log(abs(b*tan
(f*x + e) + a))/(a^4*b^4 + 2*a^2*b^6 + b^8) + 2*(B*a^2*b^5*c^3*tan(f*x + e)
 - 2*A*a*b^6*c^3*tan(f*x + e) + 2*C*a*b^6*c^3*tan(f*x + e) - B*b^7*c^3*tan(
f*x + e) - 3*C*a^4*b^3*c^2*d*tan(f*x + e) + 3*A*a^2*b^5*c^2*d*tan(f*x + e)
- 9*C*a^2*b^5*c^2*d*tan(f*x + e) + 6*B*a*b^6*c^2*d*tan(f*x + e) - 3*A*b^7*c
^2*d*tan(f*x + e) + 6*C*a^5*b^2*c*d^2*tan(f*x + e) - 3*B*a^4*b^3*c*d^2*tan(
f*x + e) + 12*C*a^3*b^4*c*d^2*tan(f*x + e) - 9*B*a^2*b^5*c*d^2*tan(f*x + e)
 + 6*A*a*b^6*c*d^2*tan(f*x + e) - 3*C*a^6*b*d^3*tan(f*x + e) + 2*B*a^5*b^2*
d^3*tan(f*x + e) - A*a^4*b^3*d^3*tan(f*x + e) - 5*C*a^4*b^3*d^3*tan(f*x + e
) + 4*B*a^3*b^4*d^3*tan(f*x + e) - 3*A*a^2*b^5*d^3*tan(f*x + e) - C*a^4*b^3
```

```
*c^3 + 2*B*a^3*b^4*c^3 - 3*A*a^2*b^5*c^3 + C*a^2*b^5*c^3 - A*b^7*c^3 - 3*B*
a^4*b^3*c^2*d + 6*A*a^3*b^4*c^2*d - 6*C*a^3*b^4*c^2*d + 3*B*a^2*b^5*c^2*d +
 3*C*a^6*b*c*d^2 - 3*A*a^4*b^3*c*d^2 + 9*C*a^4*b^3*c*d^2 - 6*B*a^3*b^4*c*d^
2 + 3*A*a^2*b^5*c*d^2 - 2*C*a^7*d^3 + B*a^6*b*d^3 - 4*C*a^5*b^2*d^3 + 3*B*a
^4*b^3*d^3 - 2*A*a^3*b^4*d^3)/((a^4*b^4 + 2*a^2*b^6 + b^8)*(b*tan(f*x + e)
+ a)) + (C*b^2*d^3*tan(f*x + e)^2 + 6*C*b^2*c*d^2*tan(f*x + e) - 4*C*a*b*d^
3*tan(f*x + e) + 2*B*b^2*d^3*tan(f*x + e))/b^4)/f
```

maple [B] time = 0.30, size = 2250, normalized size = 3.92

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2,x
)
```

```
[Out] 1/f/(a^2+b^2)/(a+b*tan(f*x+e))*B*a*c^3-1/f*b/(a^2+b^2)/(a+b*tan(f*x+e))*A*c
^3+1/f*b^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*c^3+1/f/(a^2+b^2)^2*A*arctan(ta
n(f*x+e))*a^2*c^3-1/f/(a^2+b^2)^2*A*arctan(tan(f*x+e))*b^2*c^3-1/2/f/(a^2+b
^2)^2*ln(1+tan(f*x+e)^2)*C*b^2*d^3-2/f*d^3/b^3*C*tan(f*x+e)*a+3/f*d^2/b^2*C
*c*tan(f*x+e)+3/f/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*a^2*d^3-1/f/(a^2+b^2)^2*
ln(a+b*tan(f*x+e))*B*a^2*c^3+1/f/(a^2+b^2)^2*B*arctan(tan(f*x+e))*a^2*d^3+5
/f/b^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*C*a^4*d^3-2/f*b/(a^2+b^2)^2*ln(a+b*ta
n(f*x+e))*C*a*c^3-1/f/b^3/(a^2+b^2)/(a+b*tan(f*x+e))*B*a^4*d^3-1/f/b/(a^2+b
^2)/(a+b*tan(f*x+e))*C*a^2*c^3+1/f/b^4/(a^2+b^2)/(a+b*tan(f*x+e))*C*a^5*d^3
+9/f/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*C*a^2*c^2*d+1/f/b^2/(a^2+b^2)^2*ln(a+b*
tan(f*x+e))*A*a^4*d^3+2/f*b/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*a*c^3+3/f*b^2/
(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*c^2*d-2/f/b^3/(a^2+b^2)^2*ln(a+b*tan(f*x+e
))*B*a^5*d^3-4/f/b/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*a^3*d^3+3/f/b^4/(a^2+b^
2)^2*ln(a+b*tan(f*x+e))*C*a^6*d^3+9/f/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*a^2*
c*d^2-3/f/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*a^2*c^2*d+3/f/(a^2+b^2)/(a+b*tan
(f*x+e))*A*a*c^2*d-3/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*C*a^2*c^2*d+1/f/(a^
2+b^2)^2*ln(1+tan(f*x+e)^2)*C*a*b*c^3+3/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*
C*b^2*c^2*d-3/f/(a^2+b^2)^2*A*arctan(tan(f*x+e))*a^2*c*d^2-2/f/(a^2+b^2)^2*
A*arctan(tan(f*x+e))*a*b*d^3+3/f/(a^2+b^2)^2*A*arctan(tan(f*x+e))*b^2*c*d^2
-3/f/(a^2+b^2)^2*B*arctan(tan(f*x+e))*a^2*c^2*d-12/f/b/(a^2+b^2)^2*ln(a+b*t
an(f*x+e))*C*a^3*c*d^2-6/f/(a^2+b^2)^2*C*arctan(tan(f*x+e))*a*b*c^2*d-3/f/b
/(a^2+b^2)/(a+b*tan(f*x+e))*a^2*B*c^2*d+2/f/(a^2+b^2)^2*C*arctan(tan(f*x+e)
)*a*b*d^3+3/f/b^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*a^4*c*d^2+3/f/(a^2+b^2)^
2*ln(1+tan(f*x+e)^2)*B*a*b*c^2*d-3/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*C*a*b*c
*d^2+3/f/b^2/(a^2+b^2)/(a+b*tan(f*x+e))*a^3*B*c*d^2-3/f/b/(a^2+b^2)/(a+b*ta
n(f*x+e))*A*a^2*c*d^2-6/f/(a^2+b^2)^2*B*arctan(tan(f*x+e))*a*b*c*d^2+6/f/(a
^2+b^2)^2*A*arctan(tan(f*x+e))*a*b*c^2*d-6/f*b/(a^2+b^2)^2*ln(a+b*tan(f*x+e
))*B*a*c^2*d+3/f/b^2/(a^2+b^2)/(a+b*tan(f*x+e))*C*a^3*c^2*d-6/f*b/(a^2+b^2)
^2*ln(a+b*tan(f*x+e))*A*a*c*d^2+3/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*A*a*b*c*
```

```
d^2-3/f/b^3/(a^2+b^2)/(a+b*tan(f*x+e))*C*a^4*c*d^2-6/f/b^3/(a^2+b^2)^2*ln(a
+b*tan(f*x+e))*C*a^5*c*d^2+3/f/b^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*C*a^4*c^2
*d+1/f*d^3/b^2*B*tan(f*x+e)+3/f/(a^2+b^2)^2*B*arctan(tan(f*x+e))*b^2*c^2*d+
3/f/(a^2+b^2)^2*C*arctan(tan(f*x+e))*a^2*c*d^2+2/f/(a^2+b^2)^2*B*arctan(tan
(f*x+e))*a*b*c^3-3/f/(a^2+b^2)^2*C*arctan(tan(f*x+e))*b^2*c*d^2+1/f/b^2/(a^
2+b^2)/(a+b*tan(f*x+e))*A*a^3*d^3+3/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*A*a^
2*c^2*d-1/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*A*a*b*c^3-3/2/f/(a^2+b^2)^2*ln(1
+tan(f*x+e)^2)*A*b^2*c^2*d-3/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*B*a^2*c*d^2
-1/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*B*a*b*d^3+3/2/f/(a^2+b^2)^2*ln(1+tan(f*
x+e)^2)*B*b^2*c*d^2+1/f/(a^2+b^2)^2*C*arctan(tan(f*x+e))*b^2*c^3+1/2/f*d^3/
b^2*C*tan(f*x+e)^2+1/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*C*a^2*d^3-1/f/(a^2+
b^2)^2*B*arctan(tan(f*x+e))*b^2*d^3-1/f/(a^2+b^2)^2*C*arctan(tan(f*x+e))*a^
2*c^3-1/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*A*a^2*d^3+1/2/f/(a^2+b^2)^2*ln(1
+tan(f*x+e)^2)*A*b^2*d^3+1/2/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*B*a^2*c^3-1/2
/f/(a^2+b^2)^2*ln(1+tan(f*x+e)^2)*B*b^2*c^3
```

maxima [A] time = 0.59, size = 685, normalized size = 1.19

$$\frac{2\left(\left(\left((A-C)a^2+2\,Bab-(A-C)b^2\right)c^3-3\left(Ba^2-2\,(A-C)ab-Bb^2\right)c^2d-3\left((A-C)a^2+2\,Bab-(A-C)b^2\right)cd^2+\left(Ba^2-2\,(A-C)ab-Bb^2\right)d^3\right)(fx+e)}{a^4+2\,a^2b^2+b^4}-\frac{2\left(\left(Ba^2b^4-2\,(A\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c^3 - 3*(B*a^2 - 2*(A - C)*a*
b - B*b^2)*c^2*d - 3*((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c*d^2 + (B*a^2 -
 2*(A - C)*a*b - B*b^2)*d^3)*(f*x + e)/(a^4 + 2*a^2*b^2 + b^4) - 2*((B*a^2*
b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^3 - 3*(C*a^4*b^2 - (A - 3*C)*a^2*b^4 - 2*B
*a*b^5 + A*b^6)*c^2*d + 3*(2*C*a^5*b - B*a^4*b^2 + 4*C*a^3*b^3 - 3*B*a^2*b^
4 + 2*A*a*b^5)*c*d^2 - (3*C*a^6 - 2*B*a^5*b + (A + 5*C)*a^4*b^2 - 4*B*a^3*b
^3 + 3*A*a^2*b^4)*d^3)*log(b*tan(f*x + e) + a)/(a^4*b^4 + 2*a^2*b^6 + b^8)
+ ((B*a^2 - 2*(A - C)*a*b - B*b^2)*c^3 + 3*((A - C)*a^2 + 2*B*a*b - (A - C)
*b^2)*c^2*d - 3*(B*a^2 - 2*(A - C)*a*b - B*b^2)*c*d^2 - ((A - C)*a^2 + 2*B*
a*b - (A - C)*b^2)*d^3)*log(tan(f*x + e)^2 + 1)/(a^4 + 2*a^2*b^2 + b^4) - 2
*((C*a^2*b^3 - B*a*b^4 + A*b^5)*c^3 - 3*(C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c
^2*d + 3*(C*a^4*b - B*a^3*b^2 + A*a^2*b^3)*c*d^2 - (C*a^5 - B*a^4*b + A*a^3
*b^2)*d^3)/(a^3*b^4 + a*b^6 + (a^2*b^5 + b^7)*tan(f*x + e)) + (C*b*d^3*tan(
f*x + e)^2 + 2*(3*C*b*c*d^2 - (2*C*a - B*b)*d^3)*tan(f*x + e))/b^3)/f
```

mupad [B] time = 15.70, size = 701, normalized size = 1.22

$$\frac{\tan\left(e+f\,x\right)\left(\frac{B\,d^3+3\,C\,c\,d^2}{b^2}-\frac{2\,C\,a\,d^3}{b^3}\right)}{f}-\frac{\ln\left(\tan\left(e+f\,x\right)+1\mathrm{i}\right)\left(B\,c^3-A\,d^3+C\,d^3+3\,A\,c^2\,d-3\,B\,c\,d^2-3\,C\,c^2\,d+\right.}{2\,f\,\left(-a^2+a\,b\,2\mathrm{i}+b\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x))^2,x)
```

```
[Out] (tan(e + f*x)*((B*d^3 + 3*C*c*d^2)/b^2 - (2*C*a*d^3)/b^3))/f - (log(tan(e +
 f*x) + 1i)*(A*c^3*1i - A*d^3 + B*c^3 + B*d^3*1i - C*c^3*1i + C*d^3 - A*c*d
^2*3i + 3*A*c^2*d - 3*B*c*d^2 - B*c^2*d*3i + C*c*d^2*3i - 3*C*c^2*d))/(2*f*
(a*b*2i - a^2 + b^2)) + (log(a + b*tan(e + f*x))*(b^4*(3*A*a^2*d^3 - B*a^2*
c^3 - 3*A*a^2*c^2*d + 9*B*a^2*c*d^2 + 9*C*a^2*c^2*d) - b^5*(2*C*a*c^3 - 2*A
*a*c^3 + 6*A*a*c*d^2 + 6*B*a*c^2*d) - b^3*(4*B*a^3*d^3 + 12*C*a^3*c*d^2) +
b^6*(B*c^3 + 3*A*c^2*d) - b*(2*B*a^5*d^3 + 6*C*a^5*c*d^2) + b^2*(A*a^4*d^3
+ 5*C*a^4*d^3 + 3*B*a^4*c*d^2 + 3*C*a^4*c^2*d) + 3*C*a^6*d^3))/(f*(b^8 + 2*
a^2*b^6 + a^4*b^4)) - (log(tan(e + f*x) - 1i)*(A*c^3 - A*d^3*1i + B*c^3*1i
+ B*d^3 - C*c^3 + C*d^3*1i - 3*A*c*d^2 + A*c^2*d*3i - B*c*d^2*3i - 3*B*c^2*
d + 3*C*c*d^2 - C*c^2*d*3i))/(2*f*(2*a*b - a^2*1i + b^2*1i)) - (A*b^5*c^3 -
 C*a^5*d^3 - B*a*b^4*c^3 + B*a^4*b*d^3 - A*a^3*b^2*d^3 + C*a^2*b^3*c^3 + 3*
A*a^2*b^3*c*d^2 + 3*B*a^2*b^3*c^2*d - 3*B*a^3*b^2*c*d^2 - 3*C*a^3*b^2*c^2*d
 - 3*A*a*b^4*c^2*d + 3*C*a^4*b*c*d^2)/(b*f*(a*b^3 + b^4*tan(e + f*x))*(a^2
+ b^2)) + (C*d^3*tan(e + f*x)^2)/(2*b^2*f)
```

sympy [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e))**2,x)
```

```
[Out] Timed out
```

3.69 $\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$

Optimal. Leaf size=798

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^3}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}+\frac{\left(-3Cda^4+bBda^3+b^2(2Bc+(A-7C)d)a^2-b^3(4Ac-4Cc-5Bd)a\right.}{2b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx}$$

```
[Out] -(3*a*b^2*(A*c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)+a^3*(c^3*C+3*B*
c^2*d-3*c*C*d^2-B*d^3-A*(c^3-3*c*d^2))-3*a^2*b*((A-C)*d*(3*c^2-d^2)+B*(c^3-
3*c*d^2))+b^3*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2)))*x/(a^2+b^2)^3-(b^3*(A*
c^3-3*A*c*d^2-3*B*c^2*d+B*d^3-C*c^3+3*C*c*d^2)+3*a^2*b*(c^3*C+3*B*c^2*d-3*c
*C*d^2-B*d^3-A*(c^3-3*c*d^2))+a^3*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2))-3*a
*b^2*((A-C)*d*(3*c^2-d^2)+B*(c^3-3*c*d^2)))*ln(cos(f*x+e))/(a^2+b^2)^3/f-(-
a*d+b*c)*(a^5*b*B*d^2-3*a^6*C*d^2+a^4*b^2*d*(B*c-9*C*d)+a^3*b^3*B*(c^2+3*d^
2)-b^6*(c*(3*B*d+C*c)-A*(c^2-3*d^2))-a*b^5*(8*c*(A-C)*d+3*B*(c^2-2*d^2))+a^
2*b^4*(3*c^2*C+6*B*c*d-10*C*d^2-A*(3*c^2-d^2)))*ln(a+b*tan(f*x+e))/b^4/(a^2
+b^2)^3/f-d^2*(a^3*b*B*d-3*a^4*C*d-a*b^3*(2*A*c-3*B*d-2*C*c)+a^2*b^2*(B*c-6
*C*d)-b^4*(B*c+(2*A+C)*d))*tan(f*x+e)/b^3/(a^2+b^2)^2/f+1/2*(a^3*b*B*d-3*a^
4*C*d-b^4*(3*A*d+2*B*c)-a*b^3*(4*A*c-5*B*d-4*C*c)+a^2*b^2*(2*B*c+(A-7*C)*d)
)*(c+d*tan(f*x+e))^2/b^2/(a^2+b^2)^2/f/(a+b*tan(f*x+e))-1/2*(A*b^2-a*(B*b-C
*a))*(c+d*tan(f*x+e))^3/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^2
```

Rubi [A] time = 2.84, antiderivative size = 798, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3645, 3637, 3626, 3617, 31, 3475}

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^3}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}+\frac{\left(-3Cda^4+bBda^3+b^2(2Bc+(A-7C)d)a^2-b^3(4Ac-4Cc-5Bd)a\right.}{2b^2\left(a^2+b^2\right)^2f(a+b\tan(e+fx}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a + b
*Tan[e + f*x])^3,x]
```

```
[Out] -(((3*a*b^2*(A*c^3 - c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3) + a
^3*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3 - A*(c^3 - 3*c*d^2)) - 3*a^2*b*((
A - C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)) + b^3*((A - C)*d*(3*c^2 - d^2)
+ B*(c^3 - 3*c*d^2)))*x)/(a^2 + b^2)^3) - ((b^3*(A*c^3 - c^3*C - 3*B*c^2*d
- 3*A*c*d^2 + 3*c*C*d^2 + B*d^3) + 3*a^2*b*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 -
 B*d^3 - A*(c^3 - 3*c*d^2)) + a^3*((A - C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d
^2)) - 3*a*b^2*((A - C)*d*(3*c^2 - d^2) + B*(c^3 - 3*c*d^2)))*Log[Cos[e + f
*x]])/((a^2 + b^2)^3*f) - ((b*c - a*d)*(a^5*b*B*d^2 - 3*a^6*C*d^2 + a^4*b^2
```

```
*d*(B*c - 9*C*d) + a^3*b^3*B*(c^2 + 3*d^2) - b^6*(c*(c*C + 3*B*d) - A*(c^2
- 3*d^2)) - a*b^5*(8*c*(A - C)*d + 3*B*(c^2 - 2*d^2)) + a^2*b^4*(3*c^2*C +
6*B*c*d - 10*C*d^2 - A*(3*c^2 - d^2)))*Log[a + b*Tan[e + f*x]])/(b^4*(a^2 +
 b^2)^3*f) - (d^2*(a^3*b*B*d - 3*a^4*C*d - a*b^3*(2*A*c - 2*c*C - 3*B*d) +
a^2*b^2*(B*c - 6*C*d) - b^4*(B*c + (2*A + C)*d))*Tan[e + f*x])/(b^3*(a^2 +
b^2)^2*f) + ((a^3*b*B*d - 3*a^4*C*d - b^4*(2*B*c + 3*A*d) - a*b^3*(4*A*c -
4*c*C - 5*B*d) + a^2*b^2*(2*B*c + (A - 7*C)*d))*(c + d*Tan[e + f*x])^2)/(2*
b^2*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x])) - ((A*b^2 - a*(b*B - a*C))*(c + d
*Tan[e + f*x])^3)/(2*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^2)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
```

```
!LtQ[n, -1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^3}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}+$$

$$= \frac{\left(a^3bBd-3a^4Cd-b^4(2Bc+3Ad)-ab^3(4Ac\right.}{2b^2\left(a^2+\right.}$$

$$= -\frac{d^2\left(a^3bBd-3a^4Cd-ab^3(2Ac-2cC-3Bd)\right.}{b^3}$$

$$= -\frac{\left(3ab^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2\right.\right.}{}$$

$$= -\frac{\left(3ab^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2\right.\right.}{}$$

$$= -\frac{\left(3ab^2\left(Ac^3-c^3C-3Bc^2d-3Acd^2+3cCd^2\right.\right.}{}$$

Mathematica [A] time = 15.89, size = 1451, normalized size = 1.82

result too large to display

Warning: Unable to verify antiderivative.

```
[In] Integrate[((c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(a + b*Tan[e + f*x])^3,x]
```

```
[Out] ((3*a*b^2*(-(A*c^3) + c^3*C + 3*B*c^2*d + 3*A*c*d^2 - 3*c*C*d^2 - B*d^3) +
a^3*(-(c^3*C) - 3*B*c^2*d + 3*c*C*d^2 + B*d^3 + A*(c^3 - 3*c*d^2)) + b^3*((
A - C)*d*(-3*c^2 + d^2) - B*(c^3 - 3*c*d^2)) + 3*a^2*b*(-((A - C)*d*(-3*c^2
 + d^2)) + B*(c^3 - 3*c*d^2)))*(e + f*x)*(a*Cos[e + f*x] + b*Sin[e + f*x])^
3*(c + d*Tan[e + f*x])^3)/((a^2 + b^2)^3*f*(c*Cos[e + f*x] + d*Sin[e + f*x]
)^3*(a + b*Tan[e + f*x])^3) - (d^2*(3*b*c*C + b*B*d - 3*a*C*d)*Log[1 - Tan[
(e + f*x)/2]^2]*(a*Cos[e + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^3)
/(b^4*f*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^3) + ((3*a
^2*b*(-(A*c^3) + c^3*C + 3*B*c^2*d + 3*A*c*d^2 - 3*c*C*d^2 - B*d^3) + b^3*(
-(c^3*C) - 3*B*c^2*d + 3*c*C*d^2 + B*d^3 + A*(c^3 - 3*c*d^2)) + a^3*(-((A -
 C)*d*(-3*c^2 + d^2)) + B*(c^3 - 3*c*d^2)) - 3*a*b^2*(-((A - C)*d*(-3*c^2 +
 d^2)) + B*(c^3 - 3*c*d^2)))*Log[1 + Tan[(e + f*x)/2]^2]*(a*Cos[e + f*x] +
b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^3)/((a^2 + b^2)^3*f*(c*Cos[e + f*x]
+ d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^3) - ((b*c - a*d)*(a^5*b*B*d^2 - 3
*a^6*C*d^2 + a^4*b^2*d*(B*c - 9*C*d) + a^3*b^3*B*(c^2 + 3*d^2) + b^6*(-(c*(
c*C + 3*B*d)) + A*(c^2 - 3*d^2)) + a*b^5*(8*c*(-A + C)*d - 3*B*(c^2 - 2*d^2
)) + a^2*b^4*(3*c^2*C + 6*B*c*d - 10*C*d^2 + A*(-3*c^2 + d^2)))*Log[-2*b*Ta
n[(e + f*x)/2] + a*(-1 + Tan[(e + f*x)/2]^2)]*(a*Cos[e + f*x] + b*Sin[e + f
*x])^3*(c + d*Tan[e + f*x])^3)/(b^4*(a^2 + b^2)^3*f*(c*Cos[e + f*x] + d*Sin
[e + f*x])^3*(a + b*Tan[e + f*x])^3) - (2*C*d^3*(a*Cos[e + f*x] + b*Sin[e +
 f*x])^3*Tan[(e + f*x)/2]*(c + d*Tan[e + f*x])^3)/(b^3*f*(c*Cos[e + f*x] +
d*Sin[e + f*x])^3*(-1 + Tan[(e + f*x)/2]^2)*(a + b*Tan[e + f*x])^3) + (2*(A
*b^2 + a*(-(b*B) + a*C))*(-(b*c) + a*d)^3*(a*Cos[e + f*x] + b*Sin[e + f*x])
^3*(a + 2*b*Tan[(e + f*x)/2])*(c + d*Tan[e + f*x])^3)/(a^3*b^2*(a^2 + b^2)*
f*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a + 2*b*Tan[(e + f*x)/2] - a*Tan[(e
+ f*x)/2]^2)^2*(a + b*Tan[e + f*x])^3) - (2*(b*c - a*d)^2*(a*Cos[e + f*x] +
 b*Sin[e + f*x])^3*(A*b^6*c + 2*a^6*C*d*Tan[(e + f*x)/2] - a*b^5*(B*c + A*(
d - c*Tan[(e + f*x)/2])) - a^5*b*(B*d*Tan[(e + f*x)/2] + C*(d - c*Tan[(e +
f*x)/2])) + a^4*b^2*(c*(C - 2*B*Tan[(e + f*x)/2]) + d*(B + 4*C*Tan[(e + f*x
)/2])) + a^2*b^4*(c*C + B*d + A*(c + 2*d*Tan[(e + f*x)/2])) - a^3*b^3*(A*d
+ C*d - 3*A*c*Tan[(e + f*x)/2] + c*C*Tan[(e + f*x)/2] + B*(c + 3*d*Tan[(e +
 f*x)/2])))*(c + d*Tan[e + f*x])^3)/(a^3*b^3*(a^2 + b^2)^2*f*(c*Cos[e + f*x
] + d*Sin[e + f*x])^3*(-2*b*Tan[(e + f*x)/2] + a*(-1 + Tan[(e + f*x)/2]^2))
*(a + b*Tan[e + f*x])^3)
```

fricas [B] time = 4.31, size = 2549, normalized size = 3.19

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(C*a^6*b^3 + 3*C*a^4*b^5 + 3*C*a^2*b^7 + C*b^9)*d^3*tan(f*x + e)^3 -
 (3*C*a^4*b^5 - 5*B*a^3*b^6 + (7*A - 3*C)*a^2*b^7 + B*a*b^8 + A*b^9)*c^3 +
3*(C*a^5*b^4 - 3*B*a^4*b^5 + 5*(A - C)*a^3*b^6 + 3*B*a^2*b^7 - A*a*b^8)*c^2
*d + 3*(C*a^6*b^3 + B*a^5*b^4 - (3*A - 7*C)*a^4*b^5 - 5*B*a^3*b^6 + 3*A*a^2
*b^7)*c*d^2 - (3*C*a^7*b^2 - B*a^6*b^3 - (A - 9*C)*a^5*b^4 - 7*B*a^4*b^5 +
5*A*a^3*b^6)*d^3 + 2*(((A - C)*a^5*b^4 + 3*B*a^4*b^5 - 3*(A - C)*a^3*b^6 -
B*a^2*b^7)*c^3 - 3*(B*a^5*b^4 - 3*(A - C)*a^4*b^5 - 3*B*a^3*b^6 + (A - C)*a
^2*b^7)*c^2*d - 3*((A - C)*a^5*b^4 + 3*B*a^4*b^5 - 3*(A - C)*a^3*b^6 - B*a^
2*b^7)*c*d^2 + (B*a^5*b^4 - 3*(A - C)*a^4*b^5 - 3*B*a^3*b^6 + (A - C)*a^2*b
^7)*d^3)*f*x + ((C*a^4*b^5 - 3*B*a^3*b^6 + 5*(A - C)*a^2*b^7 + 3*B*a*b^8 -
A*b^9)*c^3 + 3*(C*a^5*b^4 + B*a^4*b^5 - (3*A - 7*C)*a^3*b^6 - 5*B*a^2*b^7 +
 3*A*a*b^8)*c^2*d - 3*(3*C*a^6*b^3 - B*a^5*b^4 - (A - 9*C)*a^4*b^5 - 7*B*a^
3*b^6 + 5*A*a^2*b^7)*c*d^2 + (9*C*a^7*b^2 - 3*B*a^6*b^3 + (A + 23*C)*a^5*b^
4 - 9*B*a^4*b^5 + (7*A + 12*C)*a^3*b^6 + 4*C*a*b^8)*d^3 + 2*(((A - C)*a^3*b
^6 + 3*B*a^2*b^7 - 3*(A - C)*a*b^8 - B*b^9)*c^3 - 3*(B*a^3*b^6 - 3*(A - C)*
a^2*b^7 - 3*B*a*b^8 + (A - C)*b^9)*c^2*d - 3*((A - C)*a^3*b^6 + 3*B*a^2*b^7
 - 3*(A - C)*a*b^8 - B*b^9)*c*d^2 + (B*a^3*b^6 - 3*(A - C)*a^2*b^7 - 3*B*a*
b^8 + (A - C)*b^9)*d^3)*f*x)*tan(f*x + e)^2 - ((B*a^5*b^4 - 3*(A - C)*a^4*b
^5 - 3*B*a^3*b^6 + (A - C)*a^2*b^7)*c^3 + 3*((A - C)*a^5*b^4 + 3*B*a^4*b^5
 - 3*(A - C)*a^3*b^6 - B*a^2*b^7)*c^2*d - 3*(C*a^8*b + 3*C*a^6*b^3 + B*a^5*b
^4 - 3*(A - 2*C)*a^4*b^5 - 3*B*a^3*b^6 + A*a^2*b^7)*c*d^2 + (3*C*a^9 - B*a^
8*b + 9*C*a^7*b^2 - 3*B*a^6*b^3 - (A - 10*C)*a^5*b^4 - 6*B*a^4*b^5 + 3*A*a^
3*b^6)*d^3 + ((B*a^3*b^6 - 3*(A - C)*a^2*b^7 - 3*B*a*b^8 + (A - C)*b^9)*c^3
 + 3*((A - C)*a^3*b^6 + 3*B*a^2*b^7 - 3*(A - C)*a*b^8 - B*b^9)*c^2*d - 3*(C
*a^6*b^3 + 3*C*a^4*b^5 + B*a^3*b^6 - 3*(A - 2*C)*a^2*b^7 - 3*B*a*b^8 + A*b^
9)*c*d^2 + (3*C*a^7*b^2 - B*a^6*b^3 + 9*C*a^5*b^4 - 3*B*a^4*b^5 - (A - 10*C
)*a^3*b^6 - 6*B*a^2*b^7 + 3*A*a*b^8)*d^3)*tan(f*x + e)^2 + 2*((B*a^4*b^5 -
3*(A - C)*a^3*b^6 - 3*B*a^2*b^7 + (A - C)*a*b^8)*c^3 + 3*((A - C)*a^4*b^5 +
 3*B*a^3*b^6 - 3*(A - C)*a^2*b^7 - B*a*b^8)*c^2*d - 3*(C*a^7*b^2 + 3*C*a^5*
b^4 + B*a^4*b^5 - 3*(A - 2*C)*a^3*b^6 - 3*B*a^2*b^7 + A*a*b^8)*c*d^2 + (3*C
*a^8*b - B*a^7*b^2 + 9*C*a^6*b^3 - 3*B*a^5*b^4 - (A - 10*C)*a^4*b^5 - 6*B*a
^3*b^6 + 3*A*a^2*b^7)*d^3)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*ta
n(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - (3*(C*a^8*b + 3*C*a^6*b^3 + 3*C*a
^4*b^5 + C*a^2*b^7)*c*d^2 - (3*C*a^9 - B*a^8*b + 9*C*a^7*b^2 - 3*B*a^6*b^3
+ 9*C*a^5*b^4 - 3*B*a^4*b^5 + 3*C*a^3*b^6 - B*a^2*b^7)*d^3 + (3*(C*a^6*b^3
+ 3*C*a^4*b^5 + 3*C*a^2*b^7 + C*b^9)*c*d^2 - (3*C*a^7*b^2 - B*a^6*b^3 + 9*C
*a^5*b^4 - 3*B*a^4*b^5 + 9*C*a^3*b^6 - 3*B*a^2*b^7 + 3*C*a*b^8 - B*b^9)*d^3
)*tan(f*x + e)^2 + 2*(3*(C*a^7*b^2 + 3*C*a^5*b^4 + 3*C*a^3*b^6 + C*a*b^8)*c
*d^2 - (3*C*a^8*b - B*a^7*b^2 + 9*C*a^6*b^3 - 3*B*a^5*b^4 + 9*C*a^4*b^5 - 3
*B*a^3*b^6 + 3*C*a^2*b^7 - B*a*b^8)*d^3)*tan(f*x + e))*log(1/(tan(f*x + e)^
2 + 1)) + 2*((C*a^5*b^4 - 2*B*a^4*b^5 + 3*(A - C)*a^3*b^6 + 3*B*a^2*b^7 - (
3*A - 2*C)*a*b^8 - B*b^9)*c^3 + 3*(B*a^5*b^4 - (2*A - 3*C)*a^4*b^5 - 3*B*a^
```

```
3*b^6 + 3*(A - C)*a^2*b^7 + 2*B*a*b^8 - A*b^9)*c^2*d - 3*(C*a^7*b^2 - (A -
3*C)*a^5*b^4 - 3*B*a^4*b^5 + (3*A - 4*C)*a^3*b^6 + 3*B*a^2*b^7 - 2*A*a*b^8)
*c*d^2 + (3*C*a^8*b - B*a^7*b^2 + 6*C*a^6*b^3 - 3*B*a^5*b^4 + (3*A - 2*C)*a
^4*b^5 + 4*B*a^3*b^6 - (3*A - C)*a^2*b^7)*d^3 + 2*(((A - C)*a^4*b^5 + 3*B*a
^3*b^6 - 3*(A - C)*a^2*b^7 - B*a*b^8)*c^3 - 3*(B*a^4*b^5 - 3*(A - C)*a^3*b^
6 - 3*B*a^2*b^7 + (A - C)*a*b^8)*c^2*d - 3*((A - C)*a^4*b^5 + 3*B*a^3*b^6 -
 3*(A - C)*a^2*b^7 - B*a*b^8)*c*d^2 + (B*a^4*b^5 - 3*(A - C)*a^3*b^6 - 3*B*
a^2*b^7 + (A - C)*a*b^8)*d^3)*f*x)*tan(f*x + e))/((a^6*b^6 + 3*a^4*b^8 + 3*
a^2*b^10 + b^12)*f*tan(f*x + e)^2 + 2*(a^7*b^5 + 3*a^5*b^7 + 3*a^3*b^9 + a*
b^11)*f*tan(f*x + e) + (a^8*b^4 + 3*a^6*b^6 + 3*a^4*b^8 + a^2*b^10)*f)
```

giac [B] time = 7.35, size = 2505, normalized size = 3.14

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*d^3*tan(f*x + e)/b^3 + 2*(A*a^3*c^3 - C*a^3*c^3 + 3*B*a^2*b*c^3 -
3*A*a*b^2*c^3 + 3*C*a*b^2*c^3 - B*b^3*c^3 - 3*B*a^3*c^2*d + 9*A*a^2*b*c^2*d
 - 9*C*a^2*b*c^2*d + 9*B*a*b^2*c^2*d - 3*A*b^3*c^2*d + 3*C*b^3*c^2*d - 3*A*
a^3*c*d^2 + 3*C*a^3*c*d^2 - 9*B*a^2*b*c*d^2 + 9*A*a*b^2*c*d^2 - 9*C*a*b^2*c
*d^2 + 3*B*b^3*c*d^2 + B*a^3*d^3 - 3*A*a^2*b*d^3 + 3*C*a^2*b*d^3 - 3*B*a*b^
2*d^3 + A*b^3*d^3 - C*b^3*d^3)*(f*x + e)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6
) + (B*a^3*c^3 - 3*A*a^2*b*c^3 + 3*C*a^2*b*c^3 - 3*B*a*b^2*c^3 + A*b^3*c^3
- C*b^3*c^3 + 3*A*a^3*c^2*d - 3*C*a^3*c^2*d + 9*B*a^2*b*c^2*d - 9*A*a*b^2*c
^2*d + 9*C*a*b^2*c^2*d - 3*B*b^3*c^2*d - 3*B*a^3*c*d^2 + 9*A*a^2*b*c*d^2 -
9*C*a^2*b*c*d^2 + 9*B*a*b^2*c*d^2 - 3*A*b^3*c*d^2 + 3*C*b^3*c*d^2 - A*a^3*d
^3 + C*a^3*d^3 - 3*B*a^2*b*d^3 + 3*A*a*b^2*d^3 - 3*C*a*b^2*d^3 + B*b^3*d^3)
*log(tan(f*x + e)^2 + 1)/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - 2*(B*a^3*b^4
*c^3 - 3*A*a^2*b^5*c^3 + 3*C*a^2*b^5*c^3 - 3*B*a*b^6*c^3 + A*b^7*c^3 - C*b^
7*c^3 + 3*A*a^3*b^4*c^2*d - 3*C*a^3*b^4*c^2*d + 9*B*a^2*b^5*c^2*d - 9*A*a*b
^6*c^2*d + 9*C*a*b^6*c^2*d - 3*B*b^7*c^2*d - 3*C*a^6*b*c*d^2 - 9*C*a^4*b^3*
c*d^2 - 3*B*a^3*b^4*c*d^2 + 9*A*a^2*b^5*c*d^2 - 18*C*a^2*b^5*c*d^2 + 9*B*a*
b^6*c*d^2 - 3*A*b^7*c*d^2 + 3*C*a^7*d^3 - B*a^6*b*d^3 + 9*C*a^5*b^2*d^3 - 3
*B*a^4*b^3*d^3 - A*a^3*b^4*d^3 + 10*C*a^3*b^4*d^3 - 6*B*a^2*b^5*d^3 + 3*A*a
*b^6*d^3)*log(abs(b*tan(f*x + e) + a))/(a^6*b^4 + 3*a^4*b^6 + 3*a^2*b^8 + b
^10) + (3*B*a^3*b^6*c^3*tan(f*x + e)^2 - 9*A*a^2*b^7*c^3*tan(f*x + e)^2 + 9
*C*a^2*b^7*c^3*tan(f*x + e)^2 - 9*B*a*b^8*c^3*tan(f*x + e)^2 + 3*A*b^9*c^3*
tan(f*x + e)^2 - 3*C*b^9*c^3*tan(f*x + e)^2 + 9*A*a^3*b^6*c^2*d*tan(f*x + e
)^2 - 9*C*a^3*b^6*c^2*d*tan(f*x + e)^2 + 27*B*a^2*b^7*c^2*d*tan(f*x + e)^2
- 27*A*a*b^8*c^2*d*tan(f*x + e)^2 + 27*C*a*b^8*c^2*d*tan(f*x + e)^2 - 9*B*b
^9*c^2*d*tan(f*x + e)^2 - 9*C*a^6*b^3*c*d^2*tan(f*x + e)^2 - 27*C*a^4*b^5*c
*d^2*tan(f*x + e)^2 - 9*B*a^3*b^6*c*d^2*tan(f*x + e)^2 + 27*A*a^2*b^7*c*d^2
```

```
*tan(f*x + e)^2 - 54*C*a^2*b^7*c*d^2*tan(f*x + e)^2 + 27*B*a*b^8*c*d^2*tan(
f*x + e)^2 - 9*A*b^9*c*d^2*tan(f*x + e)^2 + 9*C*a^7*b^2*d^3*tan(f*x + e)^2
- 3*B*a^6*b^3*d^3*tan(f*x + e)^2 + 27*C*a^5*b^4*d^3*tan(f*x + e)^2 - 9*B*a^
4*b^5*d^3*tan(f*x + e)^2 - 3*A*a^3*b^6*d^3*tan(f*x + e)^2 + 30*C*a^3*b^6*d^
3*tan(f*x + e)^2 - 18*B*a^2*b^7*d^3*tan(f*x + e)^2 + 9*A*a*b^8*d^3*tan(f*x
+ e)^2 + 8*B*a^4*b^5*c^3*tan(f*x + e) - 22*A*a^3*b^6*c^3*tan(f*x + e) + 22*
C*a^3*b^6*c^3*tan(f*x + e) - 18*B*a^2*b^7*c^3*tan(f*x + e) + 2*A*a*b^8*c^3*
tan(f*x + e) - 2*C*a*b^8*c^3*tan(f*x + e) - 2*B*b^9*c^3*tan(f*x + e) - 6*C*
a^6*b^3*c^2*d*tan(f*x + e) + 24*A*a^4*b^5*c^2*d*tan(f*x + e) - 42*C*a^4*b^5
*c^2*d*tan(f*x + e) + 66*B*a^3*b^6*c^2*d*tan(f*x + e) - 54*A*a^2*b^7*c^2*d*
tan(f*x + e) + 36*C*a^2*b^7*c^2*d*tan(f*x + e) - 6*B*a*b^8*c^2*d*tan(f*x +
e) - 6*A*b^9*c^2*d*tan(f*x + e) - 6*C*a^7*b^2*c*d^2*tan(f*x + e) - 6*B*a^6*
b^3*c*d^2*tan(f*x + e) - 18*C*a^5*b^4*c*d^2*tan(f*x + e) - 42*B*a^4*b^5*c*d
^2*tan(f*x + e) + 66*A*a^3*b^6*c*d^2*tan(f*x + e) - 84*C*a^3*b^6*c*d^2*tan(
f*x + e) + 36*B*a^2*b^7*c*d^2*tan(f*x + e) - 6*A*a*b^8*c*d^2*tan(f*x + e) +
 12*C*a^8*b*d^3*tan(f*x + e) - 2*B*a^7*b^2*d^3*tan(f*x + e) - 2*A*a^6*b^3*d
^3*tan(f*x + e) + 38*C*a^6*b^3*d^3*tan(f*x + e) - 6*B*a^5*b^4*d^3*tan(f*x +
 e) - 14*A*a^4*b^5*d^3*tan(f*x + e) + 50*C*a^4*b^5*d^3*tan(f*x + e) - 28*B*
a^3*b^6*d^3*tan(f*x + e) + 12*A*a^2*b^7*d^3*tan(f*x + e) - C*a^6*b^3*c^3 +
6*B*a^5*b^4*c^3 - 14*A*a^4*b^5*c^3 + 11*C*a^4*b^5*c^3 - 7*B*a^3*b^6*c^3 - 3
*A*a^2*b^7*c^3 - B*a*b^8*c^3 - A*b^9*c^3 - 3*C*a^7*b^2*c^2*d - 3*B*a^6*b^3*
c^2*d + 18*A*a^5*b^4*c^2*d - 27*C*a^5*b^4*c^2*d + 33*B*a^4*b^5*c^2*d - 21*A
*a^3*b^6*c^2*d + 12*C*a^3*b^6*c^2*d - 3*A*a*b^8*c^2*d - 3*B*a^7*b^2*c*d^2 -
 3*A*a^6*b^3*c*d^2 + 3*C*a^6*b^3*c*d^2 - 27*B*a^5*b^4*c*d^2 + 33*A*a^4*b^5*
c*d^2 - 33*C*a^4*b^5*c*d^2 + 12*B*a^3*b^6*c*d^2 + 4*C*a^9*d^3 - A*a^7*b^2*d
^3 + 13*C*a^7*b^2*d^3 + B*a^6*b^3*d^3 - 9*A*a^5*b^4*d^3 + 21*C*a^5*b^4*d^3
- 11*B*a^4*b^5*d^3 + 4*A*a^3*b^6*d^3)/((a^6*b^4 + 3*a^4*b^6 + 3*a^2*b^8 + b
^10)*(b*tan(f*x + e) + a)^2))/f
```

maple [B] time = 0.32, size = 3522, normalized size = 4.41

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3,x
)
```

```
[Out] 3/2/f/(a^2+b^2)/(a+b*tan(f*x+e))^2*A*a*c^2*d+3/f*b/(a^2+b^2)^3*ln(a+b*tan(f
*x+e))*A*a^2*c^3-3/f*b^2/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a*d^3+3/f*b^3/(a^
2+b^2)^3*ln(a+b*tan(f*x+e))*A*c*d^2+1/f/b^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*
B*a^6*d^3-3/f/(a^2+b^2)^3*B*arctan(tan(f*x+e))*a*b^2*d^3+1/2/f/b^2/(a^2+b^2
)/(a+b*tan(f*x+e))^2*A*a^3*d^3+3/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^3*c*d
^2+3/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*a^3*c^2*d-9/f/(a^2+b^2)^2/(a+b*tan(
f*x+e))*B*a^2*c*d^2-9/f/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a^2*c^2*d-3/f/(a^2+b
^2)^3*ln(a+b*tan(f*x+e))*A*a^3*c^2*d+3/f/(a^2+b^2)^3*C*arctan(tan(f*x+e))*a
```

```
^3*c*d^2-3/2/f/b^3/(a^2+b^2)/(a+b*tan(f*x+e))^2*C*a^4*c*d^2+6/f/b^3/(a^2+b^
2)^2/(a+b*tan(f*x+e))*C*a^5*c*d^2-3/f/b^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a^
4*c^2*d+12/f/b/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a^3*c*d^2+9/f/(a^2+b^2)^3*B*a
rctan(tan(f*x+e))*a*b^2*c^2*d-3/f/b^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a^4*c*
d^2+18/f*b/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*a^2*c*d^2-9/f*b^2/(a^2+b^2)^3*l
n(a+b*tan(f*x+e))*C*a*c^2*d-9/f/(a^2+b^2)^3*B*arctan(tan(f*x+e))*a^2*b*c*d^
2+6/f*b/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a*c^2*d+3/2/f/b^2/(a^2+b^2)/(a+b*tan
(f*x+e))^2*C*a^3*c^2*d-3/2/f/b/(a^2+b^2)/(a+b*tan(f*x+e))^2*B*a^2*c^2*d-3/2
/f/b/(a^2+b^2)/(a+b*tan(f*x+e))^2*A*a^2*c*d^2+3/2/f/b^2/(a^2+b^2)/(a+b*tan(
f*x+e))^2*B*a^3*c*d^2+9/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*a*b^2*c^2*d+9/
f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*a^2*b*c^2*d+9/f/(a^2+b^2)^3*A*arctan(tan
(f*x+e))*a*b^2*c*d^2-9/f/(a^2+b^2)^3*C*arctan(tan(f*x+e))*a*b^2*c*d^2+9/f*b
^2/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a*c^2*d-9/f*b/(a^2+b^2)^3*ln(a+b*tan(f*
x+e))*A*a^2*c*d^2+6/f*b/(a^2+b^2)^2/(a+b*tan(f*x+e))*A*a*c*d^2-9/f/(a^2+b^2
)^3*C*arctan(tan(f*x+e))*a^2*b*c^2*d+9/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A
*a^2*b*c*d^2-9/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A*a*b^2*c^2*d+9/2/f/(a^2+
b^2)^3*ln(1+tan(f*x+e)^2)*B*a^2*b*c^2*d+9/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2
)*B*a*b^2*c*d^2-9/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*a^2*b*c*d^2-9/f*b/(a
^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^2*c^2*d-9/f*b^2/(a^2+b^2)^3*ln(a+b*tan(f*x
+e))*B*a*c*d^2+3/f/b^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*a^6*c*d^2+9/f/b/(a^
2+b^2)^3*ln(a+b*tan(f*x+e))*C*a^4*c*d^2+3/f/(a^2+b^2)^3*C*arctan(tan(f*x+e)
)*a^2*b*d^3+3/f/(a^2+b^2)^3*C*arctan(tan(f*x+e))*a*b^2*c^3+3/f/(a^2+b^2)^3*
C*arctan(tan(f*x+e))*b^3*c^2*d+1/2/f/b^4/(a^2+b^2)/(a+b*tan(f*x+e))^2*C*a^5
*d^3-1/2/f/b/(a^2+b^2)/(a+b*tan(f*x+e))^2*C*a^2*c^3-1/f/b^2/(a^2+b^2)^2/(a+
b*tan(f*x+e))*A*a^4*d^3-2/f*b/(a^2+b^2)^2/(a+b*tan(f*x+e))*A*a*c^3-3/f*b^2/
(a^2+b^2)^2/(a+b*tan(f*x+e))*A*c^2*d+2/f/b^3/(a^2+b^2)^2/(a+b*tan(f*x+e))*B
*a^5*d^3+4/f/b/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a^3*d^3-3/f/b^4/(a^2+b^2)^2/(
a+b*tan(f*x+e))*C*a^6*d^3-5/f/b^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a^4*d^3+2/
f*b/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a*c^3-3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^
2)*C*a*b^2*d^3-1/2/f/b^3/(a^2+b^2)/(a+b*tan(f*x+e))^2*B*a^4*d^3+3/2/f/(a^2+
b^2)^3*ln(1+tan(f*x+e)^2)*C*b^3*c*d^2-3/f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*
a^3*c*d^2-3/f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*a^2*b*d^3+1/f/(a^2+b^2)^3*A*
arctan(tan(f*x+e))*a^3*c^3+1/f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*b^3*d^3+1/f
/(a^2+b^2)^3*B*arctan(tan(f*x+e))*a^3*d^3-1/f/(a^2+b^2)^3*B*arctan(tan(f*x+
e))*b^3*c^3-1/f/(a^2+b^2)^3*C*arctan(tan(f*x+e))*a^3*c^3-1/f/(a^2+b^2)^3*C*
arctan(tan(f*x+e))*b^3*d^3-1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A*a^3*d^3+1
/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a^3*d^3-1/f/(a^2+b^2)^3*ln(a+b*tan(f*x+
e))*B*a^3*c^3-10/f/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*a^3*d^3+1/2/f/(a^2+b^2)
/(a+b*tan(f*x+e))^2*B*a*c^3-3/f/(a^2+b^2)^2/(a+b*tan(f*x+e))*A*a^2*d^3+1/f/
(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a^2*c^3-1/2/f*b/(a^2+b^2)/(a+b*tan(f*x+e))^2
*A*c^3-1/f*b^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*c^3-1/f*b^3/(a^2+b^2)^3*ln(a+
b*tan(f*x+e))*A*c^3+1/f*b^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*c^3+1/2/f/(a^2
+b^2)^3*ln(1+tan(f*x+e)^2)*A*b^3*c^3+1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*B
*a^3*c^3+1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*B*b^3*d^3+1/2/f/(a^2+b^2)^3*l
n(1+tan(f*x+e)^2)*a^3*C*d^3-1/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*b^3*c^3+
```

```
3/f/(a^2+b^2)^3*B*arctan(tan(f*x+e))*b^3*c*d^2+3/f/(a^2+b^2)^2/(a+b*tan(f*x
+e))*A*a^2*c^2*d+3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A*a^3*c^2*d-3/2/f/(a^
2+b^2)^3*ln(1+tan(f*x+e)^2)*A*a^2*b*c^3+3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2
)*A*a*b^2*d^3-3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*A*b^3*c*d^2-3/2/f/(a^2+b
^2)^3*ln(1+tan(f*x+e)^2)*B*a^3*c*d^2-3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*B
*a^2*b*d^3-3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*B*a*b^2*c^3-3/2/f/(a^2+b^2)
^3*ln(1+tan(f*x+e)^2)*B*b^3*c^2*d-3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*a^
3*c^2*d+3/2/f/(a^2+b^2)^3*ln(1+tan(f*x+e)^2)*C*a^2*b*c^3-3/f/(a^2+b^2)^3*A*
arctan(tan(f*x+e))*a*b^2*c^3-3/f/(a^2+b^2)^3*A*arctan(tan(f*x+e))*b^3*c^2*d
-3/f/(a^2+b^2)^3*B*arctan(tan(f*x+e))*a^3*c^2*d+3/f/(a^2+b^2)^3*B*arctan(ta
n(f*x+e))*a^2*b*c^3+3/f/b/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^4*d^3+6/f*b/(a
^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^2*d^3+3/f*b^2/(a^2+b^2)^3*ln(a+b*tan(f*x+e
))*B*a*c^3+3/f*b^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*c^2*d-3/f/b^4/(a^2+b^2)
^3*ln(a+b*tan(f*x+e))*C*a^7*d^3-9/f/b^2/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*a^
5*d^3-3/f*b/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*a^2*c^3+1/f*C*d^3/b^3*tan(f*x+
e)
```

maxima [A] time = 0.55, size = 1119, normalized size = 1.40

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*d^3*tan(f*x + e)/b^3 + 2*(((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b
^2 - B*b^3)*c^3 - 3*(B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c^2
*d - 3*((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c*d^2 + (B*a^3 -
 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*d^3)*(f*x + e)/(a^6 + 3*a^4*b^2
 + 3*a^2*b^4 + b^6) - 2*((B*a^3*b^4 - 3*(A - C)*a^2*b^5 - 3*B*a*b^6 + (A -
C)*b^7)*c^3 + 3*((A - C)*a^3*b^4 + 3*B*a^2*b^5 - 3*(A - C)*a*b^6 - B*b^7)*c
^2*d - 3*(C*a^6*b + 3*C*a^4*b^3 + B*a^3*b^4 - 3*(A - 2*C)*a^2*b^5 - 3*B*a*b
^6 + A*b^7)*c*d^2 + (3*C*a^7 - B*a^6*b + 9*C*a^5*b^2 - 3*B*a^4*b^3 - (A - 1
0*C)*a^3*b^4 - 6*B*a^2*b^5 + 3*A*a*b^6)*d^3)*log(b*tan(f*x + e) + a)/(a^6*b
^4 + 3*a^4*b^6 + 3*a^2*b^8 + b^10) + ((B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2
+ (A - C)*b^3)*c^3 + 3*((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*
c^2*d - 3*(B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c*d^2 - ((A -
 C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*d^3)*log(tan(f*x + e)^2 + 1)
/(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6) - ((C*a^4*b^3 - 3*B*a^3*b^4 + (5*A - 3
*C)*a^2*b^5 + B*a*b^6 + A*b^7)*c^3 + 3*(C*a^5*b^2 + B*a^4*b^3 - (3*A - 5*C)
*a^3*b^4 - 3*B*a^2*b^5 + A*a*b^6)*c^2*d - 3*(3*C*a^6*b - B*a^5*b^2 - (A - 7
*C)*a^4*b^3 - 5*B*a^3*b^4 + 3*A*a^2*b^5)*c*d^2 + (5*C*a^7 - 3*B*a^6*b + (A
+ 9*C)*a^5*b^2 - 7*B*a^4*b^3 + 5*A*a^3*b^4)*d^3 - 2*((B*a^2*b^5 - 2*(A - C)
*a*b^6 - B*b^7)*c^3 - 3*(C*a^4*b^3 - (A - 3*C)*a^2*b^5 - 2*B*a*b^6 + A*b^7)
*c^2*d + 3*(2*C*a^5*b^2 - B*a^4*b^3 + 4*C*a^3*b^4 - 3*B*a^2*b^5 + 2*A*a*b^6
```

```
)*c*d^2 - (3*C*a^6*b - 2*B*a^5*b^2 + (A + 5*C)*a^4*b^3 - 4*B*a^3*b^4 + 3*A*
a^2*b^5)*d^3)*tan(f*x + e))/(a^6*b^4 + 2*a^4*b^6 + a^2*b^8 + (a^4*b^6 + 2*a
^2*b^8 + b^10)*tan(f*x + e)^2 + 2*(a^5*b^5 + 2*a^3*b^7 + a*b^9)*tan(f*x + e
)))/f
```

mupad [B] time = 19.24, size = 1172, normalized size = 1.47

$$\frac{\ln\left(a+b\tan\left(e+f\,x\right)\right)\,\left(b^3\,\left(3\,B\,a^4\,d^3+9\,C\,c\,a^4\,d^2\right)-b^6\,\left(3\,A\,a\,d^3-3\,B\,a\,c^3-9\,A\,a\,c^2\,d+9\,B\,a\,c\,d^2+9\,C\,a\,c^2\,d\right)\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a + b
*tan(e + f*x))^3,x)
```

```
[Out] (log(tan(e + f*x) + 1i)*(A*c^3 + A*d^3*1i - B*c^3*1i + B*d^3 - C*c^3 - C*d^
3*1i - 3*A*c*d^2 - A*c^2*d*3i + B*c*d^2*3i - 3*B*c^2*d + 3*C*c*d^2 + C*c^2*
d*3i))/(2*f*(a*b^2*3i - 3*a^2*b - a^3*1i + b^3)) - ((tan(e + f*x)*(B*b^6*c^
3 + 3*C*a^6*d^3 + 2*A*a*b^5*c^3 - 2*B*a^5*b*d^3 - 2*C*a*b^5*c^3 + 3*A*b^6*c
^2*d + 3*A*a^2*b^4*d^3 + A*a^4*b^2*d^3 - B*a^2*b^4*c^3 - 4*B*a^3*b^3*d^3 +
5*C*a^4*b^2*d^3 - 3*A*a^2*b^4*c^2*d + 9*B*a^2*b^4*c*d^2 + 3*B*a^4*b^2*c*d^2
 + 9*C*a^2*b^4*c^2*d - 12*C*a^3*b^3*c*d^2 + 3*C*a^4*b^2*c^2*d - 6*A*a*b^5*c
*d^2 - 6*B*a*b^5*c^2*d - 6*C*a^5*b*c*d^2))/(a^4 + b^4 + 2*a^2*b^2) + (A*b^7
*c^3 + 5*C*a^7*d^3 + B*a*b^6*c^3 - 3*B*a^6*b*d^3 + 5*A*a^2*b^5*c^3 + 5*A*a^
3*b^4*d^3 + A*a^5*b^2*d^3 - 3*B*a^3*b^4*c^3 - 7*B*a^4*b^3*d^3 - 3*C*a^2*b^5
*c^3 + C*a^4*b^3*c^3 + 9*C*a^5*b^2*d^3 - 9*A*a^2*b^5*c*d^2 - 9*A*a^3*b^4*c^
2*d + 3*A*a^4*b^3*c*d^2 - 9*B*a^2*b^5*c^2*d + 15*B*a^3*b^4*c*d^2 + 3*B*a^4*
b^3*c^2*d + 3*B*a^5*b^2*c*d^2 + 15*C*a^3*b^4*c^2*d - 21*C*a^4*b^3*c*d^2 + 3
*C*a^5*b^2*c^2*d + 3*A*a*b^6*c^2*d - 9*C*a^6*b*c*d^2)/(2*b*(a^4 + b^4 + 2*a
^2*b^2)))/(f*(a^2*b^3 + b^5*tan(e + f*x)^2 + 2*a*b^4*tan(e + f*x))) + (log(
a + b*tan(e + f*x))*(b^3*(3*B*a^4*d^3 + 9*C*a^4*c*d^2) - b^6*(3*A*a*d^3 - 3
*B*a*c^3 - 9*A*a*c^2*d + 9*B*a*c*d^2 + 9*C*a*c^2*d) + b^5*(3*A*a^2*c^3 + 6*
B*a^2*d^3 - 3*C*a^2*c^3 - 9*A*a^2*c*d^2 - 9*B*a^2*c^2*d + 18*C*a^2*c*d^2) +
 b^4*(A*a^3*d^3 - B*a^3*c^3 - 10*C*a^3*d^3 - 3*A*a^3*c^2*d + 3*B*a^3*c*d^2
+ 3*C*a^3*c^2*d) + b*(B*a^6*d^3 + 3*C*a^6*c*d^2) + b^7*(C*c^3 - A*c^3 + 3*A
*c*d^2 + 3*B*c^2*d) - 3*C*a^7*d^3 - 9*C*a^5*b^2*d^3))/(f*(b^10 + 3*a^2*b^8
+ 3*a^4*b^6 + a^6*b^4)) + (log(tan(e + f*x) - 1i)*(A*c^3*1i + A*d^3 - B*c^3
 + B*d^3*1i - C*c^3*1i - C*d^3 - A*c*d^2*3i - 3*A*c^2*d + 3*B*c*d^2 - B*c^2
*d*3i + C*c*d^2*3i + 3*C*c^2*d))/(2*f*(3*a*b^2 - a^2*b*3i - a^3 + b^3*1i))
+ (C*d^3*tan(e + f*x))/(b^3*f)
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x
+e))**3,x)
```

```
[Out] Exception raised: AttributeError
```

3.70 $\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$

Optimal. Leaf size=337

$$-\frac{\log(\cos(e+fx))\left(a^3(Bc-d(A-C))+3a^2b(Ac+Bd-cC)-3ab^2(Bc-d(A-C))-b^3(Ac+Bd-cC)\right)}{f\left(c^2+d^2\right)}+\frac{x\left(a\right.}{}$$

```
[Out] (a^3*(A*c+B*d-C*c)-3*a*b^2*(A*c+B*d-C*c)-3*a^2*b*(B*c-(A-C)*d)+b^3*(B*c-(A-
C)*d))*x/(c^2+d^2)-(3*a^2*b*(A*c+B*d-C*c)-b^3*(A*c+B*d-C*c)+a^3*(B*c-(A-C)*
d)-3*a*b^2*(B*c-(A-C)*d))*ln(cos(f*x+e))/(c^2+d^2)/f-(-a*d+b*c)^3*(A*d^2-B*
c*d+C*c^2)*ln(c+d*tan(f*x+e))/d^4/(c^2+d^2)/f+b*(b*(A*b+B*a-C*b)*d^2+(-a*d+
b*c)*(-B*b*d-C*a*d+C*b*c))*tan(f*x+e)/d^3/f-1/2*(-B*b*d-C*a*d+C*b*c)*(a+b*t
an(f*x+e))^2/d^2/f+1/3*C*(a+b*tan(f*x+e))^3/d/f
```

Rubi [A] time = 1.59, antiderivative size = 337, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3647, 3637, 3626, 3617, 31, 3475}

$$-\frac{\log(\cos(e+fx))\left(3a^2b(Ac+Bd-cC)+a^3(Bc-d(A-C))-3ab^2(Bc-d(A-C))-b^3(Ac+Bd-cC)\right)}{f\left(c^2+d^2\right)}+\frac{x\left(-\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x]),x]
```

```
[Out] ((a^3*(A*c - c*C + B*d) - 3*a*b^2*(A*c - c*C + B*d) - 3*a^2*b*(B*c - (A - C
)*d) + b^3*(B*c - (A - C)*d))*x)/(c^2 + d^2) - ((3*a^2*b*(A*c - c*C + B*d)
- b^3*(A*c - c*C + B*d) + a^3*(B*c - (A - C)*d) - 3*a*b^2*(B*c - (A - C)*d)
)*Log[Cos[e + f*x]])/((c^2 + d^2)*f) - ((b*c - a*d)^3*(c^2*C - B*c*d + A*d^
2)*Log[c + d*Tan[e + f*x]])/(d^4*(c^2 + d^2)*f) + (b*(b*(A*b + a*B - b*C)*d
^2 + (b*c - a*d)*(b*c*C - b*B*d - a*C*d))*Tan[e + f*x])/(d^3*f) - ((b*c*C -
 b*B*d - a*C*d)*(a + b*Tan[e + f*x])^2)/(2*d^2*f) + (C*(a + b*Tan[e + f*x])
^3)/(3*d*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx = \frac{C(a+b\tan(e+fx))^3}{3df} + \frac{\int \frac{(a+b\tan(e+fx))^2(-3(b}{}}{}$$

$$= -\frac{(bcC-bBd-aCd)(a+b\tan(e+fx))^2}{2d^2f} + $$

$$= \frac{b\left(b(Ab+aB-bC)d^2+(bc-ad)(bcC-bBd\right.}{d^3f}$$

$$= \frac{\left(a^3(Ac-cC+Bd)-3ab^2(Ac-cC+Bd)-\right.}{c^2}$$

$$= \frac{\left(a^3(Ac-cC+Bd)-3ab^2(Ac-cC+Bd)-\right.}{c^2}$$

$$= \frac{\left(a^3(Ac-cC+Bd)-3ab^2(Ac-cC+Bd)-\right.}{c^2}$$

Mathematica [C] time = 4.52, size = 258, normalized size = 0.77

$$6b^2d\tan(e+fx)(aB+Ab-bC)+\frac{6(ad-bc)^3\left(Ad^2-Bcd+c^2C\right)\log(c+d\tan(e+fx))}{d^2\left(c^2+d^2\right)}+\frac{3d^2(a-ib)^3(iA+B-iC)\log(\tan(e+fx)+i)}{c-id}+\frac{3d^2(a}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x]),x]
```

```
[Out] ((3*(a + I*b)^3*((-I)*A + B + I*C)*d^2*Log[I - Tan[e + f*x]])/(c + I*d) + (
3*(a - I*b)^3*(I*A + B - I*C)*d^2*Log[I + Tan[e + f*x]])/(c - I*d) + (6*(-(
b*c) + a*d)^3*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d^2*(c^2 +
d^2)) + 6*b^2*(A*b + a*B - b*C)*d*Tan[e + f*x] - (6*b*(b*c - a*d)*(-(b*c*C)
 + b*B*d + a*C*d)*Tan[e + f*x])/d - 3*(b*c*C - b*B*d - a*C*d)*(a + b*Tan[e
+ f*x])^2 + 2*C*d*(a + b*Tan[e + f*x])^3)/(6*d^2*f)
```

fricas [A] time = 2.73, size = 627, normalized size = 1.86

$$2\left(Cb^3c^2d^3+Cb^3d^5\right)\tan\left(fx+e\right)^3+6\left(\left((A-C)a^3-3\,Ba^2b-3\,(A-C)ab^2+Bb^3\right)cd^4+\left(Ba^3+3\,(A-C)a^2b-\right.\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] 1/6*(2*(C*b^3*c^2*d^3 + C*b^3*d^5)*tan(f*x + e)^3 + 6*(((A - C)*a^3 - 3*B*a
^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d^4 + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^
2 - (A - C)*b^3)*d^5)*f*x - 3*(C*b^3*c^3*d^2 + C*b^3*c*d^4 - (3*C*a*b^2 + B
*b^3)*c^2*d^3 - (3*C*a*b^2 + B*b^3)*d^5)*tan(f*x + e)^2 - 3*(C*b^3*c^5 - A*
a^3*d^5 - (3*C*a*b^2 + B*b^3)*c^4*d + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^3*d
^2 - (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^3 + (B*a^3 + 3*A*a^2*b)*c*d^4)*l
og((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) +
3*(C*b^3*c^5 - (3*C*a*b^2 + B*b^3)*c^4*d + (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*
c^3*d^2 - (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^3 + (3*C*a^2*b + 3*B*a*b^2
+ (A - C)*b^3)*c*d^4 - (C*a^3 + 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*d^5)*l
og(1/(tan(f*x + e)^2 + 1)) + 6*(C*b^3*c^4*d - (3*C*a*b^2 + B*b^3)*c^3*d^2 +
 (3*C*a^2*b + 3*B*a*b^2 + A*b^3)*c^2*d^3 - (3*C*a*b^2 + B*b^3)*c*d^4 + (3*C
*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*d^5)*tan(f*x + e))/((c^2*d^4 + d^6)*f)
```

giac [A] time = 5.52, size = 573, normalized size = 1.70

$$\frac{\frac{6\left(Aa^3c-Ca^3c-3\,Ba^2bc-3\,Aab^2c+3\,Cab^2c+Bb^3c+Ba^3d+3\,Aa^2bd-3\,Ca^2bd-3\,Bab^2d-Ab^3d+Cb^3d\right)\left(fx+e\right)}{c^2+d^2}+\frac{3\left(Ba^3c+3\,Aa^2bc-3\,Ca^2bc-3\,Bab^2c-Ab^3c\right.}{}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/6*(6*(A*a^3*c - C*a^3*c - 3*B*a^2*b*c - 3*A*a*b^2*c + 3*C*a*b^2*c + B*b^3
*c + B*a^3*d + 3*A*a^2*b*d - 3*C*a^2*b*d - 3*B*a*b^2*d - A*b^3*d + C*b^3*d)
*(f*x + e)/(c^2 + d^2) + 3*(B*a^3*c + 3*A*a^2*b*c - 3*C*a^2*b*c - 3*B*a*b^2
*c - A*b^3*c + C*b^3*c - A*a^3*d + C*a^3*d + 3*B*a^2*b*d + 3*A*a*b^2*d - 3*
C*a*b^2*d - B*b^3*d)*log(tan(f*x + e)^2 + 1)/(c^2 + d^2) - 6*(C*b^3*c^5 - 3
*C*a*b^2*c^4*d - B*b^3*c^4*d + 3*C*a^2*b*c^3*d^2 + 3*B*a*b^2*c^3*d^2 + A*b^
3*c^3*d^2 - C*a^3*c^2*d^3 - 3*B*a^2*b*c^2*d^3 - 3*A*a*b^2*c^2*d^3 + B*a^3*c
*d^4 + 3*A*a^2*b*c*d^4 - A*a^3*d^5)*log(abs(d*tan(f*x + e) + c))/(c^2*d^4 +
 d^6) + (2*C*b^3*d^2*tan(f*x + e)^3 - 3*C*b^3*c*d*tan(f*x + e)^2 + 9*C*a*b^
2*d^2*tan(f*x + e)^2 + 3*B*b^3*d^2*tan(f*x + e)^2 + 6*C*b^3*c^2*tan(f*x + e
) - 18*C*a*b^2*c*d*tan(f*x + e) - 6*B*b^3*c*d*tan(f*x + e) + 18*C*a^2*b*d^2
*tan(f*x + e) + 18*B*a*b^2*d^2*tan(f*x + e) + 6*A*b^3*d^2*tan(f*x + e) - 6*
C*b^3*d^2*tan(f*x + e))/d^3)/f
```

maple [B] time = 0.27, size = 1304, normalized size = 3.87

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x)
```

```
[Out] 1/2/f*b^3/d*B*tan(f*x+e)^2+1/f*b^3/d*A*tan(f*x+e)-1/f*b^3/d*C*tan(f*x+e)-3/
f/d^2/(c^2+d^2)*ln(c+d*tan(f*x+e))*B*a*b^2*c^3+3/f/d/(c^2+d^2)*ln(c+d*tan(f
*x+e))*B*a^2*b*c^2-3/f/d^2/(c^2+d^2)*ln(c+d*tan(f*x+e))*C*a^2*b*c^3+3/f/d^3
/(c^2+d^2)*ln(c+d*tan(f*x+e))*C*c^4*a*b^2+3/f/d/(c^2+d^2)*ln(c+d*tan(f*x+e)
)*A*a*b^2*c^2+1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*a^3*c-3/2/f/(c^2+d^2)*ln
(1+tan(f*x+e)^2)*C*a^2*b*c-3/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*C*a*b^2*d-3/f
/(c^2+d^2)*ln(c+d*tan(f*x+e))*A*a^2*b*c-3/f/(c^2+d^2)*B*arctan(tan(f*x+e))*
a*b^2*d-3/f/(c^2+d^2)*C*arctan(tan(f*x+e))*a^2*b*d-3/2/f/(c^2+d^2)*ln(1+tan
(f*x+e)^2)*B*a*b^2*c+3/f/(c^2+d^2)*C*arctan(tan(f*x+e))*a*b^2*c-3/f*b^2/d^2
*C*a*c*tan(f*x+e)+3/f/(c^2+d^2)*A*arctan(tan(f*x+e))*a^2*b*d+1/f/d^3/(c^2+d
^2)*ln(c+d*tan(f*x+e))*B*c^4*b^3+1/f/d/(c^2+d^2)*ln(c+d*tan(f*x+e))*C*a^3*c
^2-1/f/d^4/(c^2+d^2)*ln(c+d*tan(f*x+e))*C*c^5*b^3+3/2/f/(c^2+d^2)*ln(1+tan(
f*x+e)^2)*A*a^2*b*c+3/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*a*b^2*d-3/f/(c^2+d
^2)*A*arctan(tan(f*x+e))*a*b^2*c+3/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*a^2*b
*d-1/f/d^2/(c^2+d^2)*ln(c+d*tan(f*x+e))*A*b^3*c^3-3/f/(c^2+d^2)*B*arctan(ta
n(f*x+e))*a^2*b*c-1/2/f*b^3/d^2*C*tan(f*x+e)^2*c+1/f*d/(c^2+d^2)*ln(c+d*tan
(f*x+e))*A*a^3+3/f*b^2/d*B*a*tan(f*x+e)-1/f*b^3/d^2*B*c*tan(f*x+e)-1/2/f/(c
^2+d^2)*ln(1+tan(f*x+e)^2)*A*a^3*d-1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*b^3
*c-1/f/(c^2+d^2)*C*arctan(tan(f*x+e))*a^3*c+1/f/(c^2+d^2)*C*arctan(tan(f*x+
e))*b^3*d-1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*b^3*d+1/2/f/(c^2+d^2)*ln(1+t
an(f*x+e)^2)*a^3*C*d+1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*C*b^3*c-1/f/(c^2+d^
2)*ln(c+d*tan(f*x+e))*B*a^3*c+3/f*b/d*a^2*C*tan(f*x+e)+1/f*b^3/d^3*C*c^2*ta
n(f*x+e)+3/2/f*b^2/d*C*tan(f*x+e)^2*a+1/f/(c^2+d^2)*A*arctan(tan(f*x+e))*a^
3*c-1/f/(c^2+d^2)*A*arctan(tan(f*x+e))*b^3*d+1/f/(c^2+d^2)*B*arctan(tan(f*x
+e))*a^3*d+1/f/(c^2+d^2)*B*arctan(tan(f*x+e))*b^3*c+1/3/f*b^3/d*C*tan(f*x+e
)^3
```

maxima [A] time = 0.59, size = 445, normalized size = 1.32

$$\frac{\frac{6\left(\left(\left((A-C)a^3-3\,Ba^2b-3\,(A-C)ab^2+Bb^3\right)c+\left(Ba^3+3\,(A-C)a^2b-3\,Bab^2-(A-C)b^3\right)d\right)\left(fx+e\right)}{c^2+d^2}-\frac{6\left(Cb^3c^5-Aa^3d^5-\left(3\,Cab^2+Bb^3\right)c^4d+\left(3\,Ca^2b+3\,Bab^2+\cdots\right.\right.}{\cdots}}{\cdots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/6*(6*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c + (B*a^3 + 3*
(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d)*(f*x + e)/(c^2 + d^2) - 6*(C*b^
3*c^5 - A*a^3*d^5 - (3*C*a*b^2 + B*b^3)*c^4*d + (3*C*a^2*b + 3*B*a*b^2 + A*
b^3)*c^3*d^2 - (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^3 + (B*a^3 + 3*A*a^2*b
```

```
)*c*d^4)*log(d*tan(f*x + e) + c)/(c^2*d^4 + d^6) + 3*((B*a^3 + 3*(A - C)*a^
2*b - 3*B*a*b^2 - (A - C)*b^3)*c - ((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b
^2 + B*b^3)*d)*log(tan(f*x + e)^2 + 1)/(c^2 + d^2) + (2*C*b^3*d^2*tan(f*x +
 e)^3 - 3*(C*b^3*c*d - (3*C*a*b^2 + B*b^3)*d^2)*tan(f*x + e)^2 + 6*(C*b^3*c
^2 - (3*C*a*b^2 + B*b^3)*c*d + (3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*d^2)*t
an(f*x + e))/d^3)/f
```

mupad [B] time = 13.39, size = 508, normalized size = 1.51

$$\frac{\tan\left(e+f\,x\right)^2\,\left(\frac{B\,b^3+3\,C\,a\,b^2}{2\,d}-\frac{C\,b^3\,c}{2\,d^2}\right)}{f}-\frac{\tan\left(e+f\,x\right)\,\left(\frac{c\,\left(\frac{B\,b^3+3\,C\,a\,b^2}{d}-\frac{C\,b^3\,c}{d^2}\right)}{d}-\frac{3\,C\,a^2\,b+3\,B\,a\,b^2+A\,b^3}{d}+\frac{C\,b^3}{d}\right)}{f}-\frac{\ln\left(c+d\tan\left(e\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x)),x)
```

```
[Out] (tan(e + f*x)^2*((B*b^3 + 3*C*a*b^2)/(2*d) - (C*b^3*c)/(2*d^2)))/f - (tan(e
 + f*x)*((c*((B*b^3 + 3*C*a*b^2)/d - (C*b^3*c)/d^2))/d - (A*b^3 + 3*B*a*b^2
 + 3*C*a^2*b)/d + (C*b^3)/d))/f - (log(c + d*tan(e + f*x))*(d^4*(B*a^3*c +
3*A*a^2*b*c) - d^3*(C*a^3*c^2 + 3*A*a*b^2*c^2 + 3*B*a^2*b*c^2) + d^2*(A*b^3
*c^3 + 3*B*a*b^2*c^3 + 3*C*a^2*b*c^3) - d*(B*b^3*c^4 + 3*C*a*b^2*c^4) - A*a
^3*d^5 + C*b^3*c^5))/(f*(d^6 + c^2*d^4)) - (log(tan(e + f*x) + 1i)*(A*a^3 +
 A*b^3*1i - B*a^3*1i + B*b^3 - C*a^3 - C*b^3*1i - 3*A*a*b^2 - A*a^2*b*3i +
B*a*b^2*3i - 3*B*a^2*b + 3*C*a*b^2 + C*a^2*b*3i))/(2*f*(c*1i + d)) - (log(t
an(e + f*x) - 1i)*(A*a^3*1i + A*b^3 - B*a^3 + B*b^3*1i - C*a^3*1i - C*b^3 -
 A*a*b^2*3i - 3*A*a^2*b + 3*B*a*b^2 - B*a^2*b*3i + C*a*b^2*3i + 3*C*a^2*b))
/(2*f*(c + d*1i)) + (C*b^3*tan(e + f*x)^3)/(3*d*f)
```

sympy [A] time = 115.15, size = 7205, normalized size = 21.38

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e)),x)
```

```
[Out] Piecewise((zoo*x*(a + b*tan(e))**3*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(
c, 0) & Eq(d, 0) & Eq(f, 0)), (-3*I*A*a**3*f*x*tan(e + f*x)/(-6*d*f*tan(e +
 f*x) + 6*I*d*f) - 3*A*a**3*f*x/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 3*I*A*a**
3/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 9*A*a**2*b*f*x*tan(e + f*x)/(-6*d*f*tan
(e + f*x) + 6*I*d*f) + 9*I*A*a**2*b*f*x/(-6*d*f*tan(e + f*x) + 6*I*d*f) + 9
*A*a**2*b/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 9*I*A*a*b**2*f*x*tan(e + f*x)/(
```

```
-6*d*f*tan(e + f*x) + 6*I*d*f) - 9*A*a*b**2*f*x/(-6*d*f*tan(e + f*x) + 6*I*
d*f) - 9*A*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x
) + 6*I*d*f) + 9*I*A*a*b**2*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) +
 6*I*d*f) + 9*I*A*a*b**2/(-6*d*f*tan(e + f*x) + 6*I*d*f) + 9*A*b**3*f*x*tan
(e + f*x)/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 9*I*A*b**3*f*x/(-6*d*f*tan(e +
f*x) + 6*I*d*f) - 3*I*A*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*
tan(e + f*x) + 6*I*d*f) - 3*A*b**3*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e +
 f*x) + 6*I*d*f) - 6*A*b**3*tan(e + f*x)**2/(-6*d*f*tan(e + f*x) + 6*I*d*f)
 - 9*A*b**3/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 3*B*a**3*f*x*tan(e + f*x)/(-6
*d*f*tan(e + f*x) + 6*I*d*f) + 3*I*B*a**3*f*x/(-6*d*f*tan(e + f*x) + 6*I*d*
f) + 3*B*a**3/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 9*I*B*a**2*b*f*x*tan(e + f*
x)/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 9*B*a**2*b*f*x/(-6*d*f*tan(e + f*x) +
6*I*d*f) - 9*B*a**2*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e +
 f*x) + 6*I*d*f) + 9*I*B*a**2*b*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*
x) + 6*I*d*f) + 9*I*B*a**2*b/(-6*d*f*tan(e + f*x) + 6*I*d*f) + 27*B*a*b**2*
f*x*tan(e + f*x)/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 27*I*B*a*b**2*f*x/(-6*d*
f*tan(e + f*x) + 6*I*d*f) - 9*I*B*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f
*x)/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 9*B*a*b**2*log(tan(e + f*x)**2 + 1)/(
-6*d*f*tan(e + f*x) + 6*I*d*f) - 18*B*a*b**2*tan(e + f*x)**2/(-6*d*f*tan(e
+ f*x) + 6*I*d*f) - 27*B*a*b**2/(-6*d*f*tan(e + f*x) + 6*I*d*f) + 9*I*B*b**
3*f*x*tan(e + f*x)/(-6*d*f*tan(e + f*x) + 6*I*d*f) + 9*B*b**3*f*x/(-6*d*f*t
an(e + f*x) + 6*I*d*f) + 6*B*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6
*d*f*tan(e + f*x) + 6*I*d*f) - 6*I*B*b**3*log(tan(e + f*x)**2 + 1)/(-6*d*f*
tan(e + f*x) + 6*I*d*f) - 3*B*b**3*tan(e + f*x)**3/(-6*d*f*tan(e + f*x) + 6
*I*d*f) - 3*I*B*b**3*tan(e + f*x)**2/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 9*I*
B*b**3/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 3*I*C*a**3*f*x*tan(e + f*x)/(-6*d*
f*tan(e + f*x) + 6*I*d*f) - 3*C*a**3*f*x/(-6*d*f*tan(e + f*x) + 6*I*d*f) -
3*C*a**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x) + 6*I*d
*f) + 3*I*C*a**3*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) + 6*I*d*f) +
 3*I*C*a**3/(-6*d*f*tan(e + f*x) + 6*I*d*f) + 27*C*a**2*b*f*x*tan(e + f*x)/
(-6*d*f*tan(e + f*x) + 6*I*d*f) - 27*I*C*a**2*b*f*x/(-6*d*f*tan(e + f*x) +
6*I*d*f) - 9*I*C*a**2*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e
 + f*x) + 6*I*d*f) - 9*C*a**2*b*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*
x) + 6*I*d*f) - 18*C*a**2*b*tan(e + f*x)**2/(-6*d*f*tan(e + f*x) + 6*I*d*f)
 - 27*C*a**2*b/(-6*d*f*tan(e + f*x) + 6*I*d*f) + 27*I*C*a*b**2*f*x*tan(e +
f*x)/(-6*d*f*tan(e + f*x) + 6*I*d*f) + 27*C*a*b**2*f*x/(-6*d*f*tan(e + f*x)
 + 6*I*d*f) + 18*C*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan
(e + f*x) + 6*I*d*f) - 18*I*C*a*b**2*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e
 + f*x) + 6*I*d*f) - 9*C*a*b**2*tan(e + f*x)**3/(-6*d*f*tan(e + f*x) + 6*I*
d*f) - 9*I*C*a*b**2*tan(e + f*x)**2/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 27*I*
C*a*b**2/(-6*d*f*tan(e + f*x) + 6*I*d*f) - 15*C*b**3*f*x*tan(e + f*x)/(-6*d
*f*tan(e + f*x) + 6*I*d*f) + 15*I*C*b**3*f*x/(-6*d*f*tan(e + f*x) + 6*I*d*f
) + 6*I*C*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x) +
 6*I*d*f) + 6*C*b**3*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) + 6*I*d*
f) - 2*C*b**3*tan(e + f*x)**4/(-6*d*f*tan(e + f*x) + 6*I*d*f) - I*C*b**3*ta
```

```
n(e + f*x)**3/(-6*d*f*tan(e + f*x) + 6*I*d*f) + 9*C*b**3*tan(e + f*x)**2/(-
6*d*f*tan(e + f*x) + 6*I*d*f) + 15*C*b**3/(-6*d*f*tan(e + f*x) + 6*I*d*f),
Eq(c, -I*d)), (3*I*A*a**3*f*x*tan(e + f*x)/(-6*d*f*tan(e + f*x) - 6*I*d*f)
- 3*A*a**3*f*x/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 3*I*A*a**3/(-6*d*f*tan(e +
 f*x) - 6*I*d*f) - 9*A*a**2*b*f*x*tan(e + f*x)/(-6*d*f*tan(e + f*x) - 6*I*d
*f) - 9*I*A*a**2*b*f*x/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 9*A*a**2*b/(-6*d*f
*tan(e + f*x) - 6*I*d*f) + 9*I*A*a*b**2*f*x*tan(e + f*x)/(-6*d*f*tan(e + f*
x) - 6*I*d*f) - 9*A*a*b**2*f*x/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 9*A*a*b**2
*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 9*
I*A*a*b**2*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 9*I*A
*a*b**2/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 9*A*b**3*f*x*tan(e + f*x)/(-6*d*f
*tan(e + f*x) - 6*I*d*f) + 9*I*A*b**3*f*x/(-6*d*f*tan(e + f*x) - 6*I*d*f) +
 3*I*A*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x) - 6*
I*d*f) - 3*A*b**3*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) - 6*I*d*f)
- 6*A*b**3*tan(e + f*x)**2/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 9*A*b**3/(-6*d
*f*tan(e + f*x) - 6*I*d*f) - 3*B*a**3*f*x*tan(e + f*x)/(-6*d*f*tan(e + f*x)
 - 6*I*d*f) - 3*I*B*a**3*f*x/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 3*B*a**3/(-6
*d*f*tan(e + f*x) - 6*I*d*f) + 9*I*B*a**2*b*f*x*tan(e + f*x)/(-6*d*f*tan(e
+ f*x) - 6*I*d*f) - 9*B*a**2*b*f*x/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 9*B*a*
*2*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x) - 6*I*d*f)
- 9*I*B*a**2*b*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 9
*I*B*a**2*b/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 27*B*a*b**2*f*x*tan(e + f*x)/
(-6*d*f*tan(e + f*x) - 6*I*d*f) + 27*I*B*a*b**2*f*x/(-6*d*f*tan(e + f*x) -
6*I*d*f) + 9*I*B*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e
 + f*x) - 6*I*d*f) - 9*B*a*b**2*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*
x) - 6*I*d*f) - 18*B*a*b**2*tan(e + f*x)**2/(-6*d*f*tan(e + f*x) - 6*I*d*f)
 - 27*B*a*b**2/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 9*I*B*b**3*f*x*tan(e + f*x
)/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 9*B*b**3*f*x/(-6*d*f*tan(e + f*x) - 6*I
*d*f) + 6*B*b**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x)
 - 6*I*d*f) + 6*I*B*b**3*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) - 6*
I*d*f) - 3*B*b**3*tan(e + f*x)**3/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 3*I*B*b
**3*tan(e + f*x)**2/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 9*I*B*b**3/(-6*d*f*ta
n(e + f*x) - 6*I*d*f) + 3*I*C*a**3*f*x*tan(e + f*x)/(-6*d*f*tan(e + f*x) -
6*I*d*f) - 3*C*a**3*f*x/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 3*C*a**3*log(tan(
e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 3*I*C*a**3*
log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 3*I*C*a**3/(-6*d
*f*tan(e + f*x) - 6*I*d*f) + 27*C*a**2*b*f*x*tan(e + f*x)/(-6*d*f*tan(e + f
*x) - 6*I*d*f) + 27*I*C*a**2*b*f*x/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 9*I*C*
a**2*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x) - 6*I*d*f
) - 9*C*a**2*b*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 1
8*C*a**2*b*tan(e + f*x)**2/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 27*C*a**2*b/(-
6*d*f*tan(e + f*x) - 6*I*d*f) - 27*I*C*a*b**2*f*x*tan(e + f*x)/(-6*d*f*tan(
e + f*x) - 6*I*d*f) + 27*C*a*b**2*f*x/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 18*
C*a*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x) - 6*I*d
*f) + 18*I*C*a*b**2*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) - 6*I*d*f
```

```
) - 9*C*a*b**2*tan(e + f*x)**3/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 9*I*C*a*b*
*2*tan(e + f*x)**2/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 27*I*C*a*b**2/(-6*d*f*
tan(e + f*x) - 6*I*d*f) - 15*C*b**3*f*x*tan(e + f*x)/(-6*d*f*tan(e + f*x) -
 6*I*d*f) - 15*I*C*b**3*f*x/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 6*I*C*b**3*lo
g(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-6*d*f*tan(e + f*x) - 6*I*d*f) + 6*C*b
**3*log(tan(e + f*x)**2 + 1)/(-6*d*f*tan(e + f*x) - 6*I*d*f) - 2*C*b**3*tan
(e + f*x)**4/(-6*d*f*tan(e + f*x) - 6*I*d*f) + I*C*b**3*tan(e + f*x)**3/(-6
*d*f*tan(e + f*x) - 6*I*d*f) + 9*C*b**3*tan(e + f*x)**2/(-6*d*f*tan(e + f*x
) - 6*I*d*f) + 15*C*b**3/(-6*d*f*tan(e + f*x) - 6*I*d*f), Eq(c, I*d)), ((A*
a**3*x + 3*A*a**2*b*log(tan(e + f*x)**2 + 1)/(2*f) - 3*A*a*b**2*x + 3*A*a*b
**2*tan(e + f*x)/f - A*b**3*log(tan(e + f*x)**2 + 1)/(2*f) + A*b**3*tan(e +
 f*x)**2/(2*f) + B*a**3*log(tan(e + f*x)**2 + 1)/(2*f) - 3*B*a**2*b*x + 3*B
*a**2*b*tan(e + f*x)/f - 3*B*a*b**2*log(tan(e + f*x)**2 + 1)/(2*f) + 3*B*a*
b**2*tan(e + f*x)**2/(2*f) + B*b**3*x + B*b**3*tan(e + f*x)**3/(3*f) - B*b*
*3*tan(e + f*x)/f - C*a**3*x + C*a**3*tan(e + f*x)/f - 3*C*a**2*b*log(tan(e
 + f*x)**2 + 1)/(2*f) + 3*C*a**2*b*tan(e + f*x)**2/(2*f) + 3*C*a*b**2*x + C
*a*b**2*tan(e + f*x)**3/f - 3*C*a*b**2*tan(e + f*x)/f + C*b**3*log(tan(e +
f*x)**2 + 1)/(2*f) + C*b**3*tan(e + f*x)**4/(4*f) - C*b**3*tan(e + f*x)**2/
(2*f))/c, Eq(d, 0)), (x*(a + b*tan(e))**3*(A + B*tan(e) + C*tan(e)**2)/(c +
 d*tan(e)), Eq(f, 0)), (6*A*a**3*c*d**4*f*x/(6*c**2*d**4*f + 6*d**6*f) + 6*
A*a**3*d**5*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) - 3*A*a**3*d
**5*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 18*A*a**2*b*c*d**
4*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) + 9*A*a**2*b*c*d**4*lo
g(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) + 18*A*a**2*b*d**5*f*x/(6
*c**2*d**4*f + 6*d**6*f) + 18*A*a*b**2*c**2*d**3*log(c/d + tan(e + f*x))/(6
*c**2*d**4*f + 6*d**6*f) - 18*A*a*b**2*c*d**4*f*x/(6*c**2*d**4*f + 6*d**6*f
) + 9*A*a*b**2*d**5*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 6
*A*b**3*c**3*d**2*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) + 6*A*
b**3*c**2*d**3*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*f) - 3*A*b**3*c*d**4*lo
g(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 6*A*b**3*d**5*f*x/(6*c*
*2*d**4*f + 6*d**6*f) + 6*A*b**3*d**5*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*
f) - 6*B*a**3*c*d**4*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) + 3
*B*a**3*c*d**4*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) + 6*B*a*
*3*d**5*f*x/(6*c**2*d**4*f + 6*d**6*f) + 18*B*a**2*b*c**2*d**3*log(c/d + ta
n(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) - 18*B*a**2*b*c*d**4*f*x/(6*c**2*d**
4*f + 6*d**6*f) + 9*B*a**2*b*d**5*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f +
 6*d**6*f) - 18*B*a*b**2*c**3*d**2*log(c/d + tan(e + f*x))/(6*c**2*d**4*f +
 6*d**6*f) + 18*B*a*b**2*c**2*d**3*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*f)
- 9*B*a*b**2*c*d**4*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 1
8*B*a*b**2*d**5*f*x/(6*c**2*d**4*f + 6*d**6*f) + 18*B*a*b**2*d**5*tan(e + f
*x)/(6*c**2*d**4*f + 6*d**6*f) + 6*B*b**3*c**4*d*log(c/d + tan(e + f*x))/(6
*c**2*d**4*f + 6*d**6*f) - 6*B*b**3*c**3*d**2*tan(e + f*x)/(6*c**2*d**4*f +
 6*d**6*f) + 3*B*b**3*c**2*d**3*tan(e + f*x)**2/(6*c**2*d**4*f + 6*d**6*f)
+ 6*B*b**3*c*d**4*f*x/(6*c**2*d**4*f + 6*d**6*f) - 6*B*b**3*c*d**4*tan(e +
f*x)/(6*c**2*d**4*f + 6*d**6*f) - 3*B*b**3*d**5*log(tan(e + f*x)**2 + 1)/(6
```

```
*c**2*d**4*f + 6*d**6*f) + 3*B*b**3*d**5*tan(e + f*x)**2/(6*c**2*d**4*f + 6
*d**6*f) + 6*C*a**3*c**2*d**3*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d*
*6*f) - 6*C*a**3*c*d**4*f*x/(6*c**2*d**4*f + 6*d**6*f) + 3*C*a**3*d**5*log(
tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 18*C*a**2*b*c**3*d**2*log
(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) + 18*C*a**2*b*c**2*d**3*tan
(e + f*x)/(6*c**2*d**4*f + 6*d**6*f) - 9*C*a**2*b*c*d**4*log(tan(e + f*x)**
2 + 1)/(6*c**2*d**4*f + 6*d**6*f) - 18*C*a**2*b*d**5*f*x/(6*c**2*d**4*f + 6
*d**6*f) + 18*C*a**2*b*d**5*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*f) + 18*C*
a*b**2*c**4*d*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) - 18*C*a*b
**2*c**3*d**2*tan(e + f*x)/(6*c**2*d**4*f + 6*d**6*f) + 9*C*a*b**2*c**2*d**
3*tan(e + f*x)**2/(6*c**2*d**4*f + 6*d**6*f) + 18*C*a*b**2*c*d**4*f*x/(6*c*
*2*d**4*f + 6*d**6*f) - 18*C*a*b**2*c*d**4*tan(e + f*x)/(6*c**2*d**4*f + 6*
d**6*f) - 9*C*a*b**2*d**5*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f + 6*d**6*
f) + 9*C*a*b**2*d**5*tan(e + f*x)**2/(6*c**2*d**4*f + 6*d**6*f) - 6*C*b**3*
c**5*log(c/d + tan(e + f*x))/(6*c**2*d**4*f + 6*d**6*f) + 6*C*b**3*c**4*d*t
an(e + f*x)/(6*c**2*d**4*f + 6*d**6*f) - 3*C*b**3*c**3*d**2*tan(e + f*x)**2
/(6*c**2*d**4*f + 6*d**6*f) + 2*C*b**3*c**2*d**3*tan(e + f*x)**3/(6*c**2*d*
*4*f + 6*d**6*f) + 3*C*b**3*c*d**4*log(tan(e + f*x)**2 + 1)/(6*c**2*d**4*f
+ 6*d**6*f) - 3*C*b**3*c*d**4*tan(e + f*x)**2/(6*c**2*d**4*f + 6*d**6*f) +
6*C*b**3*d**5*f*x/(6*c**2*d**4*f + 6*d**6*f) + 2*C*b**3*d**5*tan(e + f*x)**
3/(6*c**2*d**4*f + 6*d**6*f) - 6*C*b**3*d**5*tan(e + f*x)/(6*c**2*d**4*f +
6*d**6*f), True))
```

3.71 $\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$

Optimal. Leaf size=236

$$-\frac{\log(\cos(e+fx))\left(a^2(Bc-d(A-C))+2ab(Ac+Bd-cC)-b^2(Bc-d(A-C))\right)}{f\left(c^2+d^2\right)}+\frac{x\left(a^2(Ac+Bd-cC)-2ab(\right.}{}$$

```
[Out] (a^2*(A*c+B*d-C*c)-b^2*(A*c+B*d-C*c)-2*a*b*(B*c-(A-C)*d))*x/(c^2+d^2)-(2*a*
b*(A*c+B*d-C*c)+a^2*(B*c-(A-C)*d)-b^2*(B*c-(A-C)*d))*ln(cos(f*x+e))/(c^2+d^
2)/f+(-a*d+b*c)^2*(A*d^2-B*c*d+C*c^2)*ln(c+d*tan(f*x+e))/d^3/(c^2+d^2)/f-b*
(-B*b*d-C*a*d+C*b*c)*tan(f*x+e)/d^2/f+1/2*C*(a+b*tan(f*x+e))^2/d/f
```

Rubi [A] time = 0.80, antiderivative size = 236, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3647, 3637, 3626, 3617, 31, 3475}

$$-\frac{\log(\cos(e+fx))\left(a^2(Bc-d(A-C))+2ab(Ac+Bd-cC)-b^2(Bc-d(A-C))\right)}{f\left(c^2+d^2\right)}+\frac{x\left(a^2(Ac+Bd-cC)-2ab(\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x]),x]
```

```
[Out] ((a^2*(A*c - c*C + B*d) - b^2*(A*c - c*C + B*d) - 2*a*b*(B*c - (A - C)*d))*
x)/(c^2 + d^2) - ((2*a*b*(A*c - c*C + B*d) + a^2*(B*c - (A - C)*d) - b^2*(B
*c - (A - C)*d))*Log[Cos[e + f*x]])/((c^2 + d^2)*f) + ((b*c - a*d)^2*(c^2*C
 - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d^3*(c^2 + d^2)*f) - (b*(b*c*C
- b*B*d - a*C*d)*Tan[e + f*x])/(d^2*f) + (C*(a + b*Tan[e + f*x])^2)/(2*d*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
```

```
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx = \frac{C(a+b\tan(e+fx))^2}{2df} + \frac{\int \frac{(a+b\tan(e+fx))\left(-2(b\right.}{}}{}$$

$$= -\frac{b(bcC-bBd-aCd)\tan(e+fx)}{d^2f} + \frac{C(a+b}{}$$

$$= \frac{\left(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2a\right.}{c^2+d^2}$$

$$= \frac{\left(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2a\right.}{c^2+d^2}$$

$$= \frac{\left(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)-2a\right.}{c^2+d^2}$$

Mathematica [C] time = 2.99, size = 190, normalized size = 0.81

$$\frac{\frac{2(bc-ad)^2\left(Ad^2-Bcd+c^2C\right)\log(c+d\tan(e+fx))}{d^2\left(c^2+d^2\right)} + \frac{d(a-ib)^2(iA+B-iC)\log(\tan(e+fx)+i)}{c-id} + \frac{d(a+ib)^2(-iA+B+iC)\log(-\tan(e+fx)+i)}{c+id} + \frac{2b\tan(e+}{}}{2df}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x]),x]
```

```
[Out] ((((a + I*b)^2*((-I)*A + B + I*C)*d*Log[I - Tan[e + f*x]])/(c + I*d) + ((a -
 I*b)^2*(I*A + B - I*C)*d*Log[I + Tan[e + f*x]])/(c - I*d) + (2*(b*c - a*d)
^2*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d^2*(c^2 + d^2)) + (2*
b*(-(b*c*C) + b*B*d + a*C*d)*Tan[e + f*x])/d + C*(a + b*Tan[e + f*x])^2)/(2
*d*f)
```

fricas [A] time = 1.17, size = 390, normalized size = 1.65

$$\frac{2\left(\left((A-C)a^2-2\,Bab-(A-C)b^2\right)cd^3+\left(Ba^2+2\,(A-C)ab-Bb^2\right)d^4\right)fx+\left(Cb^2c^2d^2+Cb^2d^4\right)\tan\left(fx+e\right)^2}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^3 + (B*a^2 + 2*(A - C)*a*
b - B*b^2)*d^4)*f*x + (C*b^2*c^2*d^2 + C*b^2*d^4)*tan(f*x + e)^2 + (C*b^2*c
^4 + A*a^2*d^4 - (2*C*a*b + B*b^2)*c^3*d + (C*a^2 + 2*B*a*b + A*b^2)*c^2*d^
2 - (B*a^2 + 2*A*a*b)*c*d^3)*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) +
 c^2)/(tan(f*x + e)^2 + 1)) - (C*b^2*c^4 - (2*C*a*b + B*b^2)*c^3*d + (C*a^2
 + 2*B*a*b + A*b^2)*c^2*d^2 - (2*C*a*b + B*b^2)*c*d^3 + (C*a^2 + 2*B*a*b +
(A - C)*b^2)*d^4)*log(1/(tan(f*x + e)^2 + 1)) - 2*(C*b^2*c^3*d + C*b^2*c*d^
3 - (2*C*a*b + B*b^2)*c^2*d^2 - (2*C*a*b + B*b^2)*d^4)*tan(f*x + e))/((c^2*
d^3 + d^5)*f)
```

giac [A] time = 3.06, size = 338, normalized size = 1.43

$$\frac{\frac{2\left(Aa^2c-Ca^2c-2\,Babc-Ab^2c+Cb^2c+Ba^2d+2\,Aabd-2\,Cabd-Bb^2d\right)\left(fx+e\right)}{c^2+d^2}+\frac{\left(Ba^2c+2\,Aabc-2\,Cabc-Bb^2c-Aa^2d+Ca^2d+2\,Babd+Ab^2d-Cb^2d\right)\log\left(\tan\right.}{c^2+d^2}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c - C*a^2*c - 2*B*a*b*c - A*b^2*c + C*b^2*c + B*a^2*d + 2*A*a
*b*d - 2*C*a*b*d - B*b^2*d)*(f*x + e)/(c^2 + d^2) + (B*a^2*c + 2*A*a*b*c -
2*C*a*b*c - B*b^2*c - A*a^2*d + C*a^2*d + 2*B*a*b*d + A*b^2*d - C*b^2*d)*lo
g(tan(f*x + e)^2 + 1)/(c^2 + d^2) + 2*(C*b^2*c^4 - 2*C*a*b*c^3*d - B*b^2*c^
3*d + C*a^2*c^2*d^2 + 2*B*a*b*c^2*d^2 + A*b^2*c^2*d^2 - B*a^2*c*d^3 - 2*A*a
*b*c*d^3 + A*a^2*d^4)*log(abs(d*tan(f*x + e) + c))/(c^2*d^3 + d^5) + (C*b^2
*d*tan(f*x + e)^2 - 2*C*b^2*c*tan(f*x + e) + 4*C*a*b*d*tan(f*x + e) + 2*B*b
^2*d*tan(f*x + e))/d^2)/f
```

maple [B] time = 0.29, size = 861, normalized size = 3.65

$$-\frac{b^2Cc\tan\left(fx+e\right)}{f\,d^2}-\frac{2C\arctan\left(\tan\left(fx+e\right)\right)abd}{f\left(c^2+d^2\right)}+\frac{\ln\left(c+d\tan\left(fx+e\right)\right)A\,c^2b^2}{fd\left(c^2+d^2\right)}+\frac{A\arctan\left(\tan\left(fx+e\right)\right)a^2c}{f\left(c^2+d^2\right)}+$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x)
```

```
[Out] -1/f*b^2/d^2*C*c*tan(f*x+e)-1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*C*b^2*d-1/2/
f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*a^2*d-1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B
*b^2*c+1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*C*a^2*d-2/f/(c^2+d^2)*C*arctan(ta
n(f*x+e))*a*b*d+1/f/d/(c^2+d^2)*ln(c+d*tan(f*x+e))*A*c^2*b^2+1/f/(c^2+d^2)*
```

```
A*arctan(tan(f*x+e))*a^2*c+1/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*a*b*c+1/f/(c^
2+d^2)*ln(1+tan(f*x+e)^2)*B*a*b*d+2/f/(c^2+d^2)*A*arctan(tan(f*x+e))*a*b*d-
2/f/(c^2+d^2)*B*arctan(tan(f*x+e))*a*b*c+2/f/d/(c^2+d^2)*ln(c+d*tan(f*x+e))
*B*c^2*a*b-2/f/d^2/(c^2+d^2)*ln(c+d*tan(f*x+e))*C*c^3*a*b-1/f/d^2/(c^2+d^2)
*ln(c+d*tan(f*x+e))*B*c^3*b^2+1/f/d/(c^2+d^2)*ln(c+d*tan(f*x+e))*C*c^2*a^2+
1/f/d^3/(c^2+d^2)*ln(c+d*tan(f*x+e))*C*c^4*b^2-1/f/(c^2+d^2)*ln(1+tan(f*x+e
)^2)*C*a*b*c-2/f/(c^2+d^2)*ln(c+d*tan(f*x+e))*A*a*c*b+1/f*b^2/d*B*tan(f*x+e
)+1/f/(c^2+d^2)*C*arctan(tan(f*x+e))*b^2*c-1/f/(c^2+d^2)*ln(c+d*tan(f*x+e))
*B*a^2*c-1/f/(c^2+d^2)*B*arctan(tan(f*x+e))*b^2*d-1/f/(c^2+d^2)*C*arctan(ta
n(f*x+e))*a^2*c+1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*b^2*d+1/2/f/(c^2+d^2)*
ln(1+tan(f*x+e)^2)*B*a^2*c+1/2/f*b^2/d*C*tan(f*x+e)^2+2/f*b/d*C*tan(f*x+e)*
a-1/f/(c^2+d^2)*A*arctan(tan(f*x+e))*b^2*c+1/f*d/(c^2+d^2)*ln(c+d*tan(f*x+e
))*A*a^2+1/f/(c^2+d^2)*B*arctan(tan(f*x+e))*a^2*d
```

maxima [A] time = 0.55, size = 294, normalized size = 1.25

$$\frac{\frac{2\left(\left(\left((A-C)a^2-2\,Bab-(A-C)b^2\right)c+\left(Ba^2+2\,(A-C)ab-Bb^2\right)d\right)(fx+e)}{c^2+d^2}+\frac{2\left(Cb^2c^4+Aa^2d^4-\left(2\,Cab+Bb^2\right)c^3d+\left(Ca^2+2\,Bab+Ab^2\right)c^2d^2-\left(Ba^2+2\,Aab\right)cd^3\right)}{c^2d^3+d^5}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c + (B*a^2 + 2*(A - C)*a*b -
B*b^2)*d)*(f*x + e)/(c^2 + d^2) + 2*(C*b^2*c^4 + A*a^2*d^4 - (2*C*a*b + B*b
^2)*c^3*d + (C*a^2 + 2*B*a*b + A*b^2)*c^2*d^2 - (B*a^2 + 2*A*a*b)*c*d^3)*lo
g(d*tan(f*x + e) + c)/(c^2*d^3 + d^5) + ((B*a^2 + 2*(A - C)*a*b - B*b^2)*c
- ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d)*log(tan(f*x + e)^2 + 1)/(c^2 + d
^2) + (C*b^2*d*tan(f*x + e)^2 - 2*(C*b^2*c - (2*C*a*b + B*b^2)*d)*tan(f*x +
 e))/d^2)/f
```

mupad [B] time = 11.20, size = 325, normalized size = 1.38

$$\frac{\tan\left(e+f\,x\right)\,\left(\frac{B\,b^2+2\,C\,a\,b}{d}-\frac{C\,b^2\,c}{d^2}\right)}{f}+\frac{\ln\left(c+d\tan\left(e+f\,x\right)\right)\,\left(d^2\,\left(C\,a^2\,c^2+2\,B\,a\,b\,c^2+A\,b^2\,c^2\right)-d\,\left(B\,b^2\,c^3+2\,C\right.\right.}{f\,\left(c^2\,d^3+d^5\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x)),x)
```

```
[Out] (tan(e + f*x)*((B*b^2 + 2*C*a*b)/d - (C*b^2*c)/d^2))/f + (log(c + d*tan(e +
 f*x))*(d^2*(A*b^2*c^2 + C*a^2*c^2 + 2*B*a*b*c^2) - d*(B*b^2*c^3 + 2*C*a*b*
```

```
c^3) - d^3*(B*a^2*c + 2*A*a*b*c) + A*a^2*d^4 + C*b^2*c^4))/(f*(d^5 + c^2*d^
3)) + (log(tan(e + f*x) + 1i)*(A*b^2 - A*a^2 + B*a^2*1i - B*b^2*1i + C*a^2
- C*b^2 + A*a*b*2i + 2*B*a*b - C*a*b*2i))/(2*f*(c*1i + d)) + (log(tan(e + f
*x) - 1i)*(A*b^2*1i - A*a^2*1i + B*a^2 - B*b^2 + C*a^2*1i - C*b^2*1i + 2*A*
a*b + B*a*b*2i - 2*C*a*b))/(2*f*(c + d*1i)) + (C*b^2*tan(e + f*x)^2)/(2*d*f
)
```

sympy [A] time = 8.13, size = 4517, normalized size = 19.14

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e)),x)
```

```
[Out] Piecewise((zoo*x*(a + b*tan(e))**2*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(
c, 0) & Eq(d, 0) & Eq(f, 0)), (-I*A*a**2*f*x*tan(e + f*x)/(-2*d*f*tan(e + f
*x) + 2*I*d*f) - A*a**2*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) - I*A*a**2/(-2*
d*f*tan(e + f*x) + 2*I*d*f) - 2*A*a*b*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x)
 + 2*I*d*f) + 2*I*A*a*b*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) + 2*A*a*b/(-2*d
*f*tan(e + f*x) + 2*I*d*f) - I*A*b**2*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x)
 + 2*I*d*f) - A*b**2*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) - A*b**2*log(tan(e
 + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + I*A*b**2*log
(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + I*A*b**2/(-2*d*f*ta
n(e + f*x) + 2*I*d*f) - B*a**2*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*
d*f) + I*B*a**2*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) + B*a**2/(-2*d*f*tan(e
+ f*x) + 2*I*d*f) - 2*I*B*a*b*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d
*f) - 2*B*a*b*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) - 2*B*a*b*log(tan(e + f*x
)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + 2*I*B*a*b*log(tan(
e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + 2*I*B*a*b/(-2*d*f*tan(e
+ f*x) + 2*I*d*f) + 3*B*b**2*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*
f) - 3*I*B*b**2*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) - I*B*b**2*log(tan(e +
f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) - B*b**2*log(tan(
e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x) + 2*I*d*f) - 2*B*b**2*tan(e + f*x)**2
/(-2*d*f*tan(e + f*x) + 2*I*d*f) - 3*B*b**2/(-2*d*f*tan(e + f*x) + 2*I*d*f)
 - I*C*a**2*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) - C*a**2*f*x/(
-2*d*f*tan(e + f*x) + 2*I*d*f) - C*a**2*log(tan(e + f*x)**2 + 1)*tan(e + f*
x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + I*C*a**2*log(tan(e + f*x)**2 + 1)/(-2*
d*f*tan(e + f*x) + 2*I*d*f) + I*C*a**2/(-2*d*f*tan(e + f*x) + 2*I*d*f) + 6*
C*a*b*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) - 6*I*C*a*b*f*x/(-2*
d*f*tan(e + f*x) + 2*I*d*f) - 2*I*C*a*b*log(tan(e + f*x)**2 + 1)*tan(e + f*
x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) - 2*C*a*b*log(tan(e + f*x)**2 + 1)/(-2*d
*f*tan(e + f*x) + 2*I*d*f) - 4*C*a*b*tan(e + f*x)**2/(-2*d*f*tan(e + f*x) +
 2*I*d*f) - 6*C*a*b/(-2*d*f*tan(e + f*x) + 2*I*d*f) + 3*I*C*b**2*f*x*tan(e
+ f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + 3*C*b**2*f*x/(-2*d*f*tan(e + f*x)
```

```
+ 2*I*d*f) + 2*C*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e +
 f*x) + 2*I*d*f) - 2*I*C*b**2*log(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x)
 + 2*I*d*f) - C*b**2*tan(e + f*x)**3/(-2*d*f*tan(e + f*x) + 2*I*d*f) - I*C*
b**2*tan(e + f*x)**2/(-2*d*f*tan(e + f*x) + 2*I*d*f) - 3*I*C*b**2/(-2*d*f*t
an(e + f*x) + 2*I*d*f), Eq(c, -I*d)), (I*A*a**2*f*x*tan(e + f*x)/(-2*d*f*ta
n(e + f*x) - 2*I*d*f) - A*a**2*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) + I*A*a*
*2/(-2*d*f*tan(e + f*x) - 2*I*d*f) - 2*A*a*b*f*x*tan(e + f*x)/(-2*d*f*tan(e
 + f*x) - 2*I*d*f) - 2*I*A*a*b*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) + 2*A*a*
b/(-2*d*f*tan(e + f*x) - 2*I*d*f) + I*A*b**2*f*x*tan(e + f*x)/(-2*d*f*tan(e
 + f*x) - 2*I*d*f) - A*b**2*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) - A*b**2*lo
g(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - I*A*b
**2*log(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - I*A*b**2/(-2
*d*f*tan(e + f*x) - 2*I*d*f) - B*a**2*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x)
 - 2*I*d*f) - I*B*a**2*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) + B*a**2/(-2*d*f
*tan(e + f*x) - 2*I*d*f) + 2*I*B*a*b*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x)
- 2*I*d*f) - 2*B*a*b*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) - 2*B*a*b*log(tan(
e + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - 2*I*B*a*b*l
og(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - 2*I*B*a*b/(-2*d*f
*tan(e + f*x) - 2*I*d*f) + 3*B*b**2*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) -
 2*I*d*f) + 3*I*B*b**2*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) + I*B*b**2*log(t
an(e + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - B*b**2*l
og(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - 2*B*b**2*tan(e +
f*x)**2/(-2*d*f*tan(e + f*x) - 2*I*d*f) - 3*B*b**2/(-2*d*f*tan(e + f*x) - 2
*I*d*f) + I*C*a**2*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - C*a**
2*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) - C*a**2*log(tan(e + f*x)**2 + 1)*tan
(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - I*C*a**2*log(tan(e + f*x)**2 +
1)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - I*C*a**2/(-2*d*f*tan(e + f*x) - 2*I*d*
f) + 6*C*a*b*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) + 6*I*C*a*b*f
*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) + 2*I*C*a*b*log(tan(e + f*x)**2 + 1)*tan
(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - 2*C*a*b*log(tan(e + f*x)**2 + 1
)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - 4*C*a*b*tan(e + f*x)**2/(-2*d*f*tan(e +
 f*x) - 2*I*d*f) - 6*C*a*b/(-2*d*f*tan(e + f*x) - 2*I*d*f) - 3*I*C*b**2*f*x
*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) + 3*C*b**2*f*x/(-2*d*f*tan(e
+ f*x) - 2*I*d*f) + 2*C*b**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*
tan(e + f*x) - 2*I*d*f) + 2*I*C*b**2*log(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e
 + f*x) - 2*I*d*f) - C*b**2*tan(e + f*x)**3/(-2*d*f*tan(e + f*x) - 2*I*d*f)
 + I*C*b**2*tan(e + f*x)**2/(-2*d*f*tan(e + f*x) - 2*I*d*f) + 3*I*C*b**2/(-
2*d*f*tan(e + f*x) - 2*I*d*f), Eq(c, I*d)), ((A*a**2*x + A*a*b*log(tan(e +
f*x)**2 + 1)/f - A*b**2*x + A*b**2*tan(e + f*x)/f + B*a**2*log(tan(e + f*x)
**2 + 1)/(2*f) - 2*B*a*b*x + 2*B*a*b*tan(e + f*x)/f - B*b**2*log(tan(e + f*
x)**2 + 1)/(2*f) + B*b**2*tan(e + f*x)**2/(2*f) - C*a**2*x + C*a**2*tan(e +
 f*x)/f - C*a*b*log(tan(e + f*x)**2 + 1)/f + C*a*b*tan(e + f*x)**2/f + C*b*
*2*x + C*b**2*tan(e + f*x)**3/(3*f) - C*b**2*tan(e + f*x)/f)/c, Eq(d, 0)),
(x*(a + b*tan(e))**2*(A + B*tan(e) + C*tan(e)**2)/(c + d*tan(e)), Eq(f, 0))
, (2*A*a**2*c*d**3*f*x/(2*c**2*d**3*f + 2*d**5*f) + 2*A*a**2*d**4*log(c/d +
```

```
tan(e + f*x))/(2*c**2*d**3*f + 2*d**5*f) - A*a**2*d**4*log(tan(e + f*x)**2
 + 1)/(2*c**2*d**3*f + 2*d**5*f) - 4*A*a*b*c*d**3*log(c/d + tan(e + f*x))/(
2*c**2*d**3*f + 2*d**5*f) + 2*A*a*b*c*d**3*log(tan(e + f*x)**2 + 1)/(2*c**2
*d**3*f + 2*d**5*f) + 4*A*a*b*d**4*f*x/(2*c**2*d**3*f + 2*d**5*f) + 2*A*b**
2*c**2*d**2*log(c/d + tan(e + f*x))/(2*c**2*d**3*f + 2*d**5*f) - 2*A*b**2*c
*d**3*f*x/(2*c**2*d**3*f + 2*d**5*f) + A*b**2*d**4*log(tan(e + f*x)**2 + 1)
/(2*c**2*d**3*f + 2*d**5*f) - 2*B*a**2*c*d**3*log(c/d + tan(e + f*x))/(2*c*
*2*d**3*f + 2*d**5*f) + B*a**2*c*d**3*log(tan(e + f*x)**2 + 1)/(2*c**2*d**3
*f + 2*d**5*f) + 2*B*a**2*d**4*f*x/(2*c**2*d**3*f + 2*d**5*f) + 4*B*a*b*c**
2*d**2*log(c/d + tan(e + f*x))/(2*c**2*d**3*f + 2*d**5*f) - 4*B*a*b*c*d**3*
f*x/(2*c**2*d**3*f + 2*d**5*f) + 2*B*a*b*d**4*log(tan(e + f*x)**2 + 1)/(2*c
**2*d**3*f + 2*d**5*f) - 2*B*b**2*c**3*d*log(c/d + tan(e + f*x))/(2*c**2*d*
*3*f + 2*d**5*f) + 2*B*b**2*c**2*d**2*tan(e + f*x)/(2*c**2*d**3*f + 2*d**5*
f) - B*b**2*c*d**3*log(tan(e + f*x)**2 + 1)/(2*c**2*d**3*f + 2*d**5*f) - 2*
B*b**2*d**4*f*x/(2*c**2*d**3*f + 2*d**5*f) + 2*B*b**2*d**4*tan(e + f*x)/(2*
c**2*d**3*f + 2*d**5*f) + 2*C*a**2*c**2*d**2*log(c/d + tan(e + f*x))/(2*c**
2*d**3*f + 2*d**5*f) - 2*C*a**2*c*d**3*f*x/(2*c**2*d**3*f + 2*d**5*f) + C*a
**2*d**4*log(tan(e + f*x)**2 + 1)/(2*c**2*d**3*f + 2*d**5*f) - 4*C*a*b*c**3
*d*log(c/d + tan(e + f*x))/(2*c**2*d**3*f + 2*d**5*f) + 4*C*a*b*c**2*d**2*t
an(e + f*x)/(2*c**2*d**3*f + 2*d**5*f) - 2*C*a*b*c*d**3*log(tan(e + f*x)**2
 + 1)/(2*c**2*d**3*f + 2*d**5*f) - 4*C*a*b*d**4*f*x/(2*c**2*d**3*f + 2*d**5
*f) + 4*C*a*b*d**4*tan(e + f*x)/(2*c**2*d**3*f + 2*d**5*f) + 2*C*b**2*c**4*
log(c/d + tan(e + f*x))/(2*c**2*d**3*f + 2*d**5*f) - 2*C*b**2*c**3*d*tan(e
+ f*x)/(2*c**2*d**3*f + 2*d**5*f) + C*b**2*c**2*d**2*tan(e + f*x)**2/(2*c**
2*d**3*f + 2*d**5*f) + 2*C*b**2*c*d**3*f*x/(2*c**2*d**3*f + 2*d**5*f) - 2*C
*b**2*c*d**3*tan(e + f*x)/(2*c**2*d**3*f + 2*d**5*f) - C*b**2*d**4*log(tan(
e + f*x)**2 + 1)/(2*c**2*d**3*f + 2*d**5*f) + C*b**2*d**4*tan(e + f*x)**2/(
2*c**2*d**3*f + 2*d**5*f), True))
```

3.72 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$

Optimal. Leaf size=156

$$-\frac{(bc-ad)\left(Ad^2-Bcd+c^2C\right)\log(c+d\tan(e+fx))}{d^2f\left(c^2+d^2\right)}-\frac{\log(\cos(e+fx))(-aAd+aBc+aCd+Abc+bBd-bcC}{f\left(c^2+d^2\right)}$$

```
[Out] (a*(A*c+B*d-C*c)-b*(B*c-(A-C)*d))*x/(c^2+d^2)-(-A*a*d+A*b*c+B*a*c+B*b*d+C*a
*d-C*b*c)*ln(cos(f*x+e))/(c^2+d^2)/f-(-a*d+b*c)*(A*d^2-B*c*d+C*c^2)*ln(c+d*
tan(f*x+e))/d^2/(c^2+d^2)/f+b*C*tan(f*x+e)/d/f
```

Rubi [A] time = 0.34, antiderivative size = 156, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 5, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3637, 3626, 3617, 31, 3475}

$$-\frac{(bc-ad)\left(Ad^2-Bcd+c^2C\right)\log(c+d\tan(e+fx))}{d^2f\left(c^2+d^2\right)}-\frac{\log(\cos(e+fx))(-aAd+aBc+aCd+Abc+bBd-bcC}{f\left(c^2+d^2\right)}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d*T
an[e + f*x]),x]
```

```
[Out] ((a*(A*c - c*C + B*d) - b*(B*c - (A - C)*d))*x)/(c^2 + d^2) - ((A*b*c + a*B
*c - b*c*C - a*A*d + b*B*d + a*C*d)*Log[Cos[e + f*x]])/((c^2 + d^2)*f) - ((
b*c - a*d)*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d^2*(c^2 + d^2
)*f) + (b*C*Tan[e + f*x])/(d*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx=\frac{bC\tan(e+fx)}{df}-\frac{\int \frac{bcC-aAd-(Ab+aB-bC)d\tan(e+fx}{c+d\tan(e}}{d}$$

$$=\frac{(a(Ac-cC+Bd)-b(Bc-(A-C)d))x}{c^2+d^2}+\frac{bC\,t}{}$$

$$=\frac{(a(Ac-cC+Bd)-b(Bc-(A-C)d))x}{c^2+d^2}-\frac{(Ab}{}$$

$$=\frac{(a(Ac-cC+Bd)-b(Bc-(A-C)d))x}{c^2+d^2}-\frac{(Ab}{}$$

Mathematica [C] time = 1.09, size = 148, normalized size = 0.95

$$\frac{\frac{2(ad-bc)\left(Ad^2-Bcd+c^2C\right)\log(c+d\tan(e+fx))}{d^2\left(c^2+d^2\right)}+\frac{(b-ia)(A+iB-C)\log(-\tan(e+fx)+i)}{c+id}+\frac{(b+ia)(A-iB-C)\log(\tan(e+fx)+i)}{c-id}+\frac{2bC\tan(e+fx)}{d}}{2f}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x]),x]
```

```
[Out] ((((-I)*a + b)*(A + I*B - C)*Log[I - Tan[e + f*x]])/(c + I*d) + ((I*a + b)*
(A - I*B - C)*Log[I + Tan[e + f*x]])/(c - I*d) + (2*(-(b*c) + a*d)*(c^2*C -
 B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d^2*(c^2 + d^2)) + (2*b*C*Tan[e +
 f*x])/d)/(2*f)
```

fricas [A] time = 0.85, size = 212, normalized size = 1.36

$$\frac{2\left(((A-C)a-Bb)cd^2+(Ba+(A-C)b)d^3\right)fx-\left(Cbc^3-Aad^3-(Ca+Bb)c^2d+(Ba+Ab)cd^2\right)\log\left(\frac{d^2\tan\left(fx+e\right.}{\text{ta}}\right.}{2\left(c\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a - B*b)*c*d^2 + (B*a + (A - C)*b)*d^3)*f*x - (C*b*c^3 - A
*a*d^3 - (C*a + B*b)*c^2*d + (B*a + A*b)*c*d^2)*log((d^2*tan(f*x + e)^2 + 2
*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) + (C*b*c^3 + C*b*c*d^2 - (C*
a + B*b)*c^2*d - (C*a + B*b)*d^3)*log(1/(tan(f*x + e)^2 + 1)) + 2*(C*b*c^2*
d + C*b*d^3)*tan(f*x + e))/((c^2*d^2 + d^4)*f)
```

giac [A] time = 1.93, size = 186, normalized size = 1.19

$$\frac{\frac{2\,Cb\tan\left(fx+e\right)}{d}+\frac{2\,(Aac-Cac-Bbc+Bad+Abd-Cbd)\left(fx+e\right)}{c^2+d^2}+\frac{(Bac+Abc-Cbc-Aad+Cad+Bbd)\log\left(\tan\left(fx+e\right)^2+1\right)}{c^2+d^2}-\frac{2\left(Cbc^3-Cac^2d-Bbc^2d+B\right.}{}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*b*tan(f*x + e)/d + 2*(A*a*c - C*a*c - B*b*c + B*a*d + A*b*d - C*b*
d)*(f*x + e)/(c^2 + d^2) + (B*a*c + A*b*c - C*b*c - A*a*d + C*a*d + B*b*d)*
log(tan(f*x + e)^2 + 1)/(c^2 + d^2) - 2*(C*b*c^3 - C*a*c^2*d - B*b*c^2*d +
B*a*c*d^2 + A*b*c*d^2 - A*a*d^3)*log(abs(d*tan(f*x + e) + c))/(c^2*d^2 + d^
4))/f
```

maple [B] time = 0.25, size = 506, normalized size = 3.24

$$\frac{bC\tan\left(fx+e\right)}{df}+\frac{d\ln\left(c+d\tan\left(fx+e\right)\right)Aa}{f\left(c^2+d^2\right)}-\frac{\ln\left(c+d\tan\left(fx+e\right)\right)Abc}{f\left(c^2+d^2\right)}-\frac{\ln\left(c+d\tan\left(fx+e\right)\right)Bac}{f\left(c^2+d^2\right)}+\frac{\ln\left(c+\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x)
```

```
[Out] b*C*tan(f*x+e)/d/f+1/f*d/(c^2+d^2)*ln(c+d*tan(f*x+e))*A*a-1/f/(c^2+d^2)*ln(
c+d*tan(f*x+e))*A*b*c-1/f/(c^2+d^2)*ln(c+d*tan(f*x+e))*B*a*c+1/f/d/(c^2+d^2
)*ln(c+d*tan(f*x+e))*B*c^2*b+1/f/d/(c^2+d^2)*ln(c+d*tan(f*x+e))*C*c^2*a-1/f
/d^2/(c^2+d^2)*ln(c+d*tan(f*x+e))*C*c^3*b-1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2
)*A*a*d+1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*b*c+1/2/f/(c^2+d^2)*ln(1+tan(f
*x+e)^2)*B*a*c+1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*b*d+1/2/f/(c^2+d^2)*ln(
1+tan(f*x+e)^2)*a*C*d-1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*C*b*c+1/f/(c^2+d^2
)*A*arctan(tan(f*x+e))*a*c+1/f/(c^2+d^2)*A*arctan(tan(f*x+e))*b*d+1/f/(c^2+
d^2)*B*arctan(tan(f*x+e))*a*d-1/f/(c^2+d^2)*B*arctan(tan(f*x+e))*b*c-1/f/(c
^2+d^2)*C*arctan(tan(f*x+e))*a*c-1/f/(c^2+d^2)*C*arctan(tan(f*x+e))*b*d
```

maxima [A] time = 1.49, size = 178, normalized size = 1.14

$$\frac{\frac{2\,Cb\tan\left(fx+e\right)}{d}+\frac{2\,\left(\left(\left(A-C\right)a-Bb\right)c+\left(Ba+\left(A-C\right)b\right)d\right)\left(fx+e\right)}{c^2+d^2}-\frac{2\,\left(Cbc^3-Aad^3-\left(Ca+Bb\right)c^2d+\left(Ba+Ab\right)cd^2\right)\log\left(d\tan\left(fx+e\right)+c\right)}{c^2d^2+d^4}+\frac{\left(\left(Ba+\left(A-C\right)b\right)c-\left(\cdots\right.}{\cdots}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*b*tan(f*x + e)/d + 2*(((A - C)*a - B*b)*c + (B*a + (A - C)*b)*d)*(
f*x + e)/(c^2 + d^2) - 2*(C*b*c^3 - A*a*d^3 - (C*a + B*b)*c^2*d + (B*a + A*
b)*c*d^2)*log(d*tan(f*x + e) + c)/(c^2*d^2 + d^4) + ((B*a + (A - C)*b)*c -
((A - C)*a - B*b)*d)*log(tan(f*x + e)^2 + 1)/(c^2 + d^2))/f
```

mupad [B] time = 10.25, size = 186, normalized size = 1.19

$$\frac{\ln\left(\tan\left(e+f\,x\right)-\mathrm{i}\right)\,\left(A\,b+B\,a-C\,b-A\,a\,1\mathrm{i}+B\,b\,1\mathrm{i}+C\,a\,1\mathrm{i}\right)}{2\,f\,\left(c+d\,1\mathrm{i}\right)}+\frac{\ln\left(\tan\left(e+f\,x\right)+1\mathrm{i}\right)\,\left(B\,b+A\,b\,1\mathrm{i}+B\,a\,1\mathrm{i}-A\cdots\right)}{2\,f\,\left(d+c\,1\mathrm{i}\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x)),x)
```

```
[Out] (log(tan(e + f*x) - 1i)*(A*b - A*a*1i + B*a + B*b*1i + C*a*1i - C*b))/(2*f*
(c + d*1i)) + (log(tan(e + f*x) + 1i)*(A*b*1i - A*a + B*a*1i + B*b + C*a -
C*b*1i))/(2*f*(c*1i + d)) - (log(c + d*tan(e + f*x))*(d^2*(A*b*c + B*a*c) -
 d*(B*b*c^2 + C*a*c^2) - A*a*d^3 + C*b*c^3))/(f*(d^4 + c^2*d^2)) + (C*b*tan
(e + f*x))/(d*f)
```

sympy [A] time = 2.41, size = 2429, normalized size = 15.57

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
),x)
```

```
[Out] Piecewise((zoo*x*(a + b*tan(e))*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(c,
0) & Eq(d, 0) & Eq(f, 0)), ((A*a*x + A*b*log(tan(e + f*x)**2 + 1)/(2*f) + B
*a*log(tan(e + f*x)**2 + 1)/(2*f) - B*b*x + B*b*tan(e + f*x)/f - C*a*x + C*
a*tan(e + f*x)/f - C*b*log(tan(e + f*x)**2 + 1)/(2*f) + C*b*tan(e + f*x)**2
/(2*f))/c, Eq(d, 0)), (-I*A*a*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d
*f) - A*a*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) - I*A*a/(-2*d*f*tan(e + f*x)
+ 2*I*d*f) - A*b*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + I*A*b*f
*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) + A*b/(-2*d*f*tan(e + f*x) + 2*I*d*f) -
B*a*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + I*B*a*f*x/(-2*d*f*ta
n(e + f*x) + 2*I*d*f) + B*a/(-2*d*f*tan(e + f*x) + 2*I*d*f) - I*B*b*f*x*tan
(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) - B*b*f*x/(-2*d*f*tan(e + f*x) +
2*I*d*f) - B*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) +
 2*I*d*f) + I*B*b*log(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x) + 2*I*d*f)
+ I*B*b/(-2*d*f*tan(e + f*x) + 2*I*d*f) - I*C*a*f*x*tan(e + f*x)/(-2*d*f*ta
n(e + f*x) + 2*I*d*f) - C*a*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) - C*a*log(t
an(e + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + I*C*a*lo
g(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + I*C*a/(-2*d*f*tan(
e + f*x) + 2*I*d*f) + 3*C*b*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f
) - 3*I*C*b*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) - I*C*b*log(tan(e + f*x)**2
 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) - C*b*log(tan(e + f*x)**
2 + 1)/(-2*d*f*tan(e + f*x) + 2*I*d*f) - 2*C*b*tan(e + f*x)**2/(-2*d*f*tan(
e + f*x) + 2*I*d*f) - 3*C*b/(-2*d*f*tan(e + f*x) + 2*I*d*f), Eq(c, -I*d)),
(I*A*a*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - A*a*f*x/(-2*d*f*t
an(e + f*x) - 2*I*d*f) + I*A*a/(-2*d*f*tan(e + f*x) - 2*I*d*f) - A*b*f*x*ta
n(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - I*A*b*f*x/(-2*d*f*tan(e + f*x)
 - 2*I*d*f) + A*b/(-2*d*f*tan(e + f*x) - 2*I*d*f) - B*a*f*x*tan(e + f*x)/(-
2*d*f*tan(e + f*x) - 2*I*d*f) - I*B*a*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) +
 B*a/(-2*d*f*tan(e + f*x) - 2*I*d*f) + I*B*b*f*x*tan(e + f*x)/(-2*d*f*tan(e
 + f*x) - 2*I*d*f) - B*b*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) - B*b*log(tan(
e + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - I*B*b*log(t
an(e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - I*B*b/(-2*d*f*tan(e +
 f*x) - 2*I*d*f) + I*C*a*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) -
 C*a*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) - C*a*log(tan(e + f*x)**2 + 1)*tan
(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - I*C*a*log(tan(e + f*x)**2 + 1)/
(-2*d*f*tan(e + f*x) - 2*I*d*f) - I*C*a/(-2*d*f*tan(e + f*x) - 2*I*d*f) + 3
*C*b*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) + 3*I*C*b*f*x/(-2*d*f
```

```
*tan(e + f*x) - 2*I*d*f) + I*C*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*
d*f*tan(e + f*x) - 2*I*d*f) - C*b*log(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e +
f*x) - 2*I*d*f) - 2*C*b*tan(e + f*x)**2/(-2*d*f*tan(e + f*x) - 2*I*d*f) - 3
*C*b/(-2*d*f*tan(e + f*x) - 2*I*d*f), Eq(c, I*d)), (x*(a + b*tan(e))*(A + B
*tan(e) + C*tan(e)**2)/(c + d*tan(e)), Eq(f, 0)), (2*A*a*c*d**2*f*x/(2*c**2
*d**2*f + 2*d**4*f) + 2*A*a*d**3*log(c/d + tan(e + f*x))/(2*c**2*d**2*f + 2
*d**4*f) - A*a*d**3*log(tan(e + f*x)**2 + 1)/(2*c**2*d**2*f + 2*d**4*f) - 2
*A*b*c*d**2*log(c/d + tan(e + f*x))/(2*c**2*d**2*f + 2*d**4*f) + A*b*c*d**2
*log(tan(e + f*x)**2 + 1)/(2*c**2*d**2*f + 2*d**4*f) + 2*A*b*d**3*f*x/(2*c*
*2*d**2*f + 2*d**4*f) - 2*B*a*c*d**2*log(c/d + tan(e + f*x))/(2*c**2*d**2*f
 + 2*d**4*f) + B*a*c*d**2*log(tan(e + f*x)**2 + 1)/(2*c**2*d**2*f + 2*d**4*
f) + 2*B*a*d**3*f*x/(2*c**2*d**2*f + 2*d**4*f) + 2*B*b*c**2*d*log(c/d + tan
(e + f*x))/(2*c**2*d**2*f + 2*d**4*f) - 2*B*b*c*d**2*f*x/(2*c**2*d**2*f + 2
*d**4*f) + B*b*d**3*log(tan(e + f*x)**2 + 1)/(2*c**2*d**2*f + 2*d**4*f) + 2
*C*a*c**2*d*log(c/d + tan(e + f*x))/(2*c**2*d**2*f + 2*d**4*f) - 2*C*a*c*d*
*2*f*x/(2*c**2*d**2*f + 2*d**4*f) + C*a*d**3*log(tan(e + f*x)**2 + 1)/(2*c*
*2*d**2*f + 2*d**4*f) - 2*C*b*c**3*log(c/d + tan(e + f*x))/(2*c**2*d**2*f +
 2*d**4*f) + 2*C*b*c**2*d*tan(e + f*x)/(2*c**2*d**2*f + 2*d**4*f) - C*b*c*d
**2*log(tan(e + f*x)**2 + 1)/(2*c**2*d**2*f + 2*d**4*f) - 2*C*b*d**3*f*x/(2
*c**2*d**2*f + 2*d**4*f) + 2*C*b*d**3*tan(e + f*x)/(2*c**2*d**2*f + 2*d**4*
f), True))
```

3.73 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx$

Optimal. Leaf size=99

$$\frac{\left(Ad^2-Bcd+c^2C\right)\log(c+d\tan(e+fx))}{df\left(c^2+d^2\right)}-\frac{(Bc-d(A-C))\log(\cos(e+fx))}{f\left(c^2+d^2\right)}+\frac{x(Ac+Bd-cC)}{c^2+d^2}$$

```
[Out] (A*c+B*d-C*c)*x/(c^2+d^2)-(B*c-(A-C)*d)*ln(cos(f*x+e))/(c^2+d^2)/f+(A*d^2-B
*c*d+C*c^2)*ln(c+d*tan(f*x+e))/d/(c^2+d^2)/f
```

Rubi [A] time = 0.10, antiderivative size = 99, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 33, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.121, Rules used = {3626, 3617, 31, 3475}

$$\frac{\left(Ad^2-Bcd+c^2C\right)\log(c+d\tan(e+fx))}{df\left(c^2+d^2\right)}-\frac{(Bc-d(A-C))\log(\cos(e+fx))}{f\left(c^2+d^2\right)}+\frac{x(Ac+Bd-cC)}{c^2+d^2}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x]),x]
```

```
[Out] ((A*c - c*C + B*d)*x)/(c^2 + d^2) - ((B*c - (A - C)*d)*Log[Cos[e + f*x]])/(
(c^2 + d^2)*f) + ((c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/(d*(c^2
+ d^2)*f)
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{c+d\tan(e+fx)}\,dx = \frac{(Ac-cC+Bd)x}{c^2+d^2} - \frac{(-Bc+Ad-Cd)\int \tan(e+fx)\,dx}{c^2+d^2} + \frac{(c^2C-Bc\ldots}{\ldots}$$

$$= \frac{(Ac-cC+Bd)x}{c^2+d^2} - \frac{(Bc-(A-C)d)\log(\cos(e+fx))}{(c^2+d^2)f} + \frac{(c^2C-Bcd\ldots}{\ldots}$$

$$= \frac{(Ac-cC+Bd)x}{c^2+d^2} - \frac{(Bc-(A-C)d)\log(\cos(e+fx))}{(c^2+d^2)f} + \frac{(c^2C-Bcd\ldots}{\ldots}$$

Mathematica [C] time = 0.20, size = 117, normalized size = 1.18

$$\frac{\frac{2(Ad^2-Bcd+c^2C)\log(c+d\tan(e+fx))}{d(c^2+d^2)}+\frac{(-iA+B+iC)\log(-\tan(e+fx)+i)}{c+id}+\frac{(iA+B-iC)\log(\tan(e+fx)+i)}{c-id}}{2f}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x]),x]
```

```
[Out] ((((-I)*A + B + I*C)*Log[I - Tan[e + f*x]])/(c + I*d) + ((I*A + B - I*C)*Lo
g[I + Tan[e + f*x]])/(c - I*d) + (2*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e
 + f*x]])/(d*(c^2 + d^2)))/(2*f)
```

fricas [A] time = 1.18, size = 118, normalized size = 1.19

$$\frac{2\left((A-C)cd+Bd^2\right)fx+\left(Cc^2-Bcd+Ad^2\right)\log\left(\frac{d^2\tan\left(fx+e\right)^2+2\,cd\tan\left(fx+e\right)+c^2}{\tan\left(fx+e\right)^2+1}\right)-\left(Cc^2+Cd^2\right)\log\left(\frac{1}{\tan\left(fx+e\right)^2+1}\right)}{2\left(c^2d+d^3\right)f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x, algorithm="fr
icas")
```

```
[Out] 1/2*(2*((A - C)*c*d + B*d^2)*f*x + (C*c^2 - B*c*d + A*d^2)*log((d^2*tan(f*x
 + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) - (C*c^2 + C*d^2)
*log(1/(tan(f*x + e)^2 + 1)))/((c^2*d + d^3)*f)
```

giac [A] time = 2.06, size = 109, normalized size = 1.10

$$\frac{\frac{2\,(Ac-Cc+Bd)\left(fx+e\right)}{c^2+d^2}+\frac{(Bc-Ad+Cd)\log\left(\tan\left(fx+e\right)^2+1\right)}{c^2+d^2}+\frac{2\left(Cc^2-Bcd+Ad^2\right)\log\left(\left|d\tan\left(fx+e\right)+c\right|\right)}{c^2d+d^3}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x, algorithm="gi
ac")
```

```
[Out] 1/2*(2*(A*c - C*c + B*d)*(f*x + e)/(c^2 + d^2) + (B*c - A*d + C*d)*log(tan(
f*x + e)^2 + 1)/(c^2 + d^2) + 2*(C*c^2 - B*c*d + A*d^2)*log(abs(d*tan(f*x +
 e) + c))/(c^2*d + d^3))/f
```

maple [B] time = 0.26, size = 234, normalized size = 2.36

$$\frac{d\ln\left(c+d\tan\left(fx+e\right)\right)A}{f\left(c^2+d^2\right)}-\frac{\ln\left(c+d\tan\left(fx+e\right)\right)Bc}{f\left(c^2+d^2\right)}+\frac{\ln\left(c+d\tan\left(fx+e\right)\right)c^2C}{f\left(c^2+d^2\right)d}-\frac{\ln\left(1+\tan^2\left(fx+e\right)\right)Ad}{2f\left(c^2+d^2\right)}+$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x)
```

```
[Out] 1/f/(c^2+d^2)*d*ln(c+d*tan(f*x+e))*A-1/f/(c^2+d^2)*ln(c+d*tan(f*x+e))*B*c+1
/f/(c^2+d^2)/d*ln(c+d*tan(f*x+e))*c^2*C-1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*
A*d+1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*c+1/2/f/(c^2+d^2)*ln(1+tan(f*x+e)^
2)*C*d+1/f/(c^2+d^2)*A*arctan(tan(f*x+e))*c+1/f/(c^2+d^2)*B*arctan(tan(f*x+
e))*d-1/f/(c^2+d^2)*C*arctan(tan(f*x+e))*c
```

maxima [A] time = 0.55, size = 106, normalized size = 1.07

$$\frac{\frac{2\,((A-C)c+Bd)\left(fx+e\right)}{c^2+d^2}+\frac{2\left(Cc^2-Bcd+Ad^2\right)\log\left(d\tan\left(fx+e\right)+c\right)}{c^2d+d^3}+\frac{(Bc-(A-C)d)\log\left(\tan\left(fx+e\right)^2+1\right)}{c^2+d^2}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x, algorithm="ma
xima")
```

```
[Out] 1/2*(2*((A - C)*c + B*d)*(f*x + e)/(c^2 + d^2) + 2*(C*c^2 - B*c*d + A*d^2)*
log(d*tan(f*x + e) + c)/(c^2*d + d^3) + (B*c - (A - C)*d)*log(tan(f*x + e)^
2 + 1)/(c^2 + d^2))/f
```

mupad [B] time = 9.90, size = 109, normalized size = 1.10

$$\frac{\ln\left(\tan\left(e+f\,x\right)+1\mathrm{i}\right)\,\left(C-A+B\,1\mathrm{i}\right)}{2\,f\,\left(d+c\,1\mathrm{i}\right)}+\frac{\ln\left(\tan\left(e+f\,x\right)-\mathrm{i}\right)\,\left(B-A\,1\mathrm{i}+C\,1\mathrm{i}\right)}{2\,f\,\left(c+d\,1\mathrm{i}\right)}+\frac{\ln\left(c+d\tan\left(e+f\,x\right)\right)\,\left(C\,c^2-B\right.}{d\,f\,\left(c^2+d^2\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x)),x)
```

```
[Out] (log(tan(e + f*x) + 1i)*(B*1i - A + C))/(2*f*(c*1i + d)) + (log(tan(e + f*x
) - 1i)*(B - A*1i + C*1i))/(2*f*(c + d*1i)) + (log(c + d*tan(e + f*x))*(A*d
^2 + C*c^2 - B*c*d))/(d*f*(c^2 + d^2))
```

sympy [A] time = 1.31, size = 984, normalized size = 9.94

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)),x)
```

```
[Out] Piecewise((zoo*x*(A + B*tan(e) + C*tan(e)**2)/tan(e), Eq(c, 0) & Eq(d, 0) &
 Eq(f, 0)), ((A*x + B*log(tan(e + f*x)**2 + 1)/(2*f) - C*x + C*tan(e + f*x)
/f)/c, Eq(d, 0)), (-I*A*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) -
A*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f) - I*A/(-2*d*f*tan(e + f*x) + 2*I*d*f)
 - B*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + I*B*f*x/(-2*d*f*tan
(e + f*x) + 2*I*d*f) + B/(-2*d*f*tan(e + f*x) + 2*I*d*f) - I*C*f*x*tan(e +
f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f) - C*f*x/(-2*d*f*tan(e + f*x) + 2*I*d*f
) - C*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*tan(e + f*x) + 2*I*d*f)
 + I*C*log(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x) + 2*I*d*f) + I*C/(-2*d
*f*tan(e + f*x) + 2*I*d*f), Eq(c, -I*d)), (I*A*f*x*tan(e + f*x)/(-2*d*f*tan
(e + f*x) - 2*I*d*f) - A*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) + I*A/(-2*d*f*
tan(e + f*x) - 2*I*d*f) - B*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f
) - I*B*f*x/(-2*d*f*tan(e + f*x) - 2*I*d*f) + B/(-2*d*f*tan(e + f*x) - 2*I*
d*f) + I*C*f*x*tan(e + f*x)/(-2*d*f*tan(e + f*x) - 2*I*d*f) - C*f*x/(-2*d*f
*tan(e + f*x) - 2*I*d*f) - C*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(-2*d*f*
tan(e + f*x) - 2*I*d*f) - I*C*log(tan(e + f*x)**2 + 1)/(-2*d*f*tan(e + f*x)
 - 2*I*d*f) - I*C/(-2*d*f*tan(e + f*x) - 2*I*d*f), Eq(c, I*d)), (x*(A + B*t
an(e) + C*tan(e)**2)/(c + d*tan(e)), Eq(f, 0)), (2*A*c*d*f*x/(2*c**2*d*f +
2*d**3*f) + 2*A*d**2*log(c/d + tan(e + f*x))/(2*c**2*d*f + 2*d**3*f) - A*d*
*2*log(tan(e + f*x)**2 + 1)/(2*c**2*d*f + 2*d**3*f) - 2*B*c*d*log(c/d + tan
(e + f*x))/(2*c**2*d*f + 2*d**3*f) + B*c*d*log(tan(e + f*x)**2 + 1)/(2*c**2
```

```
*d*f + 2*d**3*f) + 2*B*d**2*f*x/(2*c**2*d*f + 2*d**3*f) + 2*C*c**2*log(c/d
+ tan(e + f*x))/(2*c**2*d*f + 2*d**3*f) - 2*C*c*d*f*x/(2*c**2*d*f + 2*d**3*
f) + C*d**2*log(tan(e + f*x)**2 + 1)/(2*c**2*d*f + 2*d**3*f), True))
```

3.74 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx$

Optimal. Leaf size=165

$$\frac{x(a(Ac+Bd-cC)+b(Bc-d(A-C)))}{(a^2+b^2)(c^2+d^2)}+\frac{(Ab^2-a(bB-aC))\log(a\cos(e+fx)+b\sin(e+fx))}{f(a^2+b^2)(bc-ad)}-\frac{(Ad^2-Bcd+c$$

```
[Out] (a*(A*c+B*d-C*c)+b*(B*c-(A-C)*d))*x/(a^2+b^2)/(c^2+d^2)+(A*b^2-a*(B*b-C*a))
*ln(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^2)/(-a*d+b*c)/f-(A*d^2-B*c*d+C*c^2)*l
n(c*cos(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)/(c^2+d^2)/f
```

Rubi [A] time = 0.26, antiderivative size = 164, normalized size of antiderivative = 0.99, number of steps used = 3, number of rules used = 2, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.044, Rules used = {3651, 3530}

$$\frac{x(a(Ac+Bd-cC)-bd(A-C)+bBc)}{(a^2+b^2)(c^2+d^2)}+\frac{(Ab^2-a(bB-aC))\log(a\cos(e+fx)+b\sin(e+fx))}{f(a^2+b^2)(bc-ad)}-\frac{(Ad^2-Bcd+c$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c + d*Ta
n[e + f*x])),x]
```

```
[Out] ((b*B*c - b*(A - C)*d + a*(A*c - c*C + B*d))*x)/((a^2 + b^2)*(c^2 + d^2)) +
 ((A*b^2 - a*(b*B - a*C))*Log[a*Cos[e + f*x] + b*Sin[e + f*x]])/((a^2 + b^2
)*(b*c - a*d)*f) - ((c^2*C - B*c*d + A*d^2)*Log[c*Cos[e + f*x] + d*Sin[e +
f*x]])/((b*c - a*d)*(c^2 + d^2)*f)
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
```

```
NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))}\,dx = \frac{(bBc-b(A-C)d+a(Ac-cC+Bd))x}{\left(a^2+b^2\right)\left(c^2+d^2\right)} + \frac{\left(Ab^2-a(bB-aC)\right)\int}{\left(a^2+b^2\right)(bc}$$

$$= \frac{(bBc-b(A-C)d+a(Ac-cC+Bd))x}{\left(a^2+b^2\right)\left(c^2+d^2\right)} + \frac{\left(Ab^2-a(bB-aC)\right)\mathrm{lo}}{\left(a^2+\right.}$$

Mathematica [A] time = 1.50, size = 313, normalized size = 1.90

$$-\frac{\frac{\log\left(\sqrt{-b^2}-b\tan(e+fx)\right)\left(\frac{\sqrt{-b^2}(a(Ac+Bd-cC)+bd(C-A)+bBc)}{b}+aAd-aBc-aCd+Abc+bBd-bcC\right)}{\left(a^2+b^2\right)\left(c^2+d^2\right)}+\frac{\log\left(\sqrt{-b^2}+b\tan(e+fx)\right)\left(\frac{b(a(Ac+Bd-cC)+bd(C-A)+b}{\sqrt{-b^2}}\right.}{\left(a^2+b^2\right)\left(c^2+d^2\right.}}{2f}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c
+ d*Tan[e + f*x])),x]
```

```
[Out] -1/2*(((A*b*c - a*B*c - b*c*C + a*A*d + b*B*d - a*C*d + (Sqrt[-b^2]*(b*B*c
+ b*(-A + C)*d + a*(A*c - c*C + B*d)))/b)*Log[Sqrt[-b^2] - b*Tan[e + f*x]])
/((a^2 + b^2)*(c^2 + d^2)) + (2*(A*b^2 + a*(-(b*B) + a*C))*Log[a + b*Tan[e
+ f*x]])/((a^2 + b^2)*(-(b*c) + a*d)) + ((A*b*c - a*B*c - b*c*C + a*A*d + b
*B*d - a*C*d + (b*(b*B*c + b*(-A + C)*d + a*(A*c - c*C + B*d)))/Sqrt[-b^2])
*Log[Sqrt[-b^2] + b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2)) + (2*(c^2*C -
B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/f
```

fricas [A] time = 0.92, size = 301, normalized size = 1.82

$$\frac{2\left(\left(\left((A-C)ab+Bb^2\right)c^2-\left((A-C)a^2+(A-C)b^2\right)cd-\left(Ba^2-(A-C)ab\right)d^2\right)fx+\left(\left(Ca^2-Bab+Ab^2\right)c^2+\left(C\right.\right.\right.}{2\left(\left(a^2b+b^3\right)c^3-\left(\right.\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(((A - C)*a*b + B*b^2)*c^2 - ((A - C)*a^2 + (A - C)*b^2)*c*d - (B*a^
2 - (A - C)*a*b)*d^2)*f*x + ((C*a^2 - B*a*b + A*b^2)*c^2 + (C*a^2 - B*a*b +
 A*b^2)*d^2)*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x +
 e)^2 + 1)) - ((C*a^2 + C*b^2)*c^2 - (B*a^2 + B*b^2)*c*d + (A*a^2 + A*b^2)*
d^2)*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 +
1)))/(((a^2*b + b^3)*c^3 - (a^3 + a*b^2)*c^2*d + (a^2*b + b^3)*c*d^2 - (a^3
 + a*b^2)*d^3)*f)
```

giac [A] time = 2.33, size = 272, normalized size = 1.65

$$\frac{\frac{2\,(Aac-Cac+Bbc+Bad-Abd+Cbd)\left(fx+e\right)}{a^2c^2+b^2c^2+a^2d^2+b^2d^2}+\frac{(Bac-Abc+Cbc-Aad+Cad-Bbd)\log\left(\tan\left(fx+e\right)^2+1\right)}{a^2c^2+b^2c^2+a^2d^2+b^2d^2}+\frac{2\left(Ca^2b-Bab^2+Ab^3\right)\log\left(\left|b\tan\left(fx+e\right)+a\right|\right)}{a^2b^2c+b^4c-a^3bd-ab^3d}-}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c - C*a*c + B*b*c + B*a*d - A*b*d + C*b*d)*(f*x + e)/(a^2*c^2 +
 b^2*c^2 + a^2*d^2 + b^2*d^2) + (B*a*c - A*b*c + C*b*c - A*a*d + C*a*d - B*
b*d)*log(tan(f*x + e)^2 + 1)/(a^2*c^2 + b^2*c^2 + a^2*d^2 + b^2*d^2) + 2*(C
*a^2*b - B*a*b^2 + A*b^3)*log(abs(b*tan(f*x + e) + a))/(a^2*b^2*c + b^4*c -
 a^3*b*d - a*b^3*d) - 2*(C*c^2*d - B*c*d^2 + A*d^3)*log(abs(d*tan(f*x + e)
+ c))/(b*c^3*d - a*c^2*d^2 + b*c*d^3 - a*d^4))/f
```

maple [B] time = 0.52, size = 647, normalized size = 3.92

$$-\frac{\ln\left(a+b\tan\left(fx+e\right)\right)A\,b^2}{f\,(da-cb)\left(a^2+b^2\right)}+\frac{\ln\left(a+b\tan\left(fx+e\right)\right)Bab}{f\,(da-cb)\left(a^2+b^2\right)}-\frac{\ln\left(a+b\tan\left(fx+e\right)\right)a^2C}{f\,(da-cb)\left(a^2+b^2\right)}+\frac{\ln\left(c+d\tan\left(fx+e\right)\right)A}{f\,(da-cb)\left(c^2+d^2\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)),x)
```

```
[Out] -1/f/(a*d-b*c)/(a^2+b^2)*ln(a+b*tan(f*x+e))*A*b^2+1/f/(a*d-b*c)/(a^2+b^2)*l
n(a+b*tan(f*x+e))*B*a*b-1/f/(a*d-b*c)/(a^2+b^2)*ln(a+b*tan(f*x+e))*a^2*C+1/
f/(a*d-b*c)/(c^2+d^2)*ln(c+d*tan(f*x+e))*A*d^2-1/f/(a*d-b*c)/(c^2+d^2)*ln(c
+d*tan(f*x+e))*B*c*d+1/f/(a*d-b*c)/(c^2+d^2)*ln(c+d*tan(f*x+e))*c^2*C-1/2/f
/(c^2+d^2)/(a^2+b^2)*ln(1+tan(f*x+e)^2)*A*a*d-1/2/f/(c^2+d^2)/(a^2+b^2)*ln(
1+tan(f*x+e)^2)*A*b*c+1/2/f/(c^2+d^2)/(a^2+b^2)*ln(1+tan(f*x+e)^2)*B*a*c-1/
2/f/(c^2+d^2)/(a^2+b^2)*ln(1+tan(f*x+e)^2)*B*b*d+1/2/f/(c^2+d^2)/(a^2+b^2)*
ln(1+tan(f*x+e)^2)*a*C*d+1/2/f/(c^2+d^2)/(a^2+b^2)*ln(1+tan(f*x+e)^2)*C*b*c
+1/f/(c^2+d^2)/(a^2+b^2)*A*arctan(tan(f*x+e))*a*c-1/f/(c^2+d^2)/(a^2+b^2)*A
*arctan(tan(f*x+e))*b*d+1/f/(c^2+d^2)/(a^2+b^2)*B*arctan(tan(f*x+e))*a*d+1/
```

```
f/(c^2+d^2)/(a^2+b^2)*B*arctan(tan(f*x+e))*b*c-1/f/(c^2+d^2)/(a^2+b^2)*C*ar
ctan(tan(f*x+e))*a*c+1/f/(c^2+d^2)/(a^2+b^2)*C*arctan(tan(f*x+e))*b*d
```

maxima [A] time = 0.47, size = 243, normalized size = 1.47

$$\frac{\frac{2\,(((A-C)a+Bb)c+(Ba-(A-C)b)d)\left(fx+e\right)}{\left(a^2+b^2\right)c^2+\left(a^2+b^2\right)d^2}+\frac{2\left(Ca^2-Bab+Ab^2\right)\log\left(b\tan\left(fx+e\right)+a\right)}{\left(a^2b+b^3\right)c-\left(a^3+ab^2\right)d}-\frac{2\left(Cc^2-Bcd+Ad^2\right)\log\left(d\tan\left(fx+e\right)+c\right)}{bc^3-ac^2d+bcd^2-ad^3}+\frac{((Ba-(A-C)b)c-}{(}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a + B*b)*c + (B*a - (A - C)*b)*d)*(f*x + e)/((a^2 + b^2)*c
^2 + (a^2 + b^2)*d^2) + 2*(C*a^2 - B*a*b + A*b^2)*log(b*tan(f*x + e) + a)/(
(a^2*b + b^3)*c - (a^3 + a*b^2)*d) - 2*(C*c^2 - B*c*d + A*d^2)*log(d*tan(f*
x + e) + c)/(b*c^3 - a*c^2*d + b*c*d^2 - a*d^3) + ((B*a - (A - C)*b)*c - ((
A - C)*a + B*b)*d)*log(tan(f*x + e)^2 + 1)/((a^2 + b^2)*c^2 + (a^2 + b^2)*d
^2))/f
```

mupad [B] time = 21.40, size = 196, normalized size = 1.19

$$\frac{\ln\left(c+d\tan\left(e+f\,x\right)\right)\,\left(C\,c^2-B\,c\,d+A\,d^2\right)}{f\,\left(a\,d-b\,c\right)\,\left(c^2+d^2\right)}+\frac{\ln\left(\tan\left(e+f\,x\right)+1\mathrm{i}\right)\,\left(C-A+B\,1\mathrm{i}\right)}{2\,f\,\left(a\,c\,1\mathrm{i}+a\,d+b\,c-b\,d\,1\mathrm{i}\right)}-\frac{\ln\left(a+b\tan\left(e+f\,x\right)\right)\,\left(C\right.}{f\,\left(d\,a^3-c\,a^2\,b+a\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))),x)
```

```
[Out] (log(tan(e + f*x) + 1i)*(B*1i - A + C))/(2*f*(a*c*1i + a*d + b*c - b*d*1i))
 - (log(tan(e + f*x) - 1i)*(A + B*1i - C))/(2*f*(a*d - a*c*1i + b*c + b*d*1
i)) - (log(a + b*tan(e + f*x))*(A*b^2 + C*a^2 - B*a*b))/(f*(a^3*d - b^3*c -
 a^2*b*c + a*b^2*d)) + (log(c + d*tan(e + f*x))*(A*d^2 + C*c^2 - B*c*d))/(f
*(a*d - b*c)*(c^2 + d^2))
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)
),x)
```

```
[Out] Timed out
```

3.75 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx$

Optimal. Leaf size=281

$$\frac{x\left(a^2(Ac+Bd-cC)+2ab(Bc-d(A-C))-b^2(Ac+Bd-cC)\right)}{\left(a^2+b^2\right)^2\left(c^2+d^2\right)}-\frac{Ab^2-a(bB-aC)}{f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))}+\frac{\left(a^4(-\right.}{}$$

```
[Out] (a^2*(A*c+B*d-C*c)-b^2*(A*c+B*d-C*c)+2*a*b*(B*c-(A-C)*d))*x/(a^2+b^2)^2/(c^
2+d^2)+(2*a*b^3*c*(A-C)+2*a^3*b*B*d-a^4*C*d+b^4*(-A*d+B*c)-a^2*b^2*(3*A*d+B
*c-C*d))*ln(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^2)^2/(-a*d+b*c)^2/f+d*(A*d^2-
B*c*d+C*c^2)*ln(c*cos(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)^2/(c^2+d^2)/f+(-A*b^2
+a*(B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))
```

Rubi [A] time = 0.80, antiderivative size = 281, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 3, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3649, 3651, 3530}

$$\frac{x\left(a^2(Ac+Bd-cC)+2ab(Bc-d(A-C))-b^2(Ac+Bd-cC)\right)}{\left(a^2+b^2\right)^2\left(c^2+d^2\right)}-\frac{Ab^2-a(bB-aC)}{f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))}+\frac{\left(-a^2b\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(c + d*
Tan[e + f*x])),x]
```

```
[Out] ((a^2*(A*c - c*C + B*d) - b^2*(A*c - c*C + B*d) + 2*a*b*(B*c - (A - C)*d))*
x)/((a^2 + b^2)^2*(c^2 + d^2)) + ((2*a*b^3*c*(A - C) + 2*a^3*b*B*d - a^4*C*
d + b^4*(B*c - A*d) - a^2*b^2*(B*c + 3*A*d - C*d))*Log[a*Cos[e + f*x] + b*S
in[e + f*x]])/((a^2 + b^2)^2*(b*c - a*d)^2*f) + (d*(c^2*C - B*c*d + A*d^2)*
Log[c*Cos[e + f*x] + d*Sin[e + f*x]])/((b*c - a*d)^2*(c^2 + d^2)*f) - (A*b^
2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))}\,dx = -\frac{Ab^2-a(bB-aC)}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))} - \frac{\int \frac{-abc(A-C)+a^2Ad-b^2(}{}}{}$$

$$= \frac{\left(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)+2ab(Bc-(A-C)d)\right)x}{\left(a^2+b^2\right)^2\left(c^2+d^2\right)}$$

$$= \frac{\left(a^2(Ac-cC+Bd)-b^2(Ac-cC+Bd)+2ab(Bc-(A-C)d)\right)x}{\left(a^2+b^2\right)^2\left(c^2+d^2\right)}$$

Mathematica [A] time = 6.96, size = 543, normalized size = 1.93

$$-\frac{\frac{(bc-ad)\log\left(\sqrt{-b^2}-b\tan(e+fx)\right)\left(\frac{\sqrt{-b^2}\left(a^2(Ac+Bd-cC)+2ab(d(C-A)+Bc)-b^2(Ac+Bd-cC)\right)}{b}+a^2Ad+a^2(-B)c-a^2Cd+2aAbc+2abBd-2abcC-Ab^2d+b^2Bc+b^2}{2\left(a^2+b^2\right)\left(c^2+d^2\right)}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])),x]
```

```
[Out] (-1/2*((b*c - a*d)*(2*a*A*b*c - a^2*B*c + b^2*B*c - 2*a*b*c*C + a^2*A*d - A
*b^2*d + 2*a*b*B*d - a^2*C*d + b^2*C*d + (Sqrt[-b^2]*(a^2*(A*c - c*C + B*d)
 - b^2*(A*c - c*C + B*d) + 2*a*b*(B*c + (-A + C)*d)))/b)*Log[Sqrt[-b^2] - b
*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2)) + ((2*a*b^3*c*(-A + C) - 2*a^3*b*
B*d + a^4*C*d + b^4*(-(B*c) + A*d) + a^2*b^2*(B*c + 3*A*d - C*d))*Log[a + b
*Tan[e + f*x]])/((a^2 + b^2)*(-(b*c) + a*d)) - ((b*c - a*d)*(2*a*A*b*c - a^
2*B*c + b^2*B*c - 2*a*b*c*C + a^2*A*d - A*b^2*d + 2*a*b*B*d - a^2*C*d + b^2
*C*d + (Sqrt[-b^2]*(-(a^2*(A*c - c*C + B*d)) + b^2*(A*c - c*C + B*d) - 2*a*
b*(B*c + (-A + C)*d)))/b)*Log[Sqrt[-b^2] + b*Tan[e + f*x]])/(2*(a^2 + b^2)*
(c^2 + d^2)) + ((a^2 + b^2)*d*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x
]])/((b*c - a*d)*(c^2 + d^2)) - (A*b^2)/(a + b*Tan[e + f*x]) + (a*(b*B - a*
C))/(a + b*Tan[e + f*x]))/((a^2 + b^2)*(b*c - a*d)*f)
```

fricas [B] time = 2.54, size = 1345, normalized size = 4.79

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] -1/2*(2*(C*a^2*b^3 - B*a*b^4 + A*b^5)*c^3 - 2*(C*a^3*b^2 - B*a^2*b^3 + A*a*
b^4)*c^2*d + 2*(C*a^2*b^3 - B*a*b^4 + A*b^5)*c*d^2 - 2*(C*a^3*b^2 - B*a^2*b
^3 + A*a*b^4)*d^3 - 2*(((A - C)*a^3*b^2 + 2*B*a^2*b^3 - (A - C)*a*b^4)*c^3
- (2*(A - C)*a^4*b + 3*B*a^3*b^2 + B*a*b^4)*c^2*d + ((A - C)*a^5 + 3*(A - C
)*a^3*b^2 + 2*B*a^2*b^3)*c*d^2 + (B*a^5 - 2*(A - C)*a^4*b - B*a^3*b^2)*d^3)
*f*x + ((B*a^3*b^2 - 2*(A - C)*a^2*b^3 - B*a*b^4)*c^3 + (C*a^5 - 2*B*a^4*b
+ (3*A - C)*a^3*b^2 + A*a*b^4)*c^2*d + (B*a^3*b^2 - 2*(A - C)*a^2*b^3 - B*a
*b^4)*c*d^2 + (C*a^5 - 2*B*a^4*b + (3*A - C)*a^3*b^2 + A*a*b^4)*d^3 + ((B*a
^2*b^3 - 2*(A - C)*a*b^4 - B*b^5)*c^3 + (C*a^4*b - 2*B*a^3*b^2 + (3*A - C)*
a^2*b^3 + A*b^5)*c^2*d + (B*a^2*b^3 - 2*(A - C)*a*b^4 - B*b^5)*c*d^2 + (C*a
^4*b - 2*B*a^3*b^2 + (3*A - C)*a^2*b^3 + A*b^5)*d^3)*tan(f*x + e))*log((b^2
*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - ((C*a^5
 + 2*C*a^3*b^2 + C*a*b^4)*c^2*d - (B*a^5 + 2*B*a^3*b^2 + B*a*b^4)*c*d^2 + (
A*a^5 + 2*A*a^3*b^2 + A*a*b^4)*d^3 + ((C*a^4*b + 2*C*a^2*b^3 + C*b^5)*c^2*d
 - (B*a^4*b + 2*B*a^2*b^3 + B*b^5)*c*d^2 + (A*a^4*b + 2*A*a^2*b^3 + A*b^5)*
d^3)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan
(f*x + e)^2 + 1)) - 2*((C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c^3 - (C*a^4*b - B
*a^3*b^2 + A*a^2*b^3)*c^2*d + (C*a^3*b^2 - B*a^2*b^3 + A*a*b^4)*c*d^2 - (C*
a^4*b - B*a^3*b^2 + A*a^2*b^3)*d^3 + (((A - C)*a^2*b^3 + 2*B*a*b^4 - (A - C
)*b^5)*c^3 - (2*(A - C)*a^3*b^2 + 3*B*a^2*b^3 + B*b^5)*c^2*d + ((A - C)*a^4
*b + 3*(A - C)*a^2*b^3 + 2*B*a*b^4)*c*d^2 + (B*a^4*b - 2*(A - C)*a^3*b^2 -
```

```
B*a^2*b^3)*d^3)*f*x)*tan(f*x + e))/(((a^4*b^3 + 2*a^2*b^5 + b^7)*c^4 - 2*(a
^5*b^2 + 2*a^3*b^4 + a*b^6)*c^3*d + (a^6*b + 3*a^4*b^3 + 3*a^2*b^5 + b^7)*c
^2*d^2 - 2*(a^5*b^2 + 2*a^3*b^4 + a*b^6)*c*d^3 + (a^6*b + 2*a^4*b^3 + a^2*b
^5)*d^4)*f*tan(f*x + e) + ((a^5*b^2 + 2*a^3*b^4 + a*b^6)*c^4 - 2*(a^6*b + 2
*a^4*b^3 + a^2*b^5)*c^3*d + (a^7 + 3*a^5*b^2 + 3*a^3*b^4 + a*b^6)*c^2*d^2 -
 2*(a^6*b + 2*a^4*b^3 + a^2*b^5)*c*d^3 + (a^7 + 2*a^5*b^2 + a^3*b^4)*d^4)*f
)
```

giac [B] time = 6.62, size = 846, normalized size = 3.01

$$\frac{\frac{2\left(Aa^2c-Ca^2c+2\,Babc-Ab^2c+Cb^2c+Ba^2d-2\,Aabd+2\,Cabd-Bb^2d\right)\left(fx+e\right)}{a^4c^2+2\,a^2b^2c^2+b^4c^2+a^4d^2+2\,a^2b^2d^2+b^4d^2}+\frac{\left(Ba^2c-2\,Aabc+2\,Cabc-Bb^2c-Aa^2d+Ca^2d-2\,Babd+Ab^2d-Cb^2d\right)\log\left(\text{ta}\right.}{a^4c^2+2\,a^2b^2c^2+b^4c^2+a^4d^2+2\,a^2b^2d^2+b^4d^2}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c - C*a^2*c + 2*B*a*b*c - A*b^2*c + C*b^2*c + B*a^2*d - 2*A*a
*b*d + 2*C*a*b*d - B*b^2*d)*(f*x + e)/(a^4*c^2 + 2*a^2*b^2*c^2 + b^4*c^2 +
a^4*d^2 + 2*a^2*b^2*d^2 + b^4*d^2) + (B*a^2*c - 2*A*a*b*c + 2*C*a*b*c - B*b
^2*c - A*a^2*d + C*a^2*d - 2*B*a*b*d + A*b^2*d - C*b^2*d)*log(tan(f*x + e)^
2 + 1)/(a^4*c^2 + 2*a^2*b^2*c^2 + b^4*c^2 + a^4*d^2 + 2*a^2*b^2*d^2 + b^4*d
^2) - 2*(B*a^2*b^3*c - 2*A*a*b^4*c + 2*C*a*b^4*c - B*b^5*c + C*a^4*b*d - 2*
B*a^3*b^2*d + 3*A*a^2*b^3*d - C*a^2*b^3*d + A*b^5*d)*log(abs(b*tan(f*x + e)
 + a))/(a^4*b^3*c^2 + 2*a^2*b^5*c^2 + b^7*c^2 - 2*a^5*b^2*c*d - 4*a^3*b^4*c
*d - 2*a*b^6*c*d + a^6*b*d^2 + 2*a^4*b^3*d^2 + a^2*b^5*d^2) + 2*(C*c^2*d^2
- B*c*d^3 + A*d^4)*log(abs(d*tan(f*x + e) + c))/(b^2*c^4*d - 2*a*b*c^3*d^2
+ a^2*c^2*d^3 + b^2*c^2*d^3 - 2*a*b*c*d^4 + a^2*d^5) + 2*(B*a^2*b^3*c*tan(f
*x + e) - 2*A*a*b^4*c*tan(f*x + e) + 2*C*a*b^4*c*tan(f*x + e) - B*b^5*c*tan
(f*x + e) + C*a^4*b*d*tan(f*x + e) - 2*B*a^3*b^2*d*tan(f*x + e) + 3*A*a^2*b
^3*d*tan(f*x + e) - C*a^2*b^3*d*tan(f*x + e) + A*b^5*d*tan(f*x + e) - C*a^4
*b*c + 2*B*a^3*b^2*c - 3*A*a^2*b^3*c + C*a^2*b^3*c - A*b^5*c + 2*C*a^5*d -
3*B*a^4*b*d + 4*A*a^3*b^2*d - B*a^2*b^3*d + 2*A*a*b^4*d)/((a^4*b^2*c^2 + 2*
a^2*b^4*c^2 + b^6*c^2 - 2*a^5*b*c*d - 4*a^3*b^3*c*d - 2*a*b^5*c*d + a^6*d^2
 + 2*a^4*b^2*d^2 + a^2*b^4*d^2)*(b*tan(f*x + e) + a)))/f
```

maple [B] time = 0.55, size = 1262, normalized size = 4.49

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e)),x)
```

```
[Out] 1/f/(a*d-b*c)^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*b^4*c+1/f/(a*d-b*c)^2/(a^2
+b^2)^2*ln(a+b*tan(f*x+e))*C*a^2*b^2*d-2/f/(a*d-b*c)^2/(a^2+b^2)^2*ln(a+b*t
an(f*x+e))*C*a*b^3*c+2/f/(a^2+b^2)^2/(c^2+d^2)*C*arctan(tan(f*x+e))*a*b*d+2
/f/(a*d-b*c)^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*a*b^3*c+2/f/(a*d-b*c)^2/(a^
2+b^2)^2*ln(a+b*tan(f*x+e))*a^3*b*B*d-1/f/(a*d-b*c)^2/(a^2+b^2)^2*ln(a+b*ta
n(f*x+e))*B*a^2*b^2*c-1/f/(a^2+b^2)^2/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*a*b*d+
1/f/(a^2+b^2)^2/(c^2+d^2)*ln(1+tan(f*x+e)^2)*C*a*b*c-2/f/(a^2+b^2)^2/(c^2+d
^2)*A*arctan(tan(f*x+e))*a*b*d+2/f/(a^2+b^2)^2/(c^2+d^2)*B*arctan(tan(f*x+e
))*a*b*c-1/f/(a*d-b*c)/(a^2+b^2)/(a+b*tan(f*x+e))*B*a*b-1/f*d^2/(a*d-b*c)^2
/(c^2+d^2)*ln(c+d*tan(f*x+e))*B*c-3/f/(a*d-b*c)^2/(a^2+b^2)^2*ln(a+b*tan(f*
x+e))*A*a^2*b^2*d-1/f/(a^2+b^2)^2/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*a*b*c+1/f*
d/(a*d-b*c)^2/(c^2+d^2)*ln(c+d*tan(f*x+e))*c^2*C-1/2/f/(a^2+b^2)^2/(c^2+d^2
)*ln(1+tan(f*x+e)^2)*A*a^2*d+1/2/f/(a^2+b^2)^2/(c^2+d^2)*ln(1+tan(f*x+e)^2)
*A*b^2*d+1/2/f/(a^2+b^2)^2/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*a^2*c-1/f/(a^2+b^
2)^2/(c^2+d^2)*B*arctan(tan(f*x+e))*b^2*d-1/f/(a^2+b^2)^2/(c^2+d^2)*C*arcta
n(tan(f*x+e))*a^2*c+1/f/(a^2+b^2)^2/(c^2+d^2)*C*arctan(tan(f*x+e))*b^2*c-1/
f/(a^2+b^2)^2/(c^2+d^2)*A*arctan(tan(f*x+e))*b^2*c+1/f/(a^2+b^2)^2/(c^2+d^2
)*B*arctan(tan(f*x+e))*a^2*d-1/f/(a*d-b*c)^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))
*A*b^4*d-1/f/(a*d-b*c)^2/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*a^4*C*d-1/2/f/(a^2+
b^2)^2/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*b^2*c+1/2/f/(a^2+b^2)^2/(c^2+d^2)*ln(
1+tan(f*x+e)^2)*C*a^2*d-1/2/f/(a^2+b^2)^2/(c^2+d^2)*ln(1+tan(f*x+e)^2)*C*b^
2*d+1/f/(a^2+b^2)^2/(c^2+d^2)*A*arctan(tan(f*x+e))*a^2*c+1/f/(a*d-b*c)/(a^2
+b^2)/(a+b*tan(f*x+e))*A*b^2+1/f/(a*d-b*c)/(a^2+b^2)/(a+b*tan(f*x+e))*a^2*C
+1/f*d^3/(a*d-b*c)^2/(c^2+d^2)*ln(c+d*tan(f*x+e))*A
```

maxima [A] time = 0.51, size = 520, normalized size = 1.85

$$\frac{\frac{2\left(\left((A-C)a^2+2\,Bab-(A-C)b^2\right)c+\left(Ba^2-2\,(A-C)ab-Bb^2\right)d\right)(fx+e)}{\left(a^4+2\,a^2b^2+b^4\right)c^2+\left(a^4+2\,a^2b^2+b^4\right)d^2}-\frac{2\left(\left(Ba^2b^2-2\,(A-C)ab^3-Bb^4\right)c+\left(Ca^4-2\,Ba^3b+(3\,A-C)a^2b^2+Ab^4\right)d\right)\log\left(b\tan\left(fx\right.\right.}{\left(a^4b^2+2\,a^2b^4+b^6\right)c^2-2\,\left(a^5b+2\,a^3b^3+ab^5\right)cd+\left(a^6+2\,a^4b^2+a^2b^4\right)d^2}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c + (B*a^2 - 2*(A - C)*a*b -
B*b^2)*d)*(f*x + e)/((a^4 + 2*a^2*b^2 + b^4)*c^2 + (a^4 + 2*a^2*b^2 + b^4)*
d^2) - 2*((B*a^2*b^2 - 2*(A - C)*a*b^3 - B*b^4)*c + (C*a^4 - 2*B*a^3*b + (3
*A - C)*a^2*b^2 + A*b^4)*d)*log(b*tan(f*x + e) + a)/((a^4*b^2 + 2*a^2*b^4 +
 b^6)*c^2 - 2*(a^5*b + 2*a^3*b^3 + a*b^5)*c*d + (a^6 + 2*a^4*b^2 + a^2*b^4)
*d^2) + 2*(C*c^2*d - B*c*d^2 + A*d^3)*log(d*tan(f*x + e) + c)/(b^2*c^4 - 2*
a*b*c^3*d - 2*a*b*c*d^3 + a^2*d^4 + (a^2 + b^2)*c^2*d^2) + ((B*a^2 - 2*(A -
 C)*a*b - B*b^2)*c - ((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*d)*log(tan(f*x +
 e)^2 + 1)/((a^4 + 2*a^2*b^2 + b^4)*c^2 + (a^4 + 2*a^2*b^2 + b^4)*d^2) - 2*
```

```
(C*a^2 - B*a*b + A*b^2)/((a^3*b + a*b^3)*c - (a^4 + a^2*b^2)*d + ((a^2*b^2
+ b^4)*c - (a^3*b + a*b^3)*d)*tan(f*x + e)))/f
```

mupad [B] time = 63.66, size = 393, normalized size = 1.40

$$\frac{\ln\left(\tan\left(e+f\,x\right)-\mathrm{i}\right)\,\left(B-A\,1\mathrm{i}+C\,1\mathrm{i}\right)}{2\,f\,\left(a^2\,c-b^2\,c-2\,a\,b\,d+a^2\,d\,1\mathrm{i}-b^2\,d\,1\mathrm{i}+a\,b\,c\,2\mathrm{i}\right)}-\frac{\ln\left(\tan\left(e+f\,x\right)+1\mathrm{i}\right)\,\left(A\,1\mathrm{i}+B-C\,1\mathrm{i}\right)}{2\,f\,\left(b^2\,c-a^2\,c+2\,a\,b\,d+a^2\,d\,1\mathrm{i}-b^2\,d\,1\mathrm{i}+a\,b\,c\,2\mathrm{i}\right)}-\frac{\ln\left(a\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))),x)
```

```
[Out] (log(tan(e + f*x) - 1i)*(B - A*1i + C*1i))/(2*f*(a^2*c + a^2*d*1i - b^2*c -
 b^2*d*1i + a*b*c*2i - 2*a*b*d)) - (log(tan(e + f*x) + 1i)*(A*1i + B - C*1i
))/(2*f*(a^2*d*1i - a^2*c + b^2*c - b^2*d*1i + a*b*c*2i + 2*a*b*d)) - (log(
a + b*tan(e + f*x))*(b^4*(A*d - B*c) + a^2*b^2*(3*A*d + B*c - C*d) + C*a^4*
d - a*b^3*(2*A*c - 2*C*c) - 2*B*a^3*b*d))/(f*(a^6*d^2 + b^6*c^2 + 2*a^2*b^4
*c^2 + a^4*b^2*c^2 + a^2*b^4*d^2 + 2*a^4*b^2*d^2 - 2*a*b^5*c*d - 2*a^5*b*c*
d - 4*a^3*b^3*c*d)) + (A*b^2 + C*a^2 - B*a*b)/(f*(a*d - b*c)*(a^2 + b^2)*(a
 + b*tan(e + f*x))) + (d*log(c + d*tan(e + f*x))*(A*d^2 + C*c^2 - B*c*d))/(
f*(a*d - b*c)^2*(c^2 + d^2))
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: NotImplementedError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**2/(c+d*tan(f*x
+e)),x)
```

```
[Out] Exception raised: NotImplementedError
```

3.76 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))}\,dx$

Optimal. Leaf size=477

$$-\frac{Ab^2-a(bB-aC)}{2f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))^2}+\frac{x\left(a^3(Ac+Bd-cC)+3a^2b(Bc-d(A-C))-3ab^2(Ac+Bd-cC)-\right.}{\left(a^2+b^2\right)^3\left(c^2+d^2\right)}$$

```
[Out] (a^3*(A*c+B*d-C*c)-3*a*b^2*(A*c+B*d-C*c)+3*a^2*b*(B*c-(A-C)*d)-b^3*(B*c-(A-
C)*d))*x/(a^2+b^2)^3/(c^2+d^2)+(3*a*b^5*B*c^2-3*a^5*b*B*d^2+a^6*C*d^2+3*a^4
*b^2*d*(2*A*d+B*c-C*d)+b^6*(c*(-B*d+C*c)-A*(c^2-d^2))-a^3*b^3*(8*c*(A-C)*d+
B*(c^2-d^2))-3*a^2*b^4*(c*(2*B*d+C*c)-A*(c^2+d^2)))*ln(a*cos(f*x+e)+b*sin(f
*x+e))/(a^2+b^2)^3/(-a*d+b*c)^3/f-d^2*(A*d^2-B*c*d+C*c^2)*ln(c*cos(f*x+e)+d
*sin(f*x+e))/(-a*d+b*c)^3/(c^2+d^2)/f+1/2*(-A*b^2+a*(B*b-C*a))/(a^2+b^2)/(-
a*d+b*c)/f/(a+b*tan(f*x+e))^2+(-2*a*b^3*c*(A-C)-2*a^3*b*B*d+a^4*C*d-b^4*(-A
*d+B*c)+a^2*b^2*(3*A*d+B*c-C*d))/(a^2+b^2)^2/(-a*d+b*c)^2/f/(a+b*tan(f*x+e)
)
```

Rubi [A] time = 1.79, antiderivative size = 477, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 3, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3649, 3651, 3530}

$$\frac{\left(-a^3b^3\left(8cd(A-C)+B\left(c^2-d^2\right)\right)-3a^2b^4\left(c(2Bd+cC)-A\left(c^2+d^2\right)\right)+3a^4b^2d(2Ad+Bc-Cd)-3a^5bBd^2+a\right.}{f\left(a^2+b^2\right)^3(bc-ad)^3}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^3*(c + d*
Tan[e + f*x])),x]
```

```
[Out] ((a^3*(A*c - c*C + B*d) - 3*a*b^2*(A*c - c*C + B*d) + 3*a^2*b*(B*c - (A - C
)*d) - b^3*(B*c - (A - C)*d))*x)/((a^2 + b^2)^3*(c^2 + d^2)) + ((3*a*b^5*B*
c^2 - 3*a^5*b*B*d^2 + a^6*C*d^2 + 3*a^4*b^2*d*(B*c + 2*A*d - C*d) + b^6*(c*
(c*C - B*d) - A*(c^2 - d^2)) - a^3*b^3*(8*c*(A - C)*d + B*(c^2 - d^2)) - 3*
a^2*b^4*(c*(c*C + 2*B*d) - A*(c^2 + d^2)))*Log[a*Cos[e + f*x] + b*Sin[e + f
*x]])/((a^2 + b^2)^3*(b*c - a*d)^3*f) - (d^2*(c^2*C - B*c*d + A*d^2)*Log[c*
Cos[e + f*x] + d*Sin[e + f*x]])/((b*c - a*d)^3*(c^2 + d^2)*f) - (A*b^2 - a*
(b*B - a*C))/(2*(a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])^2) - (2*a*b^
3*c*(A - C) + 2*a^3*b*B*d - a^4*C*d + b^4*(B*c - A*d) - a^2*b^2*(B*c + 3*A*
d - C*d))/((a^2 + b^2)^2*(b*c - a*d)^2*f*(a + b*Tan[e + f*x]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
```

```
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))}\,dx = -\frac{Ab^2-a(bB-aC)}{2\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^2} - \frac{\int \frac{-2\left(abc(A-C)-a^2Ad+\right.}{}}{}$$

$$= -\frac{Ab^2-a(bB-aC)}{2\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^2} - \frac{2ab^3c(A-C)+2a}{\left(a^2\right.}$$

$$= \frac{\left(a^3(Ac-cC+Bd)-3ab^2(Ac-cC+Bd)+3a^2b(Bc-(A-C)d)\right.}{\left(a^2+b^2\right)^3\left(c^2+d^2\right)}$$

$$= \frac{\left(a^3(Ac-cC+Bd)-3ab^2(Ac-cC+Bd)+3a^2b(Bc-(A-C)d)\right.}{\left(a^2+b^2\right)^3\left(c^2+d^2\right)}$$

Mathematica [A] time = 9.03, size = 898, normalized size = 1.88

$$-\frac{Ab^2-a(bB-aC)}{2\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^2} - \frac{-\frac{-2\left(-Ada^2+bc(A-C)a+b^2(Bc-Ad)\right)b^2-a\left(2b(Ab-Cb-aB)(bc-ad)-2a\left(Ab^2-a(bB-aC)\right)d\right)}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^3*(
c + d*Tan[e + f*x])),x]
```

```
[Out] -1/2*(A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]
)^2) - (-((-((b*(b*c - a*d)^2*(3*a^2*A*b*c - A*b^3*c - a^3*B*c + 3*a*b^2*B*
c - 3*a^2*b*c*C + b^3*c*C + a^3*A*d - 3*a*A*b^2*d + 3*a^2*b*B*d - b^3*B*d -
 a^3*C*d + 3*a*b^2*C*d + (Sqrt[-b^2]*(a^3*(A*c - c*C + B*d) - 3*a*b^2*(A*c
- c*C + B*d) + 3*a^2*b*(B*c - (A - C)*d) - b^3*(B*c - (A - C)*d)))/b)*Log[S
qrt[-b^2] - b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2))) + (2*b*(3*a*b^5*B*c
^2 - 3*a^5*b*B*d^2 + a^6*C*d^2 + 3*a^4*b^2*d*(B*c + 2*A*d - C*d) + b^6*(c*(
c*C - B*d) - A*(c^2 - d^2)) - a^3*b^3*(8*c*(A - C)*d + B*(c^2 - d^2)) - 3*a
^2*b^4*(c*(c*C + 2*B*d) - A*(c^2 + d^2)))*Log[a + b*Tan[e + f*x]])/((a^2 +
b^2)*(b*c - a*d)) - (b*(b*c - a*d)^2*(3*a^2*A*b*c - A*b^3*c - a^3*B*c + 3*a
*b^2*B*c - 3*a^2*b*c*C + b^3*c*C + a^3*A*d - 3*a*A*b^2*d + 3*a^2*b*B*d - b^
3*B*d - a^3*C*d + 3*a*b^2*C*d - (Sqrt[-b^2]*(a^3*(A*c - c*C + B*d) - 3*a*b^
2*(A*c - c*C + B*d) + 3*a^2*b*(B*c - (A - C)*d) - b^3*(B*c - (A - C)*d)))/b
)*Log[Sqrt[-b^2] + b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2)) - (2*b*(a^2 +
```

```
 b^2)^2*d^2*(c^2*C - B*c*d + A*d^2)*Log[c + d*Tan[e + f*x]])/((b*c - a*d)*(
c^2 + d^2)))/(b*(a^2 + b^2)*(b*c - a*d)*f)) - (-(a*(-2*a*(A*b^2 - a*(b*B -
a*C))*d + 2*b*(A*b - a*B - b*C)*(b*c - a*d))) - 2*b^2*(a*b*c*(A - C) - a^2*
A*d + b^2*(B*c - A*d)))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])))/(
2*(a^2 + b^2)*(b*c - a*d))
```

fricas [B] time = 7.18, size = 3643, normalized size = 7.64

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] -1/2*((3*C*a^4*b^4 - 5*B*a^3*b^5 + (7*A - 3*C)*a^2*b^6 + B*a*b^7 + A*b^8)*c
^4 - 4*(2*C*a^5*b^3 - 3*B*a^4*b^4 + (4*A - C)*a^3*b^5 + A*a*b^7)*c^3*d + (5
*C*a^6*b^2 - 7*B*a^5*b^3 + (9*A + 2*C)*a^4*b^4 - 6*B*a^3*b^5 + (10*A - 3*C)
*a^2*b^6 + B*a*b^7 + A*b^8)*c^2*d^2 - 4*(2*C*a^5*b^3 - 3*B*a^4*b^4 + (4*A -
 C)*a^3*b^5 + A*a*b^7)*c*d^3 + (5*C*a^6*b^2 - 7*B*a^5*b^3 + (9*A - C)*a^4*b
^4 - B*a^3*b^5 + 3*A*a^2*b^6)*d^4 - 2*(((A - C)*a^5*b^3 + 3*B*a^4*b^4 - 3*(
A - C)*a^3*b^5 - B*a^2*b^6)*c^4 - (3*(A - C)*a^6*b^2 + 8*B*a^5*b^3 - 6*(A -
 C)*a^4*b^4 - (A - C)*a^2*b^6)*c^3*d + 3*((A - C)*a^7*b + 2*B*a^6*b^2 + 2*B
*a^4*b^4 - (A - C)*a^3*b^5)*c^2*d^2 - ((A - C)*a^8 + 6*(A - C)*a^6*b^2 + 8*
B*a^5*b^3 - 3*(A - C)*a^4*b^4)*c*d^3 - (B*a^8 - 3*(A - C)*a^7*b - 3*B*a^6*b
^2 + (A - C)*a^5*b^3)*d^4)*f*x - ((C*a^4*b^4 - 3*B*a^3*b^5 + 5*(A - C)*a^2*
b^6 + 3*B*a*b^7 - A*b^8)*c^4 - 4*(C*a^5*b^3 - 2*B*a^4*b^4 + (3*A - 2*C)*a^3
*b^5 + B*a^2*b^6)*c^3*d + (3*C*a^6*b^2 - 5*B*a^5*b^3 + (7*A - 2*C)*a^4*b^4
- 2*B*a^3*b^5 + (6*A - 5*C)*a^2*b^6 + 3*B*a*b^7 - A*b^8)*c^2*d^2 - 4*(C*a^5
*b^3 - 2*B*a^4*b^4 + (3*A - 2*C)*a^3*b^5 + B*a^2*b^6)*c*d^3 + (3*C*a^6*b^2
- 5*B*a^5*b^3 + (7*A - 3*C)*a^4*b^4 + B*a^3*b^5 + A*a^2*b^6)*d^4 + 2*(((A -
 C)*a^3*b^5 + 3*B*a^2*b^6 - 3*(A - C)*a*b^7 - B*b^8)*c^4 - (3*(A - C)*a^4*b
^4 + 8*B*a^3*b^5 - 6*(A - C)*a^2*b^6 - (A - C)*b^8)*c^3*d + 3*((A - C)*a^5*
b^3 + 2*B*a^4*b^4 + 2*B*a^2*b^6 - (A - C)*a*b^7)*c^2*d^2 - ((A - C)*a^6*b^2
 + 6*(A - C)*a^4*b^4 + 8*B*a^3*b^5 - 3*(A - C)*a^2*b^6)*c*d^3 - (B*a^6*b^2
- 3*(A - C)*a^5*b^3 - 3*B*a^4*b^4 + (A - C)*a^3*b^5)*d^4)*f*x)*tan(f*x + e)
^2 + ((B*a^5*b^3 - 3*(A - C)*a^4*b^4 - 3*B*a^3*b^5 + (A - C)*a^2*b^6)*c^4 -
 (3*B*a^6*b^2 - 8*(A - C)*a^5*b^3 - 6*B*a^4*b^4 - B*a^2*b^6)*c^3*d - (C*a^8
 - 3*B*a^7*b + 3*(2*A - C)*a^6*b^2 + 3*(2*A - C)*a^4*b^4 + 3*B*a^3*b^5 + C*
a^2*b^6)*c^2*d^2 - (3*B*a^6*b^2 - 8*(A - C)*a^5*b^3 - 6*B*a^4*b^4 - B*a^2*b
^6)*c*d^3 - (C*a^8 - 3*B*a^7*b + 3*(2*A - C)*a^6*b^2 + B*a^5*b^3 + 3*A*a^4*
b^4 + A*a^2*b^6)*d^4 + ((B*a^3*b^5 - 3*(A - C)*a^2*b^6 - 3*B*a*b^7 + (A - C
)*b^8)*c^4 - (3*B*a^4*b^4 - 8*(A - C)*a^3*b^5 - 6*B*a^2*b^6 - B*b^8)*c^3*d
- (C*a^6*b^2 - 3*B*a^5*b^3 + 3*(2*A - C)*a^4*b^4 + 3*(2*A - C)*a^2*b^6 + 3*
B*a*b^7 + C*b^8)*c^2*d^2 - (3*B*a^4*b^4 - 8*(A - C)*a^3*b^5 - 6*B*a^2*b^6 -
 B*b^8)*c*d^3 - (C*a^6*b^2 - 3*B*a^5*b^3 + 3*(2*A - C)*a^4*b^4 + B*a^3*b^5
```

```
+ 3*A*a^2*b^6 + A*b^8)*d^4)*tan(f*x + e)^2 + 2*((B*a^4*b^4 - 3*(A - C)*a^3*
b^5 - 3*B*a^2*b^6 + (A - C)*a*b^7)*c^4 - (3*B*a^5*b^3 - 8*(A - C)*a^4*b^4 -
 6*B*a^3*b^5 - B*a*b^7)*c^3*d - (C*a^7*b - 3*B*a^6*b^2 + 3*(2*A - C)*a^5*b^
3 + 3*(2*A - C)*a^3*b^5 + 3*B*a^2*b^6 + C*a*b^7)*c^2*d^2 - (3*B*a^5*b^3 - 8
*(A - C)*a^4*b^4 - 6*B*a^3*b^5 - B*a*b^7)*c*d^3 - (C*a^7*b - 3*B*a^6*b^2 +
3*(2*A - C)*a^5*b^3 + B*a^4*b^4 + 3*A*a^3*b^5 + A*a*b^7)*d^4)*tan(f*x + e))
*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1))
+ ((C*a^8 + 3*C*a^6*b^2 + 3*C*a^4*b^4 + C*a^2*b^6)*c^2*d^2 - (B*a^8 + 3*B*a
^6*b^2 + 3*B*a^4*b^4 + B*a^2*b^6)*c*d^3 + (A*a^8 + 3*A*a^6*b^2 + 3*A*a^4*b^
4 + A*a^2*b^6)*d^4 + ((C*a^6*b^2 + 3*C*a^4*b^4 + 3*C*a^2*b^6 + C*b^8)*c^2*d
^2 - (B*a^6*b^2 + 3*B*a^4*b^4 + 3*B*a^2*b^6 + B*b^8)*c*d^3 + (A*a^6*b^2 + 3
*A*a^4*b^4 + 3*A*a^2*b^6 + A*b^8)*d^4)*tan(f*x + e)^2 + 2*((C*a^7*b + 3*C*a
^5*b^3 + 3*C*a^3*b^5 + C*a*b^7)*c^2*d^2 - (B*a^7*b + 3*B*a^5*b^3 + 3*B*a^3*
b^5 + B*a*b^7)*c*d^3 + (A*a^7*b + 3*A*a^5*b^3 + 3*A*a^3*b^5 + A*a*b^7)*d^4)
*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x
 + e)^2 + 1)) - 2*((C*a^5*b^3 - 2*B*a^4*b^4 + 3*(A - C)*a^3*b^5 + 3*B*a^2*b
^6 - (3*A - 2*C)*a*b^7 - B*b^8)*c^4 - (3*C*a^6*b^2 - 5*B*a^5*b^3 + (7*A - 6
*C)*a^4*b^4 + 6*B*a^3*b^5 - 3*(2*A - C)*a^2*b^6 - B*a*b^7 - A*b^8)*c^3*d +
(2*C*a^7*b - 3*B*a^6*b^2 + 2*(2*A - C)*a^5*b^3 + B*a^4*b^4 - 2*C*a^3*b^5 +
3*B*a^2*b^6 - 2*(2*A - C)*a*b^7 - B*b^8)*c^2*d^2 - (3*C*a^6*b^2 - 5*B*a^5*b
^3 + (7*A - 6*C)*a^4*b^4 + 6*B*a^3*b^5 - 3*(2*A - C)*a^2*b^6 - B*a*b^7 - A*
b^8)*c*d^3 + (2*C*a^7*b - 3*B*a^6*b^2 + (4*A - 3*C)*a^5*b^3 + 3*B*a^4*b^4 -
 (3*A - C)*a^3*b^5 - A*a*b^7)*d^4 + 2*(((A - C)*a^4*b^4 + 3*B*a^3*b^5 - 3*(
A - C)*a^2*b^6 - B*a*b^7)*c^4 - (3*(A - C)*a^5*b^3 + 8*B*a^4*b^4 - 6*(A - C
)*a^3*b^5 - (A - C)*a*b^7)*c^3*d + 3*((A - C)*a^6*b^2 + 2*B*a^5*b^3 + 2*B*a
^3*b^5 - (A - C)*a^2*b^6)*c^2*d^2 - ((A - C)*a^7*b + 6*(A - C)*a^5*b^3 + 8*
B*a^4*b^4 - 3*(A - C)*a^3*b^5)*c*d^3 - (B*a^7*b - 3*(A - C)*a^6*b^2 - 3*B*a
^5*b^3 + (A - C)*a^4*b^4)*d^4)*f*x)*tan(f*x + e))/(((a^6*b^5 + 3*a^4*b^7 +
3*a^2*b^9 + b^11)*c^5 - 3*(a^7*b^4 + 3*a^5*b^6 + 3*a^3*b^8 + a*b^10)*c^4*d
+ (3*a^8*b^3 + 10*a^6*b^5 + 12*a^4*b^7 + 6*a^2*b^9 + b^11)*c^3*d^2 - (a^9*b
^2 + 6*a^7*b^4 + 12*a^5*b^6 + 10*a^3*b^8 + 3*a*b^10)*c^2*d^3 + 3*(a^8*b^3 +
 3*a^6*b^5 + 3*a^4*b^7 + a^2*b^9)*c*d^4 - (a^9*b^2 + 3*a^7*b^4 + 3*a^5*b^6
+ a^3*b^8)*d^5)*f*tan(f*x + e)^2 + 2*((a^7*b^4 + 3*a^5*b^6 + 3*a^3*b^8 + a*
b^10)*c^5 - 3*(a^8*b^3 + 3*a^6*b^5 + 3*a^4*b^7 + a^2*b^9)*c^4*d + (3*a^9*b^
2 + 10*a^7*b^4 + 12*a^5*b^6 + 6*a^3*b^8 + a*b^10)*c^3*d^2 - (a^10*b + 6*a^8
*b^3 + 12*a^6*b^5 + 10*a^4*b^7 + 3*a^2*b^9)*c^2*d^3 + 3*(a^9*b^2 + 3*a^7*b^
4 + 3*a^5*b^6 + a^3*b^8)*c*d^4 - (a^10*b + 3*a^8*b^3 + 3*a^6*b^5 + a^4*b^7)
*d^5)*f*tan(f*x + e) + ((a^8*b^3 + 3*a^6*b^5 + 3*a^4*b^7 + a^2*b^9)*c^5 - 3
*(a^9*b^2 + 3*a^7*b^4 + 3*a^5*b^6 + a^3*b^8)*c^4*d + (3*a^10*b + 10*a^8*b^3
 + 12*a^6*b^5 + 6*a^4*b^7 + a^2*b^9)*c^3*d^2 - (a^11 + 6*a^9*b^2 + 12*a^7*b
^4 + 10*a^5*b^6 + 3*a^3*b^8)*c^2*d^3 + 3*(a^10*b + 3*a^8*b^3 + 3*a^6*b^5 +
a^4*b^7)*c*d^4 - (a^11 + 3*a^9*b^2 + 3*a^7*b^4 + a^5*b^6)*d^5)*f)
```

giac [B] time = 26.16, size = 2127, normalized size = 4.46

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^3*c - C*a^3*c + 3*B*a^2*b*c - 3*A*a*b^2*c + 3*C*a*b^2*c - B*b^3
*c + B*a^3*d - 3*A*a^2*b*d + 3*C*a^2*b*d - 3*B*a*b^2*d + A*b^3*d - C*b^3*d)
*(f*x + e)/(a^6*c^2 + 3*a^4*b^2*c^2 + 3*a^2*b^4*c^2 + b^6*c^2 + a^6*d^2 + 3
*a^4*b^2*d^2 + 3*a^2*b^4*d^2 + b^6*d^2) + (B*a^3*c - 3*A*a^2*b*c + 3*C*a^2*
b*c - 3*B*a*b^2*c + A*b^3*c - C*b^3*c - A*a^3*d + C*a^3*d - 3*B*a^2*b*d + 3
*A*a*b^2*d - 3*C*a*b^2*d + B*b^3*d)*log(tan(f*x + e)^2 + 1)/(a^6*c^2 + 3*a^
4*b^2*c^2 + 3*a^2*b^4*c^2 + b^6*c^2 + a^6*d^2 + 3*a^4*b^2*d^2 + 3*a^2*b^4*d
^2 + b^6*d^2) - 2*(B*a^3*b^4*c^2 - 3*A*a^2*b^5*c^2 + 3*C*a^2*b^5*c^2 - 3*B*
a*b^6*c^2 + A*b^7*c^2 - C*b^7*c^2 - 3*B*a^4*b^3*c*d + 8*A*a^3*b^4*c*d - 8*C
*a^3*b^4*c*d + 6*B*a^2*b^5*c*d + B*b^7*c*d - C*a^6*b*d^2 + 3*B*a^5*b^2*d^2
- 6*A*a^4*b^3*d^2 + 3*C*a^4*b^3*d^2 - B*a^3*b^4*d^2 - 3*A*a^2*b^5*d^2 - A*b
^7*d^2)*log(abs(b*tan(f*x + e) + a))/(a^6*b^4*c^3 + 3*a^4*b^6*c^3 + 3*a^2*b
^8*c^3 + b^10*c^3 - 3*a^7*b^3*c^2*d - 9*a^5*b^5*c^2*d - 9*a^3*b^7*c^2*d - 3
*a*b^9*c^2*d + 3*a^8*b^2*c*d^2 + 9*a^6*b^4*c*d^2 + 9*a^4*b^6*c*d^2 + 3*a^2*
b^8*c*d^2 - a^9*b*d^3 - 3*a^7*b^3*d^3 - 3*a^5*b^5*d^3 - a^3*b^7*d^3) - 2*(C
*c^2*d^3 - B*c*d^4 + A*d^5)*log(abs(d*tan(f*x + e) + c))/(b^3*c^5*d - 3*a*b
^2*c^4*d^2 + 3*a^2*b*c^3*d^3 + b^3*c^3*d^3 - a^3*c^2*d^4 - 3*a*b^2*c^2*d^4
+ 3*a^2*b*c*d^5 - a^3*d^6) + (3*B*a^3*b^5*c^2*tan(f*x + e)^2 - 9*A*a^2*b^6*
c^2*tan(f*x + e)^2 + 9*C*a^2*b^6*c^2*tan(f*x + e)^2 - 9*B*a*b^7*c^2*tan(f*x
 + e)^2 + 3*A*b^8*c^2*tan(f*x + e)^2 - 3*C*b^8*c^2*tan(f*x + e)^2 - 9*B*a^4
*b^4*c*d*tan(f*x + e)^2 + 24*A*a^3*b^5*c*d*tan(f*x + e)^2 - 24*C*a^3*b^5*c*
d*tan(f*x + e)^2 + 18*B*a^2*b^6*c*d*tan(f*x + e)^2 + 3*B*b^8*c*d*tan(f*x +
e)^2 - 3*C*a^6*b^2*d^2*tan(f*x + e)^2 + 9*B*a^5*b^3*d^2*tan(f*x + e)^2 - 18
*A*a^4*b^4*d^2*tan(f*x + e)^2 + 9*C*a^4*b^4*d^2*tan(f*x + e)^2 - 3*B*a^3*b^
5*d^2*tan(f*x + e)^2 - 9*A*a^2*b^6*d^2*tan(f*x + e)^2 - 3*A*b^8*d^2*tan(f*x
 + e)^2 + 8*B*a^4*b^4*c^2*tan(f*x + e) - 22*A*a^3*b^5*c^2*tan(f*x + e) + 22
*C*a^3*b^5*c^2*tan(f*x + e) - 18*B*a^2*b^6*c^2*tan(f*x + e) + 2*A*a*b^7*c^2
*tan(f*x + e) - 2*C*a*b^7*c^2*tan(f*x + e) - 2*B*b^8*c^2*tan(f*x + e) + 2*C
*a^6*b^2*c*d*tan(f*x + e) - 24*B*a^5*b^3*c*d*tan(f*x + e) + 58*A*a^4*b^4*c*
d*tan(f*x + e) - 52*C*a^4*b^4*c*d*tan(f*x + e) + 32*B*a^3*b^5*c*d*tan(f*x +
 e) + 12*A*a^2*b^6*c*d*tan(f*x + e) - 6*C*a^2*b^6*c*d*tan(f*x + e) + 8*B*a*
b^7*c*d*tan(f*x + e) + 2*A*b^8*c*d*tan(f*x + e) - 8*C*a^7*b*d^2*tan(f*x + e
) + 22*B*a^6*b^2*d^2*tan(f*x + e) - 42*A*a^5*b^3*d^2*tan(f*x + e) + 18*C*a^
5*b^3*d^2*tan(f*x + e) - 2*B*a^4*b^4*d^2*tan(f*x + e) - 26*A*a^3*b^5*d^2*ta
n(f*x + e) + 2*C*a^3*b^5*d^2*tan(f*x + e) - 8*A*a*b^7*d^2*tan(f*x + e) - C*
a^6*b^2*c^2 + 6*B*a^5*b^3*c^2 - 14*A*a^4*b^4*c^2 + 11*C*a^4*b^4*c^2 - 7*B*a
^3*b^5*c^2 - 3*A*a^2*b^6*c^2 - B*a*b^7*c^2 - A*b^8*c^2 + 4*C*a^7*b*c*d - 17
*B*a^6*b^2*c*d + 36*A*a^5*b^3*c*d - 24*C*a^5*b^3*c*d + 10*B*a^4*b^4*c*d + 1
6*A*a^3*b^5*c*d - 4*C*a^3*b^5*c*d + 3*B*a^2*b^6*c*d + 4*A*a*b^7*c*d - 6*C*a
^8*d^2 + 14*B*a^7*b*d^2 - 25*A*a^6*b^2*d^2 + 7*C*a^6*b^2*d^2 + 3*B*a^5*b^3*
```

```
d^2 - 19*A*a^4*b^4*d^2 + C*a^4*b^4*d^2 + B*a^3*b^5*d^2 - 6*A*a^2*b^6*d^2)/(
(a^6*b^3*c^3 + 3*a^4*b^5*c^3 + 3*a^2*b^7*c^3 + b^9*c^3 - 3*a^7*b^2*c^2*d -
9*a^5*b^4*c^2*d - 9*a^3*b^6*c^2*d - 3*a*b^8*c^2*d + 3*a^8*b*c*d^2 + 9*a^6*b
^3*c*d^2 + 9*a^4*b^5*c*d^2 + 3*a^2*b^7*c*d^2 - a^9*d^3 - 3*a^7*b^2*d^3 - 3*
a^5*b^4*d^3 - a^3*b^6*d^3)*(b*tan(f*x + e) + a)^2))/f
```

maple [B] time = 0.52, size = 2298, normalized size = 4.82

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e)),x)
```

```
[Out] 1/2/f/(a^2+b^2)^3/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*a^3*c+1/2/f/(a^2+b^2)^3/(c
^2+d^2)*ln(1+tan(f*x+e)^2)*B*b^3*d+1/2/f/(a^2+b^2)^3/(c^2+d^2)*ln(1+tan(f*x
+e)^2)*a^3*C*d-1/2/f/(a^2+b^2)^3/(c^2+d^2)*ln(1+tan(f*x+e)^2)*C*b^3*c+1/f/(
a^2+b^2)^3/(c^2+d^2)*A*arctan(tan(f*x+e))*a^3*c+1/f/(a^2+b^2)^3/(c^2+d^2)*A
*arctan(tan(f*x+e))*b^3*d+1/f/(a^2+b^2)^3/(c^2+d^2)*B*arctan(tan(f*x+e))*a^
3*d-1/f/(a^2+b^2)^3/(c^2+d^2)*B*arctan(tan(f*x+e))*b^3*c-1/f/(a^2+b^2)^3/(c
^2+d^2)*C*arctan(tan(f*x+e))*a^3*c-1/f/(a^2+b^2)^3/(c^2+d^2)*C*arctan(tan(f
*x+e))*b^3*d-1/2/f/(a*d-b*c)/(a^2+b^2)/(a+b*tan(f*x+e))^2*B*a*b+6/f/(a*d-b*
c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^2*b^4*c*d-1/f/(a*d-b*c)^2/(a^2+b^2)
^2/(a+b*tan(f*x+e))*B*b^4*c+1/f/(a*d-b*c)^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*a^
4*C*d+1/2/f/(a^2+b^2)^3/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*b^3*c-1/f*d^3/(a*d-b
*c)^3/(c^2+d^2)*ln(c+d*tan(f*x+e))*B*c+1/f*d^2/(a*d-b*c)^3/(c^2+d^2)*ln(c+d
*tan(f*x+e))*c^2*C-1/2/f/(a^2+b^2)^3/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*a^3*d+1
/f/(a*d-b*c)^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*A*b^4*d+1/f/(a*d-b*c)^3/(a^2+b^
2)^3*ln(a+b*tan(f*x+e))*A*b^6*c^2-1/f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*
x+e))*A*b^6*d^2-1/f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*a^6*C*d^2-1/
f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*b^6*c^2+3/f/(a*d-b*c)^3/(a^2
+b^2)^3*ln(a+b*tan(f*x+e))*a^4*b^2*C*d^2+3/f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b
*tan(f*x+e))*C*a^2*b^4*c^2-6/f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A
*a^4*b^2*d^2+3/2/f/(a^2+b^2)^3/(c^2+d^2)*ln(1+tan(f*x+e)^2)*C*a^2*b*c-1/f/(
a*d-b*c)^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*C*a^2*b^2*d+2/f/(a*d-b*c)^2/(a^2+b^
2)^2/(a+b*tan(f*x+e))*C*a*b^3*c-3/f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+
e))*A*a^2*b^4*c^2-3/f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a^2*b^4*
d^2+3/f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*a^5*b*B*d^2+1/f/(a*d-b*c
)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^3*b^3*c^2-1/f/(a*d-b*c)^3/(a^2+b^2)^
3*ln(a+b*tan(f*x+e))*B*a^3*b^3*d^2-3/f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f
*x+e))*a*b^5*B*c^2-3/2/f/(a^2+b^2)^3/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*a*b^2*c
-3/2/f/(a^2+b^2)^3/(c^2+d^2)*ln(1+tan(f*x+e)^2)*B*a^2*b*d+3/f/(a^2+b^2)^3/(
c^2+d^2)*C*arctan(tan(f*x+e))*a^2*b*d+3/f/(a^2+b^2)^3/(c^2+d^2)*C*arctan(ta
n(f*x+e))*a*b^2*c-3/2/f/(a^2+b^2)^3/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*a^2*b*c+
3/2/f/(a^2+b^2)^3/(c^2+d^2)*ln(1+tan(f*x+e)^2)*A*a*b^2*d+1/f/(a*d-b*c)^3/(a
^2+b^2)^3*ln(a+b*tan(f*x+e))*B*b^6*c*d-3/f/(a^2+b^2)^3/(c^2+d^2)*B*arctan(t
```

```
an(f*x+e))*a*b^2*d-2/f/(a*d-b*c)^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*a^3*b*B*d+3
/f/(a*d-b*c)^2/(a^2+b^2)^2/(a+b*tan(f*x+e))*A*a^2*b^2*d-3/2/f/(a^2+b^2)^3/(
c^2+d^2)*ln(1+tan(f*x+e)^2)*C*a*b^2*d-3/f/(a^2+b^2)^3/(c^2+d^2)*A*arctan(ta
n(f*x+e))*a^2*b*d-3/f/(a^2+b^2)^3/(c^2+d^2)*A*arctan(tan(f*x+e))*a*b^2*c+3/
f/(a^2+b^2)^3/(c^2+d^2)*B*arctan(tan(f*x+e))*a^2*b*c+1/f/(a*d-b*c)^2/(a^2+b
^2)^2/(a+b*tan(f*x+e))*B*a^2*b^2*c-2/f/(a*d-b*c)^2/(a^2+b^2)^2/(a+b*tan(f*x
+e))*A*a*b^3*c-3/f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^4*b^2*c*d
+8/f/(a*d-b*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a^3*b^3*c*d+1/2/f/(a*d-b*
c)/(a^2+b^2)/(a+b*tan(f*x+e))^2*A*b^2+1/2/f/(a*d-b*c)/(a^2+b^2)/(a+b*tan(f*
x+e))^2*a^2*C+1/f*d^4/(a*d-b*c)^3/(c^2+d^2)*ln(c+d*tan(f*x+e))*A-8/f/(a*d-b
*c)^3/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*a^3*b^3*c*d
```

maxima [B] time = 0.68, size = 1096, normalized size = 2.30

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c + (B*a^3 - 3*
(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*d)*(f*x + e)/((a^6 + 3*a^4*b^2 + 3
*a^2*b^4 + b^6)*c^2 + (a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)*d^2) - 2*((B*a^3*
b^3 - 3*(A - C)*a^2*b^4 - 3*B*a*b^5 + (A - C)*b^6)*c^2 - (3*B*a^4*b^2 - 8*(
A - C)*a^3*b^3 - 6*B*a^2*b^4 - B*b^6)*c*d - (C*a^6 - 3*B*a^5*b + 3*(2*A - C
)*a^4*b^2 + B*a^3*b^3 + 3*A*a^2*b^4 + A*b^6)*d^2)*log(b*tan(f*x + e) + a)/(
(a^6*b^3 + 3*a^4*b^5 + 3*a^2*b^7 + b^9)*c^3 - 3*(a^7*b^2 + 3*a^5*b^4 + 3*a^
3*b^6 + a*b^8)*c^2*d + 3*(a^8*b + 3*a^6*b^3 + 3*a^4*b^5 + a^2*b^7)*c*d^2 -
(a^9 + 3*a^7*b^2 + 3*a^5*b^4 + a^3*b^6)*d^3) - 2*(C*c^2*d^2 - B*c*d^3 + A*d
^4)*log(d*tan(f*x + e) + c)/(b^3*c^5 - 3*a*b^2*c^4*d + 3*a^2*b*c*d^4 - a^3*
d^5 + (3*a^2*b + b^3)*c^3*d^2 - (a^3 + 3*a*b^2)*c^2*d^3) + ((B*a^3 - 3*(A -
 C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c - ((A - C)*a^3 + 3*B*a^2*b - 3*(A -
C)*a*b^2 - B*b^3)*d)*log(tan(f*x + e)^2 + 1)/((a^6 + 3*a^4*b^2 + 3*a^2*b^4
+ b^6)*c^2 + (a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)*d^2) - ((C*a^4*b - 3*B*a^3
*b^2 + (5*A - 3*C)*a^2*b^3 + B*a*b^4 + A*b^5)*c - (3*C*a^5 - 5*B*a^4*b + (7
*A - C)*a^3*b^2 - B*a^2*b^3 + 3*A*a*b^4)*d - 2*((B*a^2*b^3 - 2*(A - C)*a*b^
4 - B*b^5)*c + (C*a^4*b - 2*B*a^3*b^2 + (3*A - C)*a^2*b^3 + A*b^5)*d)*tan(f
*x + e))/((a^6*b^2 + 2*a^4*b^4 + a^2*b^6)*c^2 - 2*(a^7*b + 2*a^5*b^3 + a^3*
b^5)*c*d + (a^8 + 2*a^6*b^2 + a^4*b^4)*d^2 + ((a^4*b^4 + 2*a^2*b^6 + b^8)*c
^2 - 2*(a^5*b^3 + 2*a^3*b^5 + a*b^7)*c*d + (a^6*b^2 + 2*a^4*b^4 + a^2*b^6)*
d^2)*tan(f*x + e)^2 + 2*((a^5*b^3 + 2*a^3*b^5 + a*b^7)*c^2 - 2*(a^6*b^2 + 2
*a^4*b^4 + a^2*b^6)*c*d + (a^7*b + 2*a^5*b^3 + a^3*b^5)*d^2)*tan(f*x + e)))
/f
```

mupad [B] time = 24.03, size = 65819, normalized size = 137.99

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^3*(c + d*
tan(e + f*x))),x)
```

```
[Out] -(((A*b^5*c - 3*C*a^5*d - 3*A*a*b^4*d + B*a*b^4*c + 5*B*a^4*b*d + C*a^4*b*c
 + 5*A*a^2*b^3*c - 7*A*a^3*b^2*d - 3*B*a^3*b^2*c + B*a^2*b^3*d - 3*C*a^2*b^
3*c + C*a^3*b^2*d)/(2*(a^2*d^2 + b^2*c^2 - 2*a*b*c*d)*(a^4 + b^4 + 2*a^2*b^
2)) - (tan(e + f*x)*(A*b^5*d - B*b^5*c - 2*A*a*b^4*c + 2*C*a*b^4*c + C*a^4*
b*d + 3*A*a^2*b^3*d + B*a^2*b^3*c - 2*B*a^3*b^2*d - C*a^2*b^3*d))/((a^2*d^2
 + b^2*c^2 - 2*a*b*c*d)*(a^4 + b^4 + 2*a^2*b^2)))/(a^2 + b^2*tan(e + f*x)^2
 + 2*a*b*tan(e + f*x)) - symsum(log(- (A^3*b^8*c^2*d^4 - 4*A^3*a^2*b^6*d^6
- 7*A^3*a^4*b^4*d^6 - A^3*b^8*d^6 + A^2*C*b^8*d^6 - 3*A^3*a^2*b^6*c^2*d^4 -
 B^3*a^3*b^5*c^2*d^4 - C^3*a^2*b^6*c^2*d^4 - 2*C^3*a^3*b^5*c^3*d^3 + 7*C^3*
a^4*b^4*c^2*d^4 + A^2*B*a*b^7*d^6 + A^2*B*b^8*c*d^5 + A^3*a*b^7*c*d^5 + C^3
*a^7*b*c*d^5 - A*B^2*a^2*b^6*d^6 - 3*A*B^2*a^6*b^2*d^6 + 2*A^2*B*a^3*b^5*d^
6 + 9*A^2*B*a^5*b^3*d^6 - A*C^2*a^2*b^6*d^6 - 4*A*C^2*a^4*b^4*d^6 + A*C^2*a
^6*b^2*d^6 + 5*A^2*C*a^2*b^6*d^6 + 11*A^2*C*a^4*b^4*d^6 - A^2*C*a^6*b^2*d^6
 + A*C^2*b^8*c^2*d^4 - 2*A^2*C*b^8*c^2*d^4 - B*C^2*b^8*c^3*d^3 + B^2*C*b^8*
c^2*d^4 + 9*A^3*a^3*b^5*c*d^5 - B^3*a*b^7*c^2*d^4 + B^3*a^2*b^6*c*d^5 + B^3
*a^4*b^4*c*d^5 + 2*C^3*a*b^7*c^3*d^3 - 3*C^3*a^5*b^3*c*d^5 + A*B*C*a^7*b*d^
6 - 2*A*B*C*b^8*c*d^5 + 3*A*B^2*a^2*b^6*c^2*d^4 - A*B^2*a^4*b^4*c^2*d^4 + 3
*A^2*B*a^3*b^5*c^2*d^4 - A*C^2*a^2*b^6*c^2*d^4 + 4*A*C^2*a^3*b^5*c^3*d^3 -
14*A*C^2*a^4*b^4*c^2*d^4 + 5*A^2*C*a^2*b^6*c^2*d^4 - 2*A^2*C*a^3*b^5*c^3*d^
3 + 7*A^2*C*a^4*b^4*c^2*d^4 + 6*B*C^2*a^2*b^6*c^3*d^3 - 15*B*C^2*a^3*b^5*c^
2*d^4 - B*C^2*a^4*b^4*c^3*d^3 + 3*B*C^2*a^5*b^3*c^2*d^4 + 5*B^2*C*a^2*b^6*c
^2*d^4 + 2*B^2*C*a^3*b^5*c^3*d^3 - 4*B^2*C*a^4*b^4*c^2*d^4 + A*B*C*a^3*b^5*
d^6 - 6*A*B*C*a^5*b^3*d^6 + A*B*C*b^8*c^3*d^3 + 2*A*C^2*a*b^7*c*d^5 - A*C^2
*a^7*b*c*d^5 - 3*A^2*C*a*b^7*c*d^5 - 5*A*B^2*a^3*b^5*c*d^5 + 3*A*B^2*a^5*b^
3*c*d^5 - 5*A^2*B*a*b^7*c^2*d^4 + 7*A^2*B*a^2*b^6*c*d^5 - 10*A^2*B*a^4*b^4*
c*d^5 - 4*A*C^2*a*b^7*c^3*d^3 + 12*A*C^2*a^3*b^5*c*d^5 + 9*A*C^2*a^5*b^3*c*
d^5 + 2*A^2*C*a*b^7*c^3*d^3 - 21*A^2*C*a^3*b^5*c*d^5 - 6*A^2*C*a^5*b^3*c*d^
5 - 2*B*C^2*a*b^7*c^2*d^4 + B*C^2*a^2*b^6*c*d^5 + 5*B*C^2*a^4*b^4*c*d^5 - 4
*B*C^2*a^6*b^2*c*d^5 - 2*B^2*C*a*b^7*c^3*d^3 - B^2*C*a^3*b^5*c*d^5 + 3*B^2*
C*a^5*b^3*c*d^5 - 6*A*B*C*a^2*b^6*c^3*d^3 + 12*A*B*C*a^3*b^5*c^2*d^4 + A*B*
C*a^4*b^4*c^3*d^3 - 3*A*B*C*a^5*b^3*c^2*d^4 + 7*A*B*C*a*b^7*c^2*d^4 - 11*A*
B*C*a^2*b^6*c*d^5 + 2*A*B*C*a^4*b^4*c*d^5 + 3*A*B*C*a^6*b^2*c*d^5)/(a^12*d^
4 + b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4
 + a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9
*c*d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^
5*c*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^1
```

0*c^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d^2 +
6*a^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3) - root(480*a^11*b^7*c
*d^9*f^4 + 480*a^7*b^11*c^9*d*f^4 + 360*a^13*b^5*c*d^9*f^4 + 360*a^9*b^9*c^
9*d*f^4 + 360*a^9*b^9*c*d^9*f^4 + 360*a^5*b^13*c^9*d*f^4 + 144*a^15*b^3*c*d
^9*f^4 + 144*a^11*b^7*c^9*d*f^4 + 144*a^7*b^11*c*d^9*f^4 + 144*a^3*b^15*c^9
*d*f^4 + 48*a^17*b*c^3*d^7*f^4 + 48*a*b^17*c^7*d^3*f^4 + 24*a^17*b*c^5*d^5*
f^4 + 24*a^13*b^5*c^9*d*f^4 + 24*a^5*b^13*c*d^9*f^4 + 24*a*b^17*c^5*d^5*f^4
 + 24*a^17*b*c*d^9*f^4 + 24*a*b^17*c^9*d*f^4 + 3920*a^9*b^9*c^5*d^5*f^4 - 3
360*a^10*b^8*c^4*d^6*f^4 - 3360*a^8*b^10*c^6*d^4*f^4 + 3024*a^11*b^7*c^5*d^
5*f^4 - 3024*a^10*b^8*c^6*d^4*f^4 - 3024*a^8*b^10*c^4*d^6*f^4 + 3024*a^7*b^
11*c^5*d^5*f^4 + 2320*a^9*b^9*c^7*d^3*f^4 + 2320*a^9*b^9*c^3*d^7*f^4 - 2240
*a^12*b^6*c^4*d^6*f^4 - 2240*a^6*b^12*c^6*d^4*f^4 + 2160*a^11*b^7*c^3*d^7*f
^4 + 2160*a^7*b^11*c^7*d^3*f^4 - 1624*a^12*b^6*c^6*d^4*f^4 - 1624*a^6*b^12*
c^4*d^6*f^4 + 1488*a^11*b^7*c^7*d^3*f^4 + 1488*a^7*b^11*c^3*d^7*f^4 + 1344*
a^13*b^5*c^5*d^5*f^4 + 1344*a^5*b^13*c^5*d^5*f^4 - 1320*a^10*b^8*c^2*d^8*f^
4 - 1320*a^8*b^10*c^8*d^2*f^4 + 1200*a^13*b^5*c^3*d^7*f^4 + 1200*a^5*b^13*c
^7*d^3*f^4 - 1060*a^12*b^6*c^2*d^8*f^4 - 1060*a^6*b^12*c^8*d^2*f^4 - 948*a^
10*b^8*c^8*d^2*f^4 - 948*a^8*b^10*c^2*d^8*f^4 - 840*a^14*b^4*c^4*d^6*f^4 -
840*a^4*b^14*c^6*d^4*f^4 + 528*a^13*b^5*c^7*d^3*f^4 + 528*a^5*b^13*c^3*d^7*
f^4 - 480*a^14*b^4*c^6*d^4*f^4 - 480*a^14*b^4*c^2*d^8*f^4 - 480*a^4*b^14*c^
8*d^2*f^4 - 480*a^4*b^14*c^4*d^6*f^4 + 368*a^15*b^3*c^3*d^7*f^4 - 368*a^12*
b^6*c^8*d^2*f^4 - 368*a^6*b^12*c^2*d^8*f^4 + 368*a^3*b^15*c^7*d^3*f^4 + 304
*a^15*b^3*c^5*d^5*f^4 + 304*a^3*b^15*c^5*d^5*f^4 - 144*a^16*b^2*c^4*d^6*f^4
 - 144*a^2*b^16*c^6*d^4*f^4 - 108*a^16*b^2*c^2*d^8*f^4 - 108*a^2*b^16*c^8*d
^2*f^4 + 80*a^15*b^3*c^7*d^3*f^4 + 80*a^3*b^15*c^3*d^7*f^4 - 60*a^16*b^2*c^
6*d^4*f^4 - 60*a^14*b^4*c^8*d^2*f^4 - 60*a^4*b^14*c^2*d^8*f^4 - 60*a^2*b^16
*c^4*d^6*f^4 - 8*b^18*c^8*d^2*f^4 - 4*b^18*c^6*d^4*f^4 - 8*a^18*c^2*d^8*f^4
 - 4*a^18*c^4*d^6*f^4 - 80*a^12*b^6*d^10*f^4 - 60*a^14*b^4*d^10*f^4 - 60*a^
10*b^8*d^10*f^4 - 24*a^16*b^2*d^10*f^4 - 24*a^8*b^10*d^10*f^4 - 4*a^6*b^12*
d^10*f^4 - 80*a^6*b^12*c^10*f^4 - 60*a^8*b^10*c^10*f^4 - 60*a^4*b^14*c^10*f
^4 - 24*a^10*b^8*c^10*f^4 - 24*a^2*b^16*c^10*f^4 - 4*a^12*b^6*c^10*f^4 - 4*
b^18*c^10*f^4 - 4*a^18*d^10*f^4 - 12*A*C*a^11*b*c*d^7*f^2 - 12*A*C*a*b^11*c
^7*d*f^2 - 912*B*C*a^5*b^7*c^4*d^4*f^2 - 792*B*C*a^8*b^4*c^3*d^5*f^2 + 792*
B*C*a^4*b^8*c^5*d^3*f^2 + 720*B*C*a^7*b^5*c^4*d^4*f^2 - 480*B*C*a^5*b^7*c^6
*d^2*f^2 - 408*B*C*a^5*b^7*c^2*d^6*f^2 + 384*B*C*a^7*b^5*c^2*d^6*f^2 - 336*
B*C*a^8*b^4*c^5*d^3*f^2 + 324*B*C*a^4*b^8*c^3*d^5*f^2 + 312*B*C*a^7*b^5*c^6
*d^2*f^2 - 248*B*C*a^3*b^9*c^6*d^2*f^2 + 216*B*C*a^9*b^3*c^2*d^6*f^2 - 196*
B*C*a^3*b^9*c^4*d^4*f^2 + 132*B*C*a^9*b^3*c^4*d^4*f^2 + 80*B*C*a^6*b^6*c^3*
d^5*f^2 - 64*B*C*a^6*b^6*c^5*d^3*f^2 - 36*B*C*a^2*b^10*c^3*d^5*f^2 - 28*B*C
*a^3*b^9*c^2*d^6*f^2 + 12*B*C*a^10*b^2*c^5*d^3*f^2 - 12*B*C*a^10*b^2*c^3*d^
5*f^2 - 12*B*C*a^2*b^10*c^5*d^3*f^2 - 4*B*C*a^9*b^3*c^6*d^2*f^2 - 1468*A*C*
a^6*b^6*c^4*d^4*f^2 + 996*A*C*a^7*b^5*c^3*d^5*f^2 + 900*A*C*a^5*b^7*c^5*d^3
*f^2 - 676*A*C*a^6*b^6*c^6*d^2*f^2 - 660*A*C*a^6*b^6*c^2*d^6*f^2 + 636*A*C*
a^5*b^7*c^3*d^5*f^2 + 540*A*C*a^7*b^5*c^5*d^3*f^2 - 236*A*C*a^3*b^9*c^5*d^3
*f^2 - 204*A*C*a^9*b^3*c^3*d^5*f^2 + 156*A*C*a^10*b^2*c^2*d^6*f^2 + 132*A*C

*a^2*b^10*c^6*d^2*f^2 - 72*A*C*a^9*b^3*c^5*d^3*f^2 - 72*A*C*a^4*b^8*c^6*d^2
*f^2 + 66*A*C*a^4*b^8*c^2*d^6*f^2 + 54*A*C*a^10*b^2*c^4*d^4*f^2 + 54*A*C*a^
2*b^10*c^4*d^4*f^2 - 48*A*C*a^8*b^4*c^2*d^6*f^2 - 48*A*C*a^4*b^8*c^4*d^4*f^
2 + 42*A*C*a^8*b^4*c^6*d^2*f^2 - 40*A*C*a^3*b^9*c^3*d^5*f^2 - 36*A*C*a^8*b^
4*c^4*d^4*f^2 + 24*A*C*a^2*b^10*c^2*d^6*f^2 + 960*A*B*a^5*b^7*c^4*d^4*f^2 -
864*A*B*a^4*b^8*c^5*d^3*f^2 + 756*A*B*a^8*b^4*c^3*d^5*f^2 - 744*A*B*a^7*b^
5*c^4*d^4*f^2 - 528*A*B*a^4*b^8*c^3*d^5*f^2 + 504*A*B*a^5*b^7*c^6*d^2*f^2 -
432*A*B*a^7*b^5*c^2*d^6*f^2 + 432*A*B*a^5*b^7*c^2*d^6*f^2 + 348*A*B*a^8*b^
4*c^5*d^3*f^2 - 312*A*B*a^7*b^5*c^6*d^2*f^2 - 284*A*B*a^9*b^3*c^2*d^6*f^2 +
280*A*B*a^3*b^9*c^6*d^2*f^2 + 264*A*B*a^3*b^9*c^4*d^4*f^2 - 240*A*B*a^6*b^
6*c^3*d^5*f^2 - 172*A*B*a^9*b^3*c^4*d^4*f^2 + 68*A*B*a^3*b^9*c^2*d^6*f^2 -
60*A*B*a^2*b^10*c^3*d^5*f^2 + 24*A*B*a^6*b^6*c^5*d^3*f^2 - 24*A*B*a^2*b^10*
c^5*d^3*f^2 + 12*A*B*a^10*b^2*c^3*d^5*f^2 + 360*B*C*a^4*b^8*c^7*d*f^2 - 336
*B*C*a^8*b^4*c*d^7*f^2 + 168*B*C*a^6*b^6*c*d^7*f^2 - 136*B*C*a^6*b^6*c^7*d*
f^2 - 36*B*C*a^11*b*c^2*d^6*f^2 + 36*B*C*a*b^11*c^6*d^2*f^2 + 24*B*C*a^10*b
^2*c*d^7*f^2 - 24*B*C*a^2*b^10*c^7*d*f^2 - 12*B*C*a^11*b*c^4*d^4*f^2 + 12*B
*C*a^4*b^8*c*d^7*f^2 + 12*B*C*a*b^11*c^4*d^4*f^2 + 444*A*C*a^7*b^5*c*d^7*f^
2 + 348*A*C*a^5*b^7*c^7*d*f^2 - 164*A*C*a^3*b^9*c^7*d*f^2 - 132*A*C*a^9*b^3
*c*d^7*f^2 + 84*A*C*a^5*b^7*c*d^7*f^2 + 32*A*C*a^3*b^9*c*d^7*f^2 - 12*A*C*a
^11*b*c^3*d^5*f^2 - 12*A*C*a^7*b^5*c^7*d*f^2 - 12*A*C*a*b^11*c^5*d^3*f^2 -
360*A*B*a^4*b^8*c^7*d*f^2 + 288*A*B*a^8*b^4*c*d^7*f^2 - 288*A*B*a^6*b^6*c*d
^7*f^2 - 144*A*B*a^4*b^8*c*d^7*f^2 + 136*A*B*a^6*b^6*c^7*d*f^2 - 60*A*B*a^2
*b^10*c*d^7*f^2 - 36*A*B*a^10*b^2*c*d^7*f^2 + 24*A*B*a^2*b^10*c^7*d*f^2 - 2
4*A*B*a*b^11*c^6*d^2*f^2 + 12*A*B*a^11*b*c^2*d^6*f^2 + 12*A*B*a*b^11*c^4*d^
4*f^2 + 12*A*B*a*b^11*c^2*d^6*f^2 - 8*B*C*b^12*c^5*d^3*f^2 - 8*B*C*b^12*c^3
*d^5*f^2 + 8*A*C*b^12*c^2*d^6*f^2 - 4*B*C*a^12*c^3*d^5*f^2 + 4*A*C*b^12*c^4
*d^4*f^2 - 2*A*C*b^12*c^6*d^2*f^2 + 80*B*C*a^9*b^3*d^8*f^2 - 24*B*C*a^7*b^5
*d^8*f^2 + 6*A*C*a^12*c^2*d^6*f^2 + 4*A*B*b^12*c^5*d^3*f^2 - 4*A*B*b^12*c^3
*d^5*f^2 - 90*A*C*a^8*b^4*d^8*f^2 - 80*B*C*a^3*b^9*c^8*f^2 + 54*A*C*a^10*b^
2*d^8*f^2 - 30*A*C*a^6*b^6*d^8*f^2 + 24*B*C*a^5*b^7*c^8*f^2 - 12*A*C*a^4*b^
8*d^8*f^2 - 112*A*B*a^9*b^3*d^8*f^2 - 66*A*C*a^4*b^8*c^8*f^2 + 54*A*C*a^2*b
^10*c^8*f^2 + 4*A*B*a^3*b^9*d^8*f^2 + 2*A*C*a^6*b^6*c^8*f^2 + 80*A*B*a^3*b^
9*c^8*f^2 - 24*A*B*a^5*b^7*c^8*f^2 + 726*C^2*a^6*b^6*c^4*d^4*f^2 - 402*C^2*
a^7*b^5*c^3*d^5*f^2 - 402*C^2*a^5*b^7*c^5*d^3*f^2 + 322*C^2*a^6*b^6*c^6*d^2
*f^2 + 322*C^2*a^6*b^6*c^2*d^6*f^2 - 222*C^2*a^7*b^5*c^5*d^3*f^2 - 222*C^2*
a^5*b^7*c^3*d^5*f^2 + 134*C^2*a^9*b^3*c^3*d^5*f^2 + 134*C^2*a^3*b^9*c^5*d^3
*f^2 - 66*C^2*a^10*b^2*c^2*d^6*f^2 - 66*C^2*a^2*b^10*c^6*d^2*f^2 + 52*C^2*a
^9*b^3*c^5*d^3*f^2 + 52*C^2*a^3*b^9*c^3*d^5*f^2 - 27*C^2*a^8*b^4*c^6*d^2*f^
2 - 27*C^2*a^4*b^8*c^2*d^6*f^2 + 24*C^2*a^8*b^4*c^4*d^4*f^2 + 24*C^2*a^8*b^
4*c^2*d^6*f^2 + 24*C^2*a^4*b^8*c^6*d^2*f^2 + 24*C^2*a^4*b^8*c^4*d^4*f^2 - 1
5*C^2*a^10*b^2*c^4*d^4*f^2 - 15*C^2*a^2*b^10*c^4*d^4*f^2 - 570*B^2*a^6*b^6*
c^4*d^4*f^2 + 366*B^2*a^7*b^5*c^3*d^5*f^2 + 318*B^2*a^5*b^7*c^5*d^3*f^2 - 2
62*B^2*a^6*b^6*c^6*d^2*f^2 - 222*B^2*a^6*b^6*c^2*d^6*f^2 - 210*B^2*a^3*b^9*
c^5*d^3*f^2 + 186*B^2*a^7*b^5*c^5*d^3*f^2 + 162*B^2*a^5*b^7*c^3*d^5*f^2 - 1
42*B^2*a^9*b^3*c^3*d^5*f^2 + 132*B^2*a^4*b^8*c^4*d^4*f^2 + 117*B^2*a^4*b^8*

c^2*d^6*f^2 + 102*B^2*a^2*b^10*c^6*d^2*f^2 - 96*B^2*a^3*b^9*c^3*d^5*f^2 + 9
0*B^2*a^10*b^2*c^2*d^6*f^2 + 81*B^2*a^2*b^10*c^4*d^4*f^2 - 56*B^2*a^9*b^3*c
^5*d^3*f^2 + 48*B^2*a^8*b^4*c^4*d^4*f^2 + 48*B^2*a^4*b^8*c^6*d^2*f^2 + 45*B
^2*a^8*b^4*c^6*d^2*f^2 + 36*B^2*a^8*b^4*c^2*d^6*f^2 + 36*B^2*a^2*b^10*c^2*d
^6*f^2 + 33*B^2*a^10*b^2*c^4*d^4*f^2 + 822*A^2*a^6*b^6*c^4*d^4*f^2 - 594*A^
2*a^7*b^5*c^3*d^5*f^2 + 498*A^2*a^6*b^6*c^2*d^6*f^2 - 498*A^2*a^5*b^7*c^5*d
^3*f^2 - 414*A^2*a^5*b^7*c^3*d^5*f^2 + 354*A^2*a^6*b^6*c^6*d^2*f^2 - 318*A^
2*a^7*b^5*c^5*d^3*f^2 + 144*A^2*a^8*b^4*c^2*d^6*f^2 + 102*A^2*a^3*b^9*c^5*d
^3*f^2 + 84*A^2*a^4*b^8*c^4*d^4*f^2 + 81*A^2*a^4*b^8*c^2*d^6*f^2 + 72*A^2*a
^8*b^4*c^4*d^4*f^2 + 70*A^2*a^9*b^3*c^3*d^5*f^2 - 66*A^2*a^2*b^10*c^6*d^2*f
^2 + 48*A^2*a^4*b^8*c^6*d^2*f^2 - 42*A^2*a^10*b^2*c^2*d^6*f^2 + 24*A^2*a^2*
b^10*c^2*d^6*f^2 + 20*A^2*a^9*b^3*c^5*d^3*f^2 - 15*A^2*a^10*b^2*c^4*d^4*f^2
 - 15*A^2*a^8*b^4*c^6*d^2*f^2 - 15*A^2*a^2*b^10*c^4*d^4*f^2 - 12*A^2*a^3*b^
9*c^3*d^5*f^2 - 8*B*C*b^12*c^7*d*f^2 + 4*B*C*a^12*c*d^7*f^2 - 24*B*C*a^11*b
*d^8*f^2 + 8*A*B*b^12*c^7*d*f^2 - 8*A*B*b^12*c*d^7*f^2 + 24*B*C*a*b^11*c^8*
f^2 - 8*A*B*a^12*c*d^7*f^2 + 12*A*B*a^11*b*d^8*f^2 - 24*A*B*a*b^11*c^8*f^2
- 174*C^2*a^7*b^5*c*d^7*f^2 - 174*C^2*a^5*b^7*c^7*d*f^2 + 82*C^2*a^9*b^3*c*
d^7*f^2 + 82*C^2*a^3*b^9*c^7*d*f^2 + 6*C^2*a^11*b*c^3*d^5*f^2 + 6*C^2*a^7*b
^5*c^7*d*f^2 + 6*C^2*a^5*b^7*c*d^7*f^2 + 6*C^2*a*b^11*c^5*d^3*f^2 + 162*B^2
*a^7*b^5*c*d^7*f^2 + 138*B^2*a^5*b^7*c^7*d*f^2 - 118*B^2*a^3*b^9*c^7*d*f^2
- 86*B^2*a^9*b^3*c*d^7*f^2 - 30*B^2*a*b^11*c^5*d^3*f^2 - 18*B^2*a^7*b^5*c^7
*d*f^2 - 18*B^2*a^5*b^7*c*d^7*f^2 - 12*B^2*a*b^11*c^3*d^5*f^2 - 6*B^2*a^11*
b*c^3*d^5*f^2 - 4*B^2*a^3*b^9*c*d^7*f^2 - 270*A^2*a^7*b^5*c*d^7*f^2 - 174*A
^2*a^5*b^7*c^7*d*f^2 - 90*A^2*a^5*b^7*c*d^7*f^2 + 82*A^2*a^3*b^9*c^7*d*f^2
+ 50*A^2*a^9*b^3*c*d^7*f^2 - 32*A^2*a^3*b^9*c*d^7*f^2 + 6*A^2*a^11*b*c^3*d^
5*f^2 + 6*A^2*a^7*b^5*c^7*d*f^2 + 6*A^2*a*b^11*c^5*d^3*f^2 + 6*C^2*a^11*b*c
*d^7*f^2 + 6*C^2*a*b^11*c^7*d*f^2 - 18*B^2*a*b^11*c^7*d*f^2 - 6*B^2*a^11*b*
c*d^7*f^2 + 6*A^2*a^11*b*c*d^7*f^2 + 6*A^2*a*b^11*c^7*d*f^2 - 6*A*C*b^12*c^
8*f^2 - 2*A*C*a^12*d^8*f^2 + 4*C^2*b^12*c^4*d^4*f^2 + 3*C^2*b^12*c^6*d^2*f^
2 + 4*C^2*a^12*c^4*d^4*f^2 + 4*B^2*b^12*c^4*d^4*f^2 + 4*B^2*b^12*c^2*d^6*f^
2 + 3*C^2*a^12*c^2*d^6*f^2 + 3*B^2*b^12*c^6*d^2*f^2 + 33*C^2*a^8*b^4*d^8*f^
2 - 27*C^2*a^10*b^2*d^8*f^2 - 4*A^2*b^12*c^4*d^4*f^2 + 3*B^2*a^12*c^2*d^6*f
^2 - C^2*a^6*b^6*d^8*f^2 - A^2*b^12*c^6*d^2*f^2 + 33*C^2*a^4*b^8*c^8*f^2 +
33*B^2*a^10*b^2*d^8*f^2 - 27*C^2*a^2*b^10*c^8*f^2 - 27*B^2*a^8*b^4*d^8*f^2
+ 3*B^2*a^6*b^6*d^8*f^2 - C^2*a^6*b^6*c^8*f^2 - A^2*a^12*c^2*d^6*f^2 + 117*
A^2*a^8*b^4*d^8*f^2 + 111*A^2*a^6*b^6*d^8*f^2 + 72*A^2*a^4*b^8*d^8*f^2 + 33
*B^2*a^2*b^10*c^8*f^2 - 27*B^2*a^4*b^8*c^8*f^2 + 24*A^2*a^2*b^10*d^8*f^2 +
3*B^2*a^6*b^6*c^8*f^2 - 3*A^2*a^10*b^2*d^8*f^2 + 33*A^2*a^4*b^8*c^8*f^2 - 2
7*A^2*a^2*b^10*c^8*f^2 - A^2*a^6*b^6*c^8*f^2 + 3*C^2*b^12*c^8*f^2 + 3*C^2*a
^12*d^8*f^2 + 4*A^2*b^12*d^8*f^2 - B^2*b^12*c^8*f^2 - B^2*a^12*d^8*f^2 + 3*
A^2*b^12*c^8*f^2 + 3*A^2*a^12*d^8*f^2 - 24*A*B*C*a*b^8*c*d^6*f + 342*A*B*C*
a^4*b^5*c^2*d^5*f - 186*A*B*C*a^5*b^4*c^3*d^4*f - 66*A*B*C*a^2*b^7*c^4*d^3*
f + 48*A*B*C*a^2*b^7*c^2*d^5*f + 42*A*B*C*a^6*b^3*c^2*d^5*f + 26*A*B*C*a^3*
b^6*c^5*d^2*f + 24*A*B*C*a^6*b^3*c^4*d^3*f - 18*A*B*C*a^7*b^2*c^3*d^4*f - 1
8*A*B*C*a^4*b^5*c^4*d^3*f - 8*A*B*C*a^3*b^6*c^3*d^4*f + 6*A*B*C*a^5*b^4*c^5

*d^2*f - 128*A*B*C*a^3*b^6*c*d^6*f + 126*A*B*C*a^7*b^2*c*d^6*f + 72*A*B*C*a
*b^8*c^3*d^4*f - 36*A*B*C*a^8*b*c^2*d^5*f - 36*A*B*C*a*b^8*c^5*d^2*f + 30*A
*B*C*a^2*b^7*c^6*d*f - 12*A*B*C*a^5*b^4*c*d^6*f - 12*A*B*C*a^4*b^5*c^6*d*f
- 21*B^2*C*a^8*b*c*d^6*f - 3*B^2*C*a*b^8*c^6*d*f + 21*A^2*C*a^8*b*c*d^6*f -
 21*A*C^2*a^8*b*c*d^6*f - 9*A^2*C*a*b^8*c^6*d*f + 9*A*C^2*a*b^8*c^6*d*f + 3
6*A^2*B*a*b^8*c*d^6*f + 21*A*B^2*a^8*b*c*d^6*f + 3*A*B^2*a*b^8*c^6*d*f + 16
*A*B*C*b^9*c^4*d^3*f - 16*A*B*C*b^9*c^2*d^5*f - 78*A*B*C*a^6*b^3*d^7*f + 24
*A*B*C*a^4*b^5*d^7*f + 2*A*B*C*a^3*b^6*c^7*f - 237*B^2*C*a^4*b^5*c^3*d^4*f
+ 165*B*C^2*a^5*b^4*c^3*d^4*f + 92*B^2*C*a^3*b^6*c^2*d^5*f - 81*B^2*C*a^7*b
^2*c^2*d^5*f + 77*B^2*C*a^3*b^6*c^4*d^3*f - 75*B*C^2*a^4*b^5*c^2*d^5*f + 69
*B^2*C*a^5*b^4*c^4*d^3*f + 69*B*C^2*a^4*b^5*c^4*d^3*f - 68*B*C^2*a^3*b^6*c^
3*d^4*f - 63*B^2*C*a^4*b^5*c^5*d^2*f - 61*B*C^2*a^6*b^3*c^2*d^5*f + 57*B*C^
2*a^2*b^7*c^4*d^3*f - 53*B*C^2*a^3*b^6*c^5*d^2*f - 44*B*C^2*a^6*b^3*c^4*d^3
*f - 36*B^2*C*a^2*b^7*c^3*d^4*f + 35*B^2*C*a^6*b^3*c^3*d^4*f - 33*B^2*C*a^5
*b^4*c^2*d^5*f + 33*B^2*C*a^2*b^7*c^5*d^2*f + 33*B*C^2*a^7*b^2*c^3*d^4*f -
12*B^2*C*a^7*b^2*c^4*d^3*f + 9*B*C^2*a^5*b^4*c^5*d^2*f + 4*B^2*C*a^6*b^3*c^
5*d^2*f + 225*A^2*C*a^5*b^4*c^2*d^5*f - 105*A*C^2*a^5*b^4*c^2*d^5*f - 99*A^
2*C*a^4*b^5*c^3*d^4*f - 81*A^2*C*a^4*b^5*c^5*d^2*f + 67*A^2*C*a^3*b^6*c^4*d
^3*f - 59*A*C^2*a^3*b^6*c^4*d^3*f - 57*A*C^2*a^7*b^2*c^2*d^5*f + 57*A*C^2*a
^2*b^7*c^5*d^2*f + 51*A^2*C*a^5*b^4*c^4*d^3*f + 48*A^2*C*a^2*b^7*c^3*d^4*f
+ 45*A*C^2*a^4*b^5*c^5*d^2*f - 35*A^2*C*a^6*b^3*c^3*d^4*f + 33*A^2*C*a^7*b^
2*c^2*d^5*f - 33*A^2*C*a^2*b^7*c^5*d^2*f + 33*A*C^2*a^5*b^4*c^4*d^3*f + 27*
A*C^2*a^6*b^3*c^3*d^4*f + 24*A*C^2*a^3*b^6*c^2*d^5*f - 24*A*C^2*a^2*b^7*c^3
*d^4*f - 21*A*C^2*a^4*b^5*c^3*d^4*f - 16*A^2*C*a^3*b^6*c^2*d^5*f - 243*A^2*
B*a^4*b^5*c^2*d^5*f - 156*A*B^2*a^3*b^6*c^2*d^5*f + 141*A*B^2*a^4*b^5*c^3*d
^4*f + 108*A^2*B*a^3*b^6*c^3*d^4*f - 105*A*B^2*a^3*b^6*c^4*d^3*f + 84*A*B^2
*a^2*b^7*c^3*d^4*f + 81*A*B^2*a^5*b^4*c^2*d^5*f + 51*A^2*B*a^6*b^3*c^2*d^5*
f - 51*A^2*B*a^4*b^5*c^4*d^3*f - 48*A^2*B*a^2*b^7*c^2*d^5*f + 45*A^2*B*a^5*
b^4*c^3*d^4*f + 39*A*B^2*a^4*b^5*c^5*d^2*f - 35*A*B^2*a^6*b^3*c^3*d^4*f + 3
3*A*B^2*a^7*b^2*c^2*d^5*f + 27*A^2*B*a^3*b^6*c^5*d^2*f - 21*A*B^2*a^5*b^4*c
^4*d^3*f + 20*A^2*B*a^6*b^3*c^4*d^3*f - 15*A^2*B*a^7*b^2*c^3*d^4*f - 15*A^2
*B*a^5*b^4*c^5*d^2*f + 9*A^2*B*a^2*b^7*c^4*d^3*f + 3*A*B^2*a^2*b^7*c^5*d^2*
f + 2*A*B*C*b^9*c^6*d*f - 6*A*B*C*a^9*c*d^6*f + 18*A*B*C*a^8*b*d^7*f - 6*A*
B*C*a*b^8*c^7*f + 63*B^2*C*a^6*b^3*c*d^6*f - 48*B^2*C*a*b^8*c^4*d^3*f + 42*
B*C^2*a^8*b*c^2*d^5*f + 42*B*C^2*a^5*b^4*c*d^6*f - 39*B*C^2*a^7*b^2*c*d^6*f
 + 30*B*C^2*a*b^8*c^5*d^2*f - 24*B^2*C*a^4*b^5*c*d^6*f - 24*B*C^2*a*b^8*c^3
*d^4*f + 17*B^2*C*a^3*b^6*c^6*d*f - 15*B*C^2*a^2*b^7*c^6*d*f + 12*B^2*C*a^8
*b*c^3*d^4*f + 12*B^2*C*a*b^8*c^2*d^5*f + 6*B*C^2*a^4*b^5*c^6*d*f - 192*A^2
*C*a^4*b^5*c*d^6*f - 99*A^2*C*a^6*b^3*c*d^6*f + 84*A*C^2*a^4*b^5*c*d^6*f +
59*A*C^2*a^6*b^3*c*d^6*f + 51*A^2*C*a^3*b^6*c^6*d*f - 51*A*C^2*a^3*b^6*c^6*
d*f - 36*A^2*C*a*b^8*c^2*d^5*f - 24*A*C^2*a*b^8*c^4*d^3*f + 24*A*C^2*a*b^8*
c^2*d^5*f + 12*A^2*C*a*b^8*c^4*d^3*f + 12*A*C^2*a^8*b*c^3*d^4*f + 160*A^2*B
*a^3*b^6*c*d^6*f - 99*A*B^2*a^6*b^3*c*d^6*f - 87*A^2*B*a^7*b^2*c*d^6*f - 72
*A*B^2*a^4*b^5*c*d^6*f - 48*A*B^2*a*b^8*c^2*d^5*f - 36*A^2*B*a*b^8*c^3*d^4*
f + 24*A*B^2*a*b^8*c^4*d^3*f - 17*A*B^2*a^3*b^6*c^6*d*f - 15*A^2*B*a^2*b^7*

c^6*d*f + 12*A*B^2*a^2*b^7*c*d^6*f + 6*A^2*B*a^8*b*c^2*d^5*f - 6*A^2*B*a^5*
b^4*c*d^6*f + 6*A^2*B*a^4*b^5*c^6*d*f + 6*A^2*B*a*b^8*c^5*d^2*f + 12*B^2*C*
b^9*c^3*d^4*f - 12*B*C^2*b^9*c^4*d^3*f - 12*A^2*C*b^9*c^3*d^4*f - 8*A*C^2*b
^9*c^5*d^2*f + 8*A*C^2*b^9*c^3*d^4*f + 4*B^2*C*a^9*c^2*d^5*f + 4*A^2*C*b^9*
c^5*d^2*f - 4*B*C^2*a^9*c^3*d^4*f + 12*A^2*B*b^9*c^2*d^5*f - 8*A*B^2*b^9*c^
3*d^4*f - 4*A^2*B*b^9*c^4*d^3*f + 4*A*C^2*a^9*c^2*d^5*f + 3*B^2*C*a^7*b^2*d
^7*f - B*C^2*a^6*b^3*d^7*f + 96*A^2*C*a^5*b^4*d^7*f - 39*A^2*C*a^7*b^2*d^7*
f - 36*A*C^2*a^5*b^4*d^7*f + 32*A^2*C*a^3*b^6*d^7*f + 15*A*C^2*a^7*b^2*d^7*
f - 3*B^2*C*a^2*b^7*c^7*f - B*C^2*a^3*b^6*c^7*f + 111*A^2*B*a^6*b^3*d^7*f -
39*A*B^2*a^7*b^2*d^7*f + 24*A*B^2*a^5*b^4*d^7*f - 9*A^2*C*a^2*b^7*c^7*f +
9*A*C^2*a^2*b^7*c^7*f - 4*A*B^2*a^3*b^6*d^7*f + 3*A*B^2*a^2*b^7*c^7*f - A^2
*B*a^3*b^6*c^7*f + 3*C^3*a^8*b*c*d^6*f - 3*C^3*a*b^8*c^6*d*f - 3*A^3*a^8*b*
c*d^6*f + 3*A^3*a*b^8*c^6*d*f - B*C^2*b^9*c^6*d*f + 4*A^2*C*b^9*c*d^6*f + 3
*B*C^2*a^9*c*d^6*f + 8*A*B^2*b^9*c*d^6*f + 3*B*C^2*a^8*b*d^7*f - A^2*B*b^9*
c^6*d*f + 12*A^2*C*a*b^8*d^7*f + 3*B*C^2*a*b^8*c^7*f - A^2*B*a^9*c*d^6*f -
9*A^2*B*a^8*b*d^7*f + 3*A^2*B*a*b^8*c^7*f - 39*C^3*a^5*b^4*c^4*d^3*f + 39*C
^3*a^4*b^5*c^3*d^4*f + 27*C^3*a^7*b^2*c^2*d^5*f - 27*C^3*a^2*b^7*c^5*d^2*f
- 17*C^3*a^6*b^3*c^3*d^4*f + 17*C^3*a^3*b^6*c^4*d^3*f + 3*C^3*a^5*b^4*c^2*d
^5*f - 3*C^3*a^4*b^5*c^5*d^2*f - 63*B^3*a^5*b^4*c^3*d^4*f + 57*B^3*a^4*b^5*
c^2*d^5*f - 51*B^3*a^2*b^7*c^4*d^3*f + 48*B^3*a^3*b^6*c^3*d^4*f + 31*B^3*a^
6*b^3*c^2*d^5*f + 27*B^3*a^3*b^6*c^5*d^2*f + 16*B^3*a^6*b^3*c^4*d^3*f - 15*
B^3*a^5*b^4*c^5*d^2*f - 12*B^3*a^2*b^7*c^2*d^5*f + 9*B^3*a^4*b^5*c^4*d^3*f
- 3*B^3*a^7*b^2*c^3*d^4*f - 123*A^3*a^5*b^4*c^2*d^5*f + 81*A^3*a^4*b^5*c^3*
d^4*f - 45*A^3*a^5*b^4*c^4*d^3*f + 39*A^3*a^4*b^5*c^5*d^2*f + 25*A^3*a^6*b^
3*c^3*d^4*f - 25*A^3*a^3*b^6*c^4*d^3*f - 24*A^3*a^2*b^7*c^3*d^4*f - 8*A^3*a
^3*b^6*c^2*d^5*f - 3*A^3*a^7*b^2*c^2*d^5*f + 3*A^3*a^2*b^7*c^5*d^2*f - 17*C
^3*a^6*b^3*c*d^6*f + 17*C^3*a^3*b^6*c^6*d*f - 12*C^3*a^8*b*c^3*d^4*f + 12*C
^3*a*b^8*c^4*d^3*f + 24*B^3*a*b^8*c^3*d^4*f + 21*B^3*a^7*b^2*c*d^6*f - 18*B
^3*a^5*b^4*c*d^6*f - 15*B^3*a^2*b^7*c^6*d*f - 6*B^3*a^8*b*c^2*d^5*f + 6*B^3
*a^4*b^5*c^6*d*f + 6*B^3*a*b^8*c^5*d^2*f + 4*B^3*a^3*b^6*c*d^6*f + 108*A^3*
a^4*b^5*c*d^6*f + 57*A^3*a^6*b^3*c*d^6*f - 17*A^3*a^3*b^6*c^6*d*f + 12*A^3*
a*b^8*c^2*d^5*f + 4*C^3*b^9*c^5*d^2*f - 4*C^3*a^9*c^2*d^5*f - 4*B^3*b^9*c^2
*d^5*f + 4*A^3*b^9*c^3*d^4*f + 3*C^3*a^7*b^2*d^7*f - 3*C^3*a^2*b^7*c^7*f -
B^3*a^6*b^3*d^7*f - 60*A^3*a^5*b^4*d^7*f - 32*A^3*a^3*b^6*d^7*f + 21*A^3*a^
7*b^2*d^7*f - B^3*a^3*b^6*c^7*f + 3*A^3*a^2*b^7*c^7*f - B^3*b^9*c^6*d*f - 4
*A^3*b^9*c*d^6*f - B^3*a^9*c*d^6*f + 3*B^3*a^8*b*d^7*f - 12*A^3*a*b^8*d^7*f
+ 3*B^3*a*b^8*c^7*f - B^2*C*a^9*d^7*f - 4*A^2*B*b^9*d^7*f + 3*A^2*C*b^9*c^
7*f - 3*A*C^2*b^9*c^7*f - A*C^2*a^9*d^7*f - A*B^2*b^9*c^7*f - C^3*a^9*d^7*f
- A^3*b^9*c^7*f + B^2*C*b^9*c^7*f + A^2*C*a^9*d^7*f + A*B^2*a^9*d^7*f + C^
3*b^9*c^7*f + A^3*a^9*d^7*f - 6*A*B^2*C*a^5*b*c*d^5 - 21*A^2*B*C*a^3*b^3*c^
2*d^4 + 21*A*B*C^2*a^3*b^3*c^2*d^4 + 12*A*B^2*C*a^4*b^2*c^2*d^4 - 12*A*B^2*
C*a^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^4*b^2*c^3*d^3
+ 3*A^2*B*C*a^4*b^2*c^3*d^3 + 3*A^2*B*C*a^2*b^4*c^3*d^3 + 3*A*B^2*C*a^2*b^4
*c^4*d^2 + 3*A*B*C^2*a^2*b^4*c^3*d^3 + 2*A*B*C^2*a^3*b^3*c^4*d^2 - A^2*B*C*
a^3*b^3*c^4*d^2 + 18*A^2*B*C*a^2*b^4*c*d^5 + 10*A*B^2*C*a^3*b^3*c*d^5 + 9*A

^2*B*C*a^4*b^2*c*d^5 - 9*A*B*C^2*a^4*b^2*c*d^5 - 9*A*B*C^2*a^2*b^4*c*d^5 -
6*A^2*B*C*a*b^5*c^2*d^4 + 6*A*B^2*C*a*b^5*c^3*d^3 + 6*A*B*C^2*a^5*b*c^2*d^4
 - 6*A*B*C^2*a*b^5*c^4*d^2 - 3*A^2*B*C*a^5*b*c^2*d^4 + 3*A^2*B*C*a*b^5*c^4*
d^2 + 3*A*B*C^2*a*b^5*c^2*d^4 - 3*B^3*C*a^5*b*c^2*d^4 + 3*B^3*C*a^4*b^2*c*d
^5 + 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a^5*b*c*d^5 - 3*B*C^3*a^5*b*c^2*d^4
+ 3*B*C^3*a^4*b^2*c*d^5 + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a^3*b^3*c*d^5 +
8*A*C^3*a^3*b^3*c*d^5 - 9*A^3*B*a^2*b^4*c*d^5 - 9*A*B^3*a^2*b^4*c*d^5 - 3*A
^3*B*a^4*b^2*c*d^5 + 3*A^3*B*a*b^5*c^2*d^4 + 3*A^2*B^2*a^5*b*c*d^5 - 3*A*B^
3*a^4*b^2*c*d^5 + 3*A*B^3*a*b^5*c^2*d^4 + 5*A*B*C^2*b^6*c^3*d^3 - 4*A^2*B*C
*b^6*c^3*d^3 - A*B^2*C*b^6*c^4*d^2 - 3*A*B^2*C*a^4*b^2*d^6 - 2*A^2*B*C*a^3*
b^3*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^4*b^2*c^2*d^4 + 6*B^2*C^2
*a^2*b^4*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 24*A^2*C^2*a^3*b^3*c^3*d^3 -
 15*A^2*C^2*a^4*b^2*c^2*d^4 - 9*A^2*C^2*a^2*b^4*c^4*d^2 + 3*A^2*C^2*a^2*b^4
*c^2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^4*b^2*c^2*d^4 + 4*A^2*B*
C*b^6*c*d^5 - 2*A*B*C^2*b^6*c*d^5 + 2*A*B*C^2*a^6*c*d^5 - A^2*B*C*a^6*c*d^5
 + 6*A^2*B*C*a^5*b*d^6 - 3*A*B*C^2*a^5*b*d^6 - 7*B^3*C*a^3*b^3*c^2*d^4 - 7*
B*C^3*a^3*b^3*c^2*d^4 + 3*B^3*C*a^4*b^2*c^3*d^3 - 3*B^3*C*a^2*b^4*c^3*d^3 -
 3*B^2*C^2*a*b^5*c^3*d^3 + 3*B*C^3*a^4*b^2*c^3*d^3 - 3*B*C^3*a^2*b^4*c^3*d^
3 - B^3*C*a^3*b^3*c^4*d^2 - B^2*C^2*a^3*b^3*c*d^5 - B*C^3*a^3*b^3*c^4*d^2 -
 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^4*b^2*c^2
*d^4 + 9*A*C^3*a^2*b^4*c^4*d^2 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^4*b^2*
c^2*d^4 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^2*b^4*c^4*d^2 - 9*A^2*B^2*a^3
*b^3*c*d^5 + 7*A^3*B*a^3*b^3*c^2*d^4 + 7*A*B^3*a^3*b^3*c^2*d^4 - 3*A^3*B*a^
2*b^4*c^3*d^3 - 3*A^2*B^2*a*b^5*c^3*d^3 - 3*A*B^3*a^2*b^4*c^3*d^3 - 5*A^2*C
^2*b^6*c^2*d^4 + 3*A^2*C^2*b^6*c^4*d^2 + 12*A^2*C^2*a^4*b^2*d^6 + 3*A^2*C^2
*a^2*b^4*d^6 + 6*A^2*B^2*a^4*b^2*d^6 + 3*A^2*B^2*a^2*b^4*d^6 + A*B*C^2*a^3*
b^3*d^6 - 3*B^4*a*b^5*c^3*d^3 - B^4*a^3*b^3*c*d^5 + A^2*B^2*a^3*b^3*c^3*d^3
 - 8*A^4*a^3*b^3*c*d^5 - 2*B^3*C*b^6*c^3*d^3 - 2*B*C^3*b^6*c^3*d^3 + 4*A^3*
C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 + 2*A*C^3*b^6*c^2*d^4 - A^3*C*b^6*c^4*d
^2 - 2*A*C^3*a^6*c^2*d^4 - 15*A^3*C*a^4*b^2*d^6 - 6*A^3*C*a^2*b^4*d^6 - 3*A
*C^3*a^4*b^2*d^6 + 3*B^4*a^5*b*c*d^5 - B^3*C*a^6*c*d^5 - B*C^3*a^6*c*d^5 -
2*A^3*B*b^6*c*d^5 - 2*A*B^3*b^6*c*d^5 - 3*A^3*B*a^5*b*d^6 - 3*A*B^3*a^5*b*d
^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2
+ 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^4*b^2*c^2*d^4 + 3*A^4*a^2*b^4*c^2*d^4 + B
^2*C^2*b^6*c^4*d^2 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^6*c^2*d^4 + A^2*C^2*a^
6*c^2*d^4 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B
*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 - A^4*b^6*c^2*d^4 + 6*A^4*a^4*b^2*d^6 + 3*
A^4*a^2*b^4*d^6 - 2*A^2*C^2*a^6*d^6 + A*B^2*C*a^6*d^6 + B^4*a^3*b^3*c^3*d^3
 + A^3*C*a^6*d^6 + A*C^3*a^6*d^6 + C^4*b^6*c^4*d^2 + C^4*a^6*c^2*d^4 + B^4*
b^6*c^2*d^4 + A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k)*(root(
480*a^11*b^7*c*d^9*f^4 + 480*a^7*b^11*c^9*d*f^4 + 360*a^13*b^5*c*d^9*f^4 +
360*a^9*b^9*c^9*d*f^4 + 360*a^9*b^9*c*d^9*f^4 + 360*a^5*b^13*c^9*d*f^4 + 14
4*a^15*b^3*c*d^9*f^4 + 144*a^11*b^7*c^9*d*f^4 + 144*a^7*b^11*c*d^9*f^4 + 14
4*a^3*b^15*c^9*d*f^4 + 48*a^17*b*c^3*d^7*f^4 + 48*a*b^17*c^7*d^3*f^4 + 24*a
^17*b*c^5*d^5*f^4 + 24*a^13*b^5*c^9*d*f^4 + 24*a^5*b^13*c*d^9*f^4 + 24*a*b^

17*c^5*d^5*f^4 + 24*a^17*b*c*d^9*f^4 + 24*a*b^17*c^9*d*f^4 + 3920*a^9*b^9*c
^5*d^5*f^4 - 3360*a^10*b^8*c^4*d^6*f^4 - 3360*a^8*b^10*c^6*d^4*f^4 + 3024*a
^11*b^7*c^5*d^5*f^4 - 3024*a^10*b^8*c^6*d^4*f^4 - 3024*a^8*b^10*c^4*d^6*f^4
 + 3024*a^7*b^11*c^5*d^5*f^4 + 2320*a^9*b^9*c^7*d^3*f^4 + 2320*a^9*b^9*c^3*
d^7*f^4 - 2240*a^12*b^6*c^4*d^6*f^4 - 2240*a^6*b^12*c^6*d^4*f^4 + 2160*a^11
*b^7*c^3*d^7*f^4 + 2160*a^7*b^11*c^7*d^3*f^4 - 1624*a^12*b^6*c^6*d^4*f^4 -
1624*a^6*b^12*c^4*d^6*f^4 + 1488*a^11*b^7*c^7*d^3*f^4 + 1488*a^7*b^11*c^3*d
^7*f^4 + 1344*a^13*b^5*c^5*d^5*f^4 + 1344*a^5*b^13*c^5*d^5*f^4 - 1320*a^10*
b^8*c^2*d^8*f^4 - 1320*a^8*b^10*c^8*d^2*f^4 + 1200*a^13*b^5*c^3*d^7*f^4 + 1
200*a^5*b^13*c^7*d^3*f^4 - 1060*a^12*b^6*c^2*d^8*f^4 - 1060*a^6*b^12*c^8*d^
2*f^4 - 948*a^10*b^8*c^8*d^2*f^4 - 948*a^8*b^10*c^2*d^8*f^4 - 840*a^14*b^4*
c^4*d^6*f^4 - 840*a^4*b^14*c^6*d^4*f^4 + 528*a^13*b^5*c^7*d^3*f^4 + 528*a^5
*b^13*c^3*d^7*f^4 - 480*a^14*b^4*c^6*d^4*f^4 - 480*a^14*b^4*c^2*d^8*f^4 - 4
80*a^4*b^14*c^8*d^2*f^4 - 480*a^4*b^14*c^4*d^6*f^4 + 368*a^15*b^3*c^3*d^7*f
^4 - 368*a^12*b^6*c^8*d^2*f^4 - 368*a^6*b^12*c^2*d^8*f^4 + 368*a^3*b^15*c^7
*d^3*f^4 + 304*a^15*b^3*c^5*d^5*f^4 + 304*a^3*b^15*c^5*d^5*f^4 - 144*a^16*b
^2*c^4*d^6*f^4 - 144*a^2*b^16*c^6*d^4*f^4 - 108*a^16*b^2*c^2*d^8*f^4 - 108*
a^2*b^16*c^8*d^2*f^4 + 80*a^15*b^3*c^7*d^3*f^4 + 80*a^3*b^15*c^3*d^7*f^4 -
60*a^16*b^2*c^6*d^4*f^4 - 60*a^14*b^4*c^8*d^2*f^4 - 60*a^4*b^14*c^2*d^8*f^4
 - 60*a^2*b^16*c^4*d^6*f^4 - 8*b^18*c^8*d^2*f^4 - 4*b^18*c^6*d^4*f^4 - 8*a^
18*c^2*d^8*f^4 - 4*a^18*c^4*d^6*f^4 - 80*a^12*b^6*d^10*f^4 - 60*a^14*b^4*d^
10*f^4 - 60*a^10*b^8*d^10*f^4 - 24*a^16*b^2*d^10*f^4 - 24*a^8*b^10*d^10*f^4
 - 4*a^6*b^12*d^10*f^4 - 80*a^6*b^12*c^10*f^4 - 60*a^8*b^10*c^10*f^4 - 60*a
^4*b^14*c^10*f^4 - 24*a^10*b^8*c^10*f^4 - 24*a^2*b^16*c^10*f^4 - 4*a^12*b^6
*c^10*f^4 - 4*b^18*c^10*f^4 - 4*a^18*d^10*f^4 - 12*A*C*a^11*b*c*d^7*f^2 - 1
2*A*C*a*b^11*c^7*d*f^2 - 912*B*C*a^5*b^7*c^4*d^4*f^2 - 792*B*C*a^8*b^4*c^3*
d^5*f^2 + 792*B*C*a^4*b^8*c^5*d^3*f^2 + 720*B*C*a^7*b^5*c^4*d^4*f^2 - 480*B
*C*a^5*b^7*c^6*d^2*f^2 - 408*B*C*a^5*b^7*c^2*d^6*f^2 + 384*B*C*a^7*b^5*c^2*
d^6*f^2 - 336*B*C*a^8*b^4*c^5*d^3*f^2 + 324*B*C*a^4*b^8*c^3*d^5*f^2 + 312*B
*C*a^7*b^5*c^6*d^2*f^2 - 248*B*C*a^3*b^9*c^6*d^2*f^2 + 216*B*C*a^9*b^3*c^2*
d^6*f^2 - 196*B*C*a^3*b^9*c^4*d^4*f^2 + 132*B*C*a^9*b^3*c^4*d^4*f^2 + 80*B*
C*a^6*b^6*c^3*d^5*f^2 - 64*B*C*a^6*b^6*c^5*d^3*f^2 - 36*B*C*a^2*b^10*c^3*d^
5*f^2 - 28*B*C*a^3*b^9*c^2*d^6*f^2 + 12*B*C*a^10*b^2*c^5*d^3*f^2 - 12*B*C*a
^10*b^2*c^3*d^5*f^2 - 12*B*C*a^2*b^10*c^5*d^3*f^2 - 4*B*C*a^9*b^3*c^6*d^2*f
^2 - 1468*A*C*a^6*b^6*c^4*d^4*f^2 + 996*A*C*a^7*b^5*c^3*d^5*f^2 + 900*A*C*a
^5*b^7*c^5*d^3*f^2 - 676*A*C*a^6*b^6*c^6*d^2*f^2 - 660*A*C*a^6*b^6*c^2*d^6*
f^2 + 636*A*C*a^5*b^7*c^3*d^5*f^2 + 540*A*C*a^7*b^5*c^5*d^3*f^2 - 236*A*C*a
^3*b^9*c^5*d^3*f^2 - 204*A*C*a^9*b^3*c^3*d^5*f^2 + 156*A*C*a^10*b^2*c^2*d^6
*f^2 + 132*A*C*a^2*b^10*c^6*d^2*f^2 - 72*A*C*a^9*b^3*c^5*d^3*f^2 - 72*A*C*a
^4*b^8*c^6*d^2*f^2 + 66*A*C*a^4*b^8*c^2*d^6*f^2 + 54*A*C*a^10*b^2*c^4*d^4*f
^2 + 54*A*C*a^2*b^10*c^4*d^4*f^2 - 48*A*C*a^8*b^4*c^2*d^6*f^2 - 48*A*C*a^4*
b^8*c^4*d^4*f^2 + 42*A*C*a^8*b^4*c^6*d^2*f^2 - 40*A*C*a^3*b^9*c^3*d^5*f^2 -
 36*A*C*a^8*b^4*c^4*d^4*f^2 + 24*A*C*a^2*b^10*c^2*d^6*f^2 + 960*A*B*a^5*b^7
*c^4*d^4*f^2 - 864*A*B*a^4*b^8*c^5*d^3*f^2 + 756*A*B*a^8*b^4*c^3*d^5*f^2 -
744*A*B*a^7*b^5*c^4*d^4*f^2 - 528*A*B*a^4*b^8*c^3*d^5*f^2 + 504*A*B*a^5*b^7

*c^6*d^2*f^2 - 432*A*B*a^7*b^5*c^2*d^6*f^2 + 432*A*B*a^5*b^7*c^2*d^6*f^2 +
348*A*B*a^8*b^4*c^5*d^3*f^2 - 312*A*B*a^7*b^5*c^6*d^2*f^2 - 284*A*B*a^9*b^3
*c^2*d^6*f^2 + 280*A*B*a^3*b^9*c^6*d^2*f^2 + 264*A*B*a^3*b^9*c^4*d^4*f^2 -
240*A*B*a^6*b^6*c^3*d^5*f^2 - 172*A*B*a^9*b^3*c^4*d^4*f^2 + 68*A*B*a^3*b^9*
c^2*d^6*f^2 - 60*A*B*a^2*b^10*c^3*d^5*f^2 + 24*A*B*a^6*b^6*c^5*d^3*f^2 - 24
*A*B*a^2*b^10*c^5*d^3*f^2 + 12*A*B*a^10*b^2*c^3*d^5*f^2 + 360*B*C*a^4*b^8*c
^7*d*f^2 - 336*B*C*a^8*b^4*c*d^7*f^2 + 168*B*C*a^6*b^6*c*d^7*f^2 - 136*B*C*
a^6*b^6*c^7*d*f^2 - 36*B*C*a^11*b*c^2*d^6*f^2 + 36*B*C*a*b^11*c^6*d^2*f^2 +
24*B*C*a^10*b^2*c*d^7*f^2 - 24*B*C*a^2*b^10*c^7*d*f^2 - 12*B*C*a^11*b*c^4*
d^4*f^2 + 12*B*C*a^4*b^8*c*d^7*f^2 + 12*B*C*a*b^11*c^4*d^4*f^2 + 444*A*C*a^
7*b^5*c*d^7*f^2 + 348*A*C*a^5*b^7*c^7*d*f^2 - 164*A*C*a^3*b^9*c^7*d*f^2 - 1
32*A*C*a^9*b^3*c*d^7*f^2 + 84*A*C*a^5*b^7*c*d^7*f^2 + 32*A*C*a^3*b^9*c*d^7*
f^2 - 12*A*C*a^11*b*c^3*d^5*f^2 - 12*A*C*a^7*b^5*c^7*d*f^2 - 12*A*C*a*b^11*
c^5*d^3*f^2 - 360*A*B*a^4*b^8*c^7*d*f^2 + 288*A*B*a^8*b^4*c*d^7*f^2 - 288*A
*B*a^6*b^6*c*d^7*f^2 - 144*A*B*a^4*b^8*c*d^7*f^2 + 136*A*B*a^6*b^6*c^7*d*f^
2 - 60*A*B*a^2*b^10*c*d^7*f^2 - 36*A*B*a^10*b^2*c*d^7*f^2 + 24*A*B*a^2*b^10
*c^7*d*f^2 - 24*A*B*a*b^11*c^6*d^2*f^2 + 12*A*B*a^11*b*c^2*d^6*f^2 + 12*A*B
*a*b^11*c^4*d^4*f^2 + 12*A*B*a*b^11*c^2*d^6*f^2 - 8*B*C*b^12*c^5*d^3*f^2 -
8*B*C*b^12*c^3*d^5*f^2 + 8*A*C*b^12*c^2*d^6*f^2 - 4*B*C*a^12*c^3*d^5*f^2 +
4*A*C*b^12*c^4*d^4*f^2 - 2*A*C*b^12*c^6*d^2*f^2 + 80*B*C*a^9*b^3*d^8*f^2 -
24*B*C*a^7*b^5*d^8*f^2 + 6*A*C*a^12*c^2*d^6*f^2 + 4*A*B*b^12*c^5*d^3*f^2 -
4*A*B*b^12*c^3*d^5*f^2 - 90*A*C*a^8*b^4*d^8*f^2 - 80*B*C*a^3*b^9*c^8*f^2 +
54*A*C*a^10*b^2*d^8*f^2 - 30*A*C*a^6*b^6*d^8*f^2 + 24*B*C*a^5*b^7*c^8*f^2 -
12*A*C*a^4*b^8*d^8*f^2 - 112*A*B*a^9*b^3*d^8*f^2 - 66*A*C*a^4*b^8*c^8*f^2
+ 54*A*C*a^2*b^10*c^8*f^2 + 4*A*B*a^3*b^9*d^8*f^2 + 2*A*C*a^6*b^6*c^8*f^2 +
80*A*B*a^3*b^9*c^8*f^2 - 24*A*B*a^5*b^7*c^8*f^2 + 726*C^2*a^6*b^6*c^4*d^4*
f^2 - 402*C^2*a^7*b^5*c^3*d^5*f^2 - 402*C^2*a^5*b^7*c^5*d^3*f^2 + 322*C^2*a
^6*b^6*c^6*d^2*f^2 + 322*C^2*a^6*b^6*c^2*d^6*f^2 - 222*C^2*a^7*b^5*c^5*d^3*
f^2 - 222*C^2*a^5*b^7*c^3*d^5*f^2 + 134*C^2*a^9*b^3*c^3*d^5*f^2 + 134*C^2*a
^3*b^9*c^5*d^3*f^2 - 66*C^2*a^10*b^2*c^2*d^6*f^2 - 66*C^2*a^2*b^10*c^6*d^2*
f^2 + 52*C^2*a^9*b^3*c^5*d^3*f^2 + 52*C^2*a^3*b^9*c^3*d^5*f^2 - 27*C^2*a^8*
b^4*c^6*d^2*f^2 - 27*C^2*a^4*b^8*c^2*d^6*f^2 + 24*C^2*a^8*b^4*c^4*d^4*f^2 +
24*C^2*a^8*b^4*c^2*d^6*f^2 + 24*C^2*a^4*b^8*c^6*d^2*f^2 + 24*C^2*a^4*b^8*c
^4*d^4*f^2 - 15*C^2*a^10*b^2*c^4*d^4*f^2 - 15*C^2*a^2*b^10*c^4*d^4*f^2 - 57
0*B^2*a^6*b^6*c^4*d^4*f^2 + 366*B^2*a^7*b^5*c^3*d^5*f^2 + 318*B^2*a^5*b^7*c
^5*d^3*f^2 - 262*B^2*a^6*b^6*c^6*d^2*f^2 - 222*B^2*a^6*b^6*c^2*d^6*f^2 - 21
0*B^2*a^3*b^9*c^5*d^3*f^2 + 186*B^2*a^7*b^5*c^5*d^3*f^2 + 162*B^2*a^5*b^7*c
^3*d^5*f^2 - 142*B^2*a^9*b^3*c^3*d^5*f^2 + 132*B^2*a^4*b^8*c^4*d^4*f^2 + 11
7*B^2*a^4*b^8*c^2*d^6*f^2 + 102*B^2*a^2*b^10*c^6*d^2*f^2 - 96*B^2*a^3*b^9*c
^3*d^5*f^2 + 90*B^2*a^10*b^2*c^2*d^6*f^2 + 81*B^2*a^2*b^10*c^4*d^4*f^2 - 56
*B^2*a^9*b^3*c^5*d^3*f^2 + 48*B^2*a^8*b^4*c^4*d^4*f^2 + 48*B^2*a^4*b^8*c^6*
d^2*f^2 + 45*B^2*a^8*b^4*c^6*d^2*f^2 + 36*B^2*a^8*b^4*c^2*d^6*f^2 + 36*B^2*
a^2*b^10*c^2*d^6*f^2 + 33*B^2*a^10*b^2*c^4*d^4*f^2 + 822*A^2*a^6*b^6*c^4*d^
4*f^2 - 594*A^2*a^7*b^5*c^3*d^5*f^2 + 498*A^2*a^6*b^6*c^2*d^6*f^2 - 498*A^2
*a^5*b^7*c^5*d^3*f^2 - 414*A^2*a^5*b^7*c^3*d^5*f^2 + 354*A^2*a^6*b^6*c^6*d^

2*f^2 - 318*A^2*a^7*b^5*c^5*d^3*f^2 + 144*A^2*a^8*b^4*c^2*d^6*f^2 + 102*A^2
*a^3*b^9*c^5*d^3*f^2 + 84*A^2*a^4*b^8*c^4*d^4*f^2 + 81*A^2*a^4*b^8*c^2*d^6*
f^2 + 72*A^2*a^8*b^4*c^4*d^4*f^2 + 70*A^2*a^9*b^3*c^3*d^5*f^2 - 66*A^2*a^2*
b^10*c^6*d^2*f^2 + 48*A^2*a^4*b^8*c^6*d^2*f^2 - 42*A^2*a^10*b^2*c^2*d^6*f^2
 + 24*A^2*a^2*b^10*c^2*d^6*f^2 + 20*A^2*a^9*b^3*c^5*d^3*f^2 - 15*A^2*a^10*b
^2*c^4*d^4*f^2 - 15*A^2*a^8*b^4*c^6*d^2*f^2 - 15*A^2*a^2*b^10*c^4*d^4*f^2 -
 12*A^2*a^3*b^9*c^3*d^5*f^2 - 8*B*C*b^12*c^7*d*f^2 + 4*B*C*a^12*c*d^7*f^2 -
 24*B*C*a^11*b*d^8*f^2 + 8*A*B*b^12*c^7*d*f^2 - 8*A*B*b^12*c*d^7*f^2 + 24*B
*C*a*b^11*c^8*f^2 - 8*A*B*a^12*c*d^7*f^2 + 12*A*B*a^11*b*d^8*f^2 - 24*A*B*a
*b^11*c^8*f^2 - 174*C^2*a^7*b^5*c*d^7*f^2 - 174*C^2*a^5*b^7*c^7*d*f^2 + 82*
C^2*a^9*b^3*c*d^7*f^2 + 82*C^2*a^3*b^9*c^7*d*f^2 + 6*C^2*a^11*b*c^3*d^5*f^2
 + 6*C^2*a^7*b^5*c^7*d*f^2 + 6*C^2*a^5*b^7*c*d^7*f^2 + 6*C^2*a*b^11*c^5*d^3
*f^2 + 162*B^2*a^7*b^5*c*d^7*f^2 + 138*B^2*a^5*b^7*c^7*d*f^2 - 118*B^2*a^3*
b^9*c^7*d*f^2 - 86*B^2*a^9*b^3*c*d^7*f^2 - 30*B^2*a*b^11*c^5*d^3*f^2 - 18*B
^2*a^7*b^5*c^7*d*f^2 - 18*B^2*a^5*b^7*c*d^7*f^2 - 12*B^2*a*b^11*c^3*d^5*f^2
 - 6*B^2*a^11*b*c^3*d^5*f^2 - 4*B^2*a^3*b^9*c*d^7*f^2 - 270*A^2*a^7*b^5*c*d
^7*f^2 - 174*A^2*a^5*b^7*c^7*d*f^2 - 90*A^2*a^5*b^7*c*d^7*f^2 + 82*A^2*a^3*
b^9*c^7*d*f^2 + 50*A^2*a^9*b^3*c*d^7*f^2 - 32*A^2*a^3*b^9*c*d^7*f^2 + 6*A^2
*a^11*b*c^3*d^5*f^2 + 6*A^2*a^7*b^5*c^7*d*f^2 + 6*A^2*a*b^11*c^5*d^3*f^2 +
6*C^2*a^11*b*c*d^7*f^2 + 6*C^2*a*b^11*c^7*d*f^2 - 18*B^2*a*b^11*c^7*d*f^2 -
 6*B^2*a^11*b*c*d^7*f^2 + 6*A^2*a^11*b*c*d^7*f^2 + 6*A^2*a*b^11*c^7*d*f^2 -
 6*A*C*b^12*c^8*f^2 - 2*A*C*a^12*d^8*f^2 + 4*C^2*b^12*c^4*d^4*f^2 + 3*C^2*b
^12*c^6*d^2*f^2 + 4*C^2*a^12*c^4*d^4*f^2 + 4*B^2*b^12*c^4*d^4*f^2 + 4*B^2*b
^12*c^2*d^6*f^2 + 3*C^2*a^12*c^2*d^6*f^2 + 3*B^2*b^12*c^6*d^2*f^2 + 33*C^2*
a^8*b^4*d^8*f^2 - 27*C^2*a^10*b^2*d^8*f^2 - 4*A^2*b^12*c^4*d^4*f^2 + 3*B^2*
a^12*c^2*d^6*f^2 - C^2*a^6*b^6*d^8*f^2 - A^2*b^12*c^6*d^2*f^2 + 33*C^2*a^4*
b^8*c^8*f^2 + 33*B^2*a^10*b^2*d^8*f^2 - 27*C^2*a^2*b^10*c^8*f^2 - 27*B^2*a^
8*b^4*d^8*f^2 + 3*B^2*a^6*b^6*d^8*f^2 - C^2*a^6*b^6*c^8*f^2 - A^2*a^12*c^2*
d^6*f^2 + 117*A^2*a^8*b^4*d^8*f^2 + 111*A^2*a^6*b^6*d^8*f^2 + 72*A^2*a^4*b^
8*d^8*f^2 + 33*B^2*a^2*b^10*c^8*f^2 - 27*B^2*a^4*b^8*c^8*f^2 + 24*A^2*a^2*b
^10*d^8*f^2 + 3*B^2*a^6*b^6*c^8*f^2 - 3*A^2*a^10*b^2*d^8*f^2 + 33*A^2*a^4*b
^8*c^8*f^2 - 27*A^2*a^2*b^10*c^8*f^2 - A^2*a^6*b^6*c^8*f^2 + 3*C^2*b^12*c^8
*f^2 + 3*C^2*a^12*d^8*f^2 + 4*A^2*b^12*d^8*f^2 - B^2*b^12*c^8*f^2 - B^2*a^1
2*d^8*f^2 + 3*A^2*b^12*c^8*f^2 + 3*A^2*a^12*d^8*f^2 - 24*A*B*C*a*b^8*c*d^6*
f + 342*A*B*C*a^4*b^5*c^2*d^5*f - 186*A*B*C*a^5*b^4*c^3*d^4*f - 66*A*B*C*a^
2*b^7*c^4*d^3*f + 48*A*B*C*a^2*b^7*c^2*d^5*f + 42*A*B*C*a^6*b^3*c^2*d^5*f +
 26*A*B*C*a^3*b^6*c^5*d^2*f + 24*A*B*C*a^6*b^3*c^4*d^3*f - 18*A*B*C*a^7*b^2
*c^3*d^4*f - 18*A*B*C*a^4*b^5*c^4*d^3*f - 8*A*B*C*a^3*b^6*c^3*d^4*f + 6*A*B
*C*a^5*b^4*c^5*d^2*f - 128*A*B*C*a^3*b^6*c*d^6*f + 126*A*B*C*a^7*b^2*c*d^6*
f + 72*A*B*C*a*b^8*c^3*d^4*f - 36*A*B*C*a^8*b*c^2*d^5*f - 36*A*B*C*a*b^8*c^
5*d^2*f + 30*A*B*C*a^2*b^7*c^6*d*f - 12*A*B*C*a^5*b^4*c*d^6*f - 12*A*B*C*a^
4*b^5*c^6*d*f - 21*B^2*C*a^8*b*c*d^6*f - 3*B^2*C*a*b^8*c^6*d*f + 21*A^2*C*a
^8*b*c*d^6*f - 21*A*C^2*a^8*b*c*d^6*f - 9*A^2*C*a*b^8*c^6*d*f + 9*A*C^2*a*b
^8*c^6*d*f + 36*A^2*B*a*b^8*c*d^6*f + 21*A*B^2*a^8*b*c*d^6*f + 3*A*B^2*a*b^
8*c^6*d*f + 16*A*B*C*b^9*c^4*d^3*f - 16*A*B*C*b^9*c^2*d^5*f - 78*A*B*C*a^6*

b^3*d^7*f + 24*A*B*C*a^4*b^5*d^7*f + 2*A*B*C*a^3*b^6*c^7*f - 237*B^2*C*a^4*
b^5*c^3*d^4*f + 165*B*C^2*a^5*b^4*c^3*d^4*f + 92*B^2*C*a^3*b^6*c^2*d^5*f -
81*B^2*C*a^7*b^2*c^2*d^5*f + 77*B^2*C*a^3*b^6*c^4*d^3*f - 75*B*C^2*a^4*b^5*
c^2*d^5*f + 69*B^2*C*a^5*b^4*c^4*d^3*f + 69*B*C^2*a^4*b^5*c^4*d^3*f - 68*B*
C^2*a^3*b^6*c^3*d^4*f - 63*B^2*C*a^4*b^5*c^5*d^2*f - 61*B*C^2*a^6*b^3*c^2*d
^5*f + 57*B*C^2*a^2*b^7*c^4*d^3*f - 53*B*C^2*a^3*b^6*c^5*d^2*f - 44*B*C^2*a
^6*b^3*c^4*d^3*f - 36*B^2*C*a^2*b^7*c^3*d^4*f + 35*B^2*C*a^6*b^3*c^3*d^4*f
- 33*B^2*C*a^5*b^4*c^2*d^5*f + 33*B^2*C*a^2*b^7*c^5*d^2*f + 33*B*C^2*a^7*b^
2*c^3*d^4*f - 12*B^2*C*a^7*b^2*c^4*d^3*f + 9*B*C^2*a^5*b^4*c^5*d^2*f + 4*B^
2*C*a^6*b^3*c^5*d^2*f + 225*A^2*C*a^5*b^4*c^2*d^5*f - 105*A*C^2*a^5*b^4*c^2
*d^5*f - 99*A^2*C*a^4*b^5*c^3*d^4*f - 81*A^2*C*a^4*b^5*c^5*d^2*f + 67*A^2*C
*a^3*b^6*c^4*d^3*f - 59*A*C^2*a^3*b^6*c^4*d^3*f - 57*A*C^2*a^7*b^2*c^2*d^5*
f + 57*A*C^2*a^2*b^7*c^5*d^2*f + 51*A^2*C*a^5*b^4*c^4*d^3*f + 48*A^2*C*a^2*
b^7*c^3*d^4*f + 45*A*C^2*a^4*b^5*c^5*d^2*f - 35*A^2*C*a^6*b^3*c^3*d^4*f + 3
3*A^2*C*a^7*b^2*c^2*d^5*f - 33*A^2*C*a^2*b^7*c^5*d^2*f + 33*A*C^2*a^5*b^4*c
^4*d^3*f + 27*A*C^2*a^6*b^3*c^3*d^4*f + 24*A*C^2*a^3*b^6*c^2*d^5*f - 24*A*C
^2*a^2*b^7*c^3*d^4*f - 21*A*C^2*a^4*b^5*c^3*d^4*f - 16*A^2*C*a^3*b^6*c^2*d^
5*f - 243*A^2*B*a^4*b^5*c^2*d^5*f - 156*A*B^2*a^3*b^6*c^2*d^5*f + 141*A*B^2
*a^4*b^5*c^3*d^4*f + 108*A^2*B*a^3*b^6*c^3*d^4*f - 105*A*B^2*a^3*b^6*c^4*d^
3*f + 84*A*B^2*a^2*b^7*c^3*d^4*f + 81*A*B^2*a^5*b^4*c^2*d^5*f + 51*A^2*B*a^
6*b^3*c^2*d^5*f - 51*A^2*B*a^4*b^5*c^4*d^3*f - 48*A^2*B*a^2*b^7*c^2*d^5*f +
45*A^2*B*a^5*b^4*c^3*d^4*f + 39*A*B^2*a^4*b^5*c^5*d^2*f - 35*A*B^2*a^6*b^3
*c^3*d^4*f + 33*A*B^2*a^7*b^2*c^2*d^5*f + 27*A^2*B*a^3*b^6*c^5*d^2*f - 21*A
*B^2*a^5*b^4*c^4*d^3*f + 20*A^2*B*a^6*b^3*c^4*d^3*f - 15*A^2*B*a^7*b^2*c^3*
d^4*f - 15*A^2*B*a^5*b^4*c^5*d^2*f + 9*A^2*B*a^2*b^7*c^4*d^3*f + 3*A*B^2*a^
2*b^7*c^5*d^2*f + 2*A*B*C*b^9*c^6*d*f - 6*A*B*C*a^9*c*d^6*f + 18*A*B*C*a^8*
b*d^7*f - 6*A*B*C*a*b^8*c^7*f + 63*B^2*C*a^6*b^3*c*d^6*f - 48*B^2*C*a*b^8*c
^4*d^3*f + 42*B*C^2*a^8*b*c^2*d^5*f + 42*B*C^2*a^5*b^4*c*d^6*f - 39*B*C^2*a
^7*b^2*c*d^6*f + 30*B*C^2*a*b^8*c^5*d^2*f - 24*B^2*C*a^4*b^5*c*d^6*f - 24*B
*C^2*a*b^8*c^3*d^4*f + 17*B^2*C*a^3*b^6*c^6*d*f - 15*B*C^2*a^2*b^7*c^6*d*f
+ 12*B^2*C*a^8*b*c^3*d^4*f + 12*B^2*C*a*b^8*c^2*d^5*f + 6*B*C^2*a^4*b^5*c^6
*d*f - 192*A^2*C*a^4*b^5*c*d^6*f - 99*A^2*C*a^6*b^3*c*d^6*f + 84*A*C^2*a^4*
b^5*c*d^6*f + 59*A*C^2*a^6*b^3*c*d^6*f + 51*A^2*C*a^3*b^6*c^6*d*f - 51*A*C^
2*a^3*b^6*c^6*d*f - 36*A^2*C*a*b^8*c^2*d^5*f - 24*A*C^2*a*b^8*c^4*d^3*f + 2
4*A*C^2*a*b^8*c^2*d^5*f + 12*A^2*C*a*b^8*c^4*d^3*f + 12*A*C^2*a^8*b*c^3*d^4
*f + 160*A^2*B*a^3*b^6*c*d^6*f - 99*A*B^2*a^6*b^3*c*d^6*f - 87*A^2*B*a^7*b^
2*c*d^6*f - 72*A*B^2*a^4*b^5*c*d^6*f - 48*A*B^2*a*b^8*c^2*d^5*f - 36*A^2*B*
a*b^8*c^3*d^4*f + 24*A*B^2*a*b^8*c^4*d^3*f - 17*A*B^2*a^3*b^6*c^6*d*f - 15*
A^2*B*a^2*b^7*c^6*d*f + 12*A*B^2*a^2*b^7*c*d^6*f + 6*A^2*B*a^8*b*c^2*d^5*f
- 6*A^2*B*a^5*b^4*c*d^6*f + 6*A^2*B*a^4*b^5*c^6*d*f + 6*A^2*B*a*b^8*c^5*d^2
*f + 12*B^2*C*b^9*c^3*d^4*f - 12*B*C^2*b^9*c^4*d^3*f - 12*A^2*C*b^9*c^3*d^4
*f - 8*A*C^2*b^9*c^5*d^2*f + 8*A*C^2*b^9*c^3*d^4*f + 4*B^2*C*a^9*c^2*d^5*f
+ 4*A^2*C*b^9*c^5*d^2*f - 4*B*C^2*a^9*c^3*d^4*f + 12*A^2*B*b^9*c^2*d^5*f -
8*A*B^2*b^9*c^3*d^4*f - 4*A^2*B*b^9*c^4*d^3*f + 4*A*C^2*a^9*c^2*d^5*f + 3*B
^2*C*a^7*b^2*d^7*f - B*C^2*a^6*b^3*d^7*f + 96*A^2*C*a^5*b^4*d^7*f - 39*A^2*

C*a^7*b^2*d^7*f - 36*A*C^2*a^5*b^4*d^7*f + 32*A^2*C*a^3*b^6*d^7*f + 15*A*C^
2*a^7*b^2*d^7*f - 3*B^2*C*a^2*b^7*c^7*f - B*C^2*a^3*b^6*c^7*f + 111*A^2*B*a
^6*b^3*d^7*f - 39*A*B^2*a^7*b^2*d^7*f + 24*A*B^2*a^5*b^4*d^7*f - 9*A^2*C*a^
2*b^7*c^7*f + 9*A*C^2*a^2*b^7*c^7*f - 4*A*B^2*a^3*b^6*d^7*f + 3*A*B^2*a^2*b
^7*c^7*f - A^2*B*a^3*b^6*c^7*f + 3*C^3*a^8*b*c*d^6*f - 3*C^3*a*b^8*c^6*d*f
- 3*A^3*a^8*b*c*d^6*f + 3*A^3*a*b^8*c^6*d*f - B*C^2*b^9*c^6*d*f + 4*A^2*C*b
^9*c*d^6*f + 3*B*C^2*a^9*c*d^6*f + 8*A*B^2*b^9*c*d^6*f + 3*B*C^2*a^8*b*d^7*
f - A^2*B*b^9*c^6*d*f + 12*A^2*C*a*b^8*d^7*f + 3*B*C^2*a*b^8*c^7*f - A^2*B*
a^9*c*d^6*f - 9*A^2*B*a^8*b*d^7*f + 3*A^2*B*a*b^8*c^7*f - 39*C^3*a^5*b^4*c^
4*d^3*f + 39*C^3*a^4*b^5*c^3*d^4*f + 27*C^3*a^7*b^2*c^2*d^5*f - 27*C^3*a^2*
b^7*c^5*d^2*f - 17*C^3*a^6*b^3*c^3*d^4*f + 17*C^3*a^3*b^6*c^4*d^3*f + 3*C^3
*a^5*b^4*c^2*d^5*f - 3*C^3*a^4*b^5*c^5*d^2*f - 63*B^3*a^5*b^4*c^3*d^4*f + 5
7*B^3*a^4*b^5*c^2*d^5*f - 51*B^3*a^2*b^7*c^4*d^3*f + 48*B^3*a^3*b^6*c^3*d^4
*f + 31*B^3*a^6*b^3*c^2*d^5*f + 27*B^3*a^3*b^6*c^5*d^2*f + 16*B^3*a^6*b^3*c
^4*d^3*f - 15*B^3*a^5*b^4*c^5*d^2*f - 12*B^3*a^2*b^7*c^2*d^5*f + 9*B^3*a^4*
b^5*c^4*d^3*f - 3*B^3*a^7*b^2*c^3*d^4*f - 123*A^3*a^5*b^4*c^2*d^5*f + 81*A^
3*a^4*b^5*c^3*d^4*f - 45*A^3*a^5*b^4*c^4*d^3*f + 39*A^3*a^4*b^5*c^5*d^2*f +
25*A^3*a^6*b^3*c^3*d^4*f - 25*A^3*a^3*b^6*c^4*d^3*f - 24*A^3*a^2*b^7*c^3*d
^4*f - 8*A^3*a^3*b^6*c^2*d^5*f - 3*A^3*a^7*b^2*c^2*d^5*f + 3*A^3*a^2*b^7*c^
5*d^2*f - 17*C^3*a^6*b^3*c*d^6*f + 17*C^3*a^3*b^6*c^6*d*f - 12*C^3*a^8*b*c^
3*d^4*f + 12*C^3*a*b^8*c^4*d^3*f + 24*B^3*a*b^8*c^3*d^4*f + 21*B^3*a^7*b^2*
c*d^6*f - 18*B^3*a^5*b^4*c*d^6*f - 15*B^3*a^2*b^7*c^6*d*f - 6*B^3*a^8*b*c^2
*d^5*f + 6*B^3*a^4*b^5*c^6*d*f + 6*B^3*a*b^8*c^5*d^2*f + 4*B^3*a^3*b^6*c*d^
6*f + 108*A^3*a^4*b^5*c*d^6*f + 57*A^3*a^6*b^3*c*d^6*f - 17*A^3*a^3*b^6*c^6
*d*f + 12*A^3*a*b^8*c^2*d^5*f + 4*C^3*b^9*c^5*d^2*f - 4*C^3*a^9*c^2*d^5*f -
4*B^3*b^9*c^2*d^5*f + 4*A^3*b^9*c^3*d^4*f + 3*C^3*a^7*b^2*d^7*f - 3*C^3*a^
2*b^7*c^7*f - B^3*a^6*b^3*d^7*f - 60*A^3*a^5*b^4*d^7*f - 32*A^3*a^3*b^6*d^7
*f + 21*A^3*a^7*b^2*d^7*f - B^3*a^3*b^6*c^7*f + 3*A^3*a^2*b^7*c^7*f - B^3*b
^9*c^6*d*f - 4*A^3*b^9*c*d^6*f - B^3*a^9*c*d^6*f + 3*B^3*a^8*b*d^7*f - 12*A
^3*a*b^8*d^7*f + 3*B^3*a*b^8*c^7*f - B^2*C*a^9*d^7*f - 4*A^2*B*b^9*d^7*f +
3*A^2*C*b^9*c^7*f - 3*A*C^2*b^9*c^7*f - A*C^2*a^9*d^7*f - A*B^2*b^9*c^7*f -
C^3*a^9*d^7*f - A^3*b^9*c^7*f + B^2*C*b^9*c^7*f + A^2*C*a^9*d^7*f + A*B^2*
a^9*d^7*f + C^3*b^9*c^7*f + A^3*a^9*d^7*f - 6*A*B^2*C*a^5*b*c*d^5 - 21*A^2*
B*C*a^3*b^3*c^2*d^4 + 21*A*B*C^2*a^3*b^3*c^2*d^4 + 12*A*B^2*C*a^4*b^2*c^2*d
^4 - 12*A*B^2*C*a^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^
4*b^2*c^3*d^3 + 3*A^2*B*C*a^4*b^2*c^3*d^3 + 3*A^2*B*C*a^2*b^4*c^3*d^3 + 3*A
*B^2*C*a^2*b^4*c^4*d^2 + 3*A*B*C^2*a^2*b^4*c^3*d^3 + 2*A*B*C^2*a^3*b^3*c^4*
d^2 - A^2*B*C*a^3*b^3*c^4*d^2 + 18*A^2*B*C*a^2*b^4*c*d^5 + 10*A*B^2*C*a^3*b
^3*c*d^5 + 9*A^2*B*C*a^4*b^2*c*d^5 - 9*A*B*C^2*a^4*b^2*c*d^5 - 9*A*B*C^2*a^
2*b^4*c*d^5 - 6*A^2*B*C*a*b^5*c^2*d^4 + 6*A*B^2*C*a*b^5*c^3*d^3 + 6*A*B*C^2
*a^5*b*c^2*d^4 - 6*A*B*C^2*a*b^5*c^4*d^2 - 3*A^2*B*C*a^5*b*c^2*d^4 + 3*A^2*
B*C*a*b^5*c^4*d^2 + 3*A*B*C^2*a*b^5*c^2*d^4 - 3*B^3*C*a^5*b*c^2*d^4 + 3*B^3
*C*a^4*b^2*c*d^5 + 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a^5*b*c*d^5 - 3*B*C^3*
a^5*b*c^2*d^4 + 3*B*C^3*a^4*b^2*c*d^5 + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a^
3*b^3*c*d^5 + 8*A*C^3*a^3*b^3*c*d^5 - 9*A^3*B*a^2*b^4*c*d^5 - 9*A*B^3*a^2*b

^4*c*d^5 - 3*A^3*B*a^4*b^2*c*d^5 + 3*A^3*B*a*b^5*c^2*d^4 + 3*A^2*B^2*a^5*b*
c*d^5 - 3*A*B^3*a^4*b^2*c*d^5 + 3*A*B^3*a*b^5*c^2*d^4 + 5*A*B*C^2*b^6*c^3*d
^3 - 4*A^2*B*C*b^6*c^3*d^3 - A*B^2*C*b^6*c^4*d^2 - 3*A*B^2*C*a^4*b^2*d^6 -
2*A^2*B*C*a^3*b^3*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^4*b^2*c^2*d
^4 + 6*B^2*C^2*a^2*b^4*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 24*A^2*C^2*a^3
*b^3*c^3*d^3 - 15*A^2*C^2*a^4*b^2*c^2*d^4 - 9*A^2*C^2*a^2*b^4*c^4*d^2 + 3*A
^2*C^2*a^2*b^4*c^2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^4*b^2*c^2*
d^4 + 4*A^2*B*C*b^6*c*d^5 - 2*A*B*C^2*b^6*c*d^5 + 2*A*B*C^2*a^6*c*d^5 - A^2
*B*C*a^6*c*d^5 + 6*A^2*B*C*a^5*b*d^6 - 3*A*B*C^2*a^5*b*d^6 - 7*B^3*C*a^3*b^
3*c^2*d^4 - 7*B*C^3*a^3*b^3*c^2*d^4 + 3*B^3*C*a^4*b^2*c^3*d^3 - 3*B^3*C*a^2
*b^4*c^3*d^3 - 3*B^2*C^2*a*b^5*c^3*d^3 + 3*B*C^3*a^4*b^2*c^3*d^3 - 3*B*C^3*
a^2*b^4*c^3*d^3 - B^3*C*a^3*b^3*c^4*d^2 - B^2*C^2*a^3*b^3*c*d^5 - B*C^3*a^3
*b^3*c^4*d^2 - 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C
^3*a^4*b^2*c^2*d^4 + 9*A*C^3*a^2*b^4*c^4*d^2 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*
A^3*C*a^4*b^2*c^2*d^4 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^2*b^4*c^4*d^2 -
9*A^2*B^2*a^3*b^3*c*d^5 + 7*A^3*B*a^3*b^3*c^2*d^4 + 7*A*B^3*a^3*b^3*c^2*d^
4 - 3*A^3*B*a^2*b^4*c^3*d^3 - 3*A^2*B^2*a*b^5*c^3*d^3 - 3*A*B^3*a^2*b^4*c^3
*d^3 - 5*A^2*C^2*b^6*c^2*d^4 + 3*A^2*C^2*b^6*c^4*d^2 + 12*A^2*C^2*a^4*b^2*d
^6 + 3*A^2*C^2*a^2*b^4*d^6 + 6*A^2*B^2*a^4*b^2*d^6 + 3*A^2*B^2*a^2*b^4*d^6
+ A*B*C^2*a^3*b^3*d^6 - 3*B^4*a*b^5*c^3*d^3 - B^4*a^3*b^3*c*d^5 + A^2*B^2*a
^3*b^3*c^3*d^3 - 8*A^4*a^3*b^3*c*d^5 - 2*B^3*C*b^6*c^3*d^3 - 2*B*C^3*b^6*c^
3*d^3 + 4*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 + 2*A*C^3*b^6*c^2*d^4 - A
^3*C*b^6*c^4*d^2 - 2*A*C^3*a^6*c^2*d^4 - 15*A^3*C*a^4*b^2*d^6 - 6*A^3*C*a^2
*b^4*d^6 - 3*A*C^3*a^4*b^2*d^6 + 3*B^4*a^5*b*c*d^5 - B^3*C*a^6*c*d^5 - B*C^
3*a^6*c*d^5 - 2*A^3*B*b^6*c*d^5 - 2*A*B^3*b^6*c*d^5 - 3*A^3*B*a^5*b*d^6 - 3
*A*B^3*a^5*b*d^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^
2*b^4*c^4*d^2 + 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^4*b^2*c^2*d^4 + 3*A^4*a^2*b
^4*c^2*d^4 + B^2*C^2*b^6*c^4*d^2 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^6*c^2*d^
4 + A^2*C^2*a^6*c^2*d^4 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c
^3*d^3 + A^3*B*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 - A^4*b^6*c^2*d^4 + 6*A^4*a^
4*b^2*d^6 + 3*A^4*a^2*b^4*d^6 - 2*A^2*C^2*a^6*d^6 + A*B^2*C*a^6*d^6 + B^4*a
^3*b^3*c^3*d^3 + A^3*C*a^6*d^6 + A*C^3*a^6*d^6 + C^4*b^6*c^4*d^2 + C^4*a^6*
c^2*d^4 + B^4*b^6*c^2*d^4 + A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6
, f, k)*((B*b^14*c^7*d - B*a^13*b*d^8 - 4*A*a^2*b^12*d^8 - 16*A*a^4*b^10*d^
8 - 35*A*a^6*b^8*d^8 - 33*A*a^8*b^6*d^8 - 5*A*a^10*b^4*d^8 + 5*A*a^12*b^2*d
^8 - 4*B*a^5*b^9*d^8 + 3*B*a^7*b^7*d^8 + 17*B*a^9*b^5*d^8 + 9*B*a^11*b^3*d^
8 - 4*A*b^14*c^2*d^6 + 4*A*b^14*c^4*d^4 - 3*A*b^14*c^6*d^2 + 11*C*a^6*b^8*d
^8 + 17*C*a^8*b^6*d^8 + C*a^10*b^4*d^8 - 5*C*a^12*b^2*d^8 + 4*B*b^14*c^3*d^
5 - 4*B*b^14*c^5*d^3 - 4*C*b^14*c^4*d^4 + 3*C*b^14*c^6*d^2 - 6*A*a*b^13*c^5
*d^3 + 40*A*a^3*b^11*c*d^7 + 3*A*a^3*b^11*c^7*d + 122*A*a^5*b^9*c*d^7 + 3*A
*a^5*b^9*c^7*d + 175*A*a^7*b^7*c*d^7 + A*a^7*b^7*c^7*d + 105*A*a^9*b^5*c*d^
7 + 21*A*a^11*b^3*c*d^7 - 8*B*a*b^13*c^2*d^6 - 4*B*a*b^13*c^4*d^4 + 5*B*a*b
^13*c^6*d^2 + 4*B*a^2*b^12*c*d^7 + 3*B*a^2*b^12*c^7*d + 32*B*a^4*b^10*c*d^7
+ 3*B*a^4*b^10*c^7*d + 31*B*a^6*b^8*c*d^7 + B*a^6*b^8*c^7*d - 27*B*a^8*b^6
*c*d^7 - 39*B*a^10*b^4*c*d^7 - 9*B*a^12*b^2*c*d^7 + 8*C*a*b^13*c^3*d^5 + 10

*C*a*b^13*c^5*d^3 - 3*C*a^3*b^11*c^7*d - 38*C*a^5*b^9*c*d^7 - 3*C*a^5*b^9*c
^7*d - 79*C*a^7*b^7*c*d^7 - C*a^7*b^7*c^7*d - 41*C*a^9*b^5*c*d^7 + 3*C*a^11
*b^3*c*d^7 - 28*A*a^2*b^12*c^2*d^6 + 43*A*a^2*b^12*c^4*d^4 + A*a^2*b^12*c^6
*d^2 - 4*A*a^3*b^11*c^3*d^5 - 35*A*a^3*b^11*c^5*d^3 - 117*A*a^4*b^10*c^2*d^
6 + 69*A*a^4*b^10*c^4*d^4 + 5*A*a^4*b^10*c^6*d^2 + 67*A*a^5*b^9*c^3*d^5 - 3
7*A*a^5*b^9*c^5*d^3 - 245*A*a^6*b^8*c^2*d^6 + 5*A*a^6*b^8*c^4*d^4 - 5*A*a^6
*b^8*c^6*d^2 + 161*A*a^7*b^7*c^3*d^5 + 7*A*a^7*b^7*c^5*d^3 - 237*A*a^8*b^6*
c^2*d^6 - 45*A*a^8*b^6*c^4*d^4 - 6*A*a^8*b^6*c^6*d^2 + 105*A*a^9*b^5*c^3*d^
5 + 15*A*a^9*b^5*c^5*d^3 - 91*A*a^10*b^4*c^2*d^6 - 20*A*a^10*b^4*c^4*d^4 +
15*A*a^11*b^3*c^3*d^5 - 6*A*a^12*b^2*c^2*d^6 + 44*B*a^2*b^12*c^3*d^5 - 11*B
*a^2*b^12*c^5*d^3 - 64*B*a^3*b^11*c^2*d^6 - 71*B*a^3*b^11*c^4*d^4 - B*a^3*b
^11*c^6*d^2 + 187*B*a^4*b^10*c^3*d^5 + 23*B*a^4*b^10*c^5*d^3 - 145*B*a^5*b^
9*c^2*d^6 - 173*B*a^5*b^9*c^4*d^4 - 17*B*a^5*b^9*c^6*d^2 + 273*B*a^6*b^8*c^
3*d^5 + 63*B*a^6*b^8*c^5*d^3 - 115*B*a^7*b^7*c^2*d^6 - 149*B*a^7*b^7*c^4*d^
4 - 11*B*a^7*b^7*c^6*d^2 + 141*B*a^8*b^6*c^3*d^5 + 33*B*a^8*b^6*c^5*d^3 - 1
1*B*a^9*b^5*c^2*d^6 - 43*B*a^9*b^5*c^4*d^4 + 15*B*a^10*b^4*c^3*d^5 + 15*B*a
^11*b^3*c^2*d^6 - 4*C*a^2*b^12*c^2*d^6 - 47*C*a^2*b^12*c^4*d^4 - C*a^2*b^12
*c^6*d^2 + 36*C*a^3*b^11*c^3*d^5 + 51*C*a^3*b^11*c^5*d^3 + 25*C*a^4*b^10*c^
2*d^6 - 85*C*a^4*b^10*c^4*d^4 - 5*C*a^4*b^10*c^6*d^2 - 19*C*a^5*b^9*c^3*d^5
 + 61*C*a^5*b^9*c^5*d^3 + 117*C*a^6*b^8*c^2*d^6 - 29*C*a^6*b^8*c^4*d^4 + 5*
C*a^6*b^8*c^6*d^2 - 129*C*a^7*b^7*c^3*d^5 + 9*C*a^7*b^7*c^5*d^3 + 145*C*a^8
*b^6*c^2*d^6 + 29*C*a^8*b^6*c^4*d^4 + 6*C*a^8*b^6*c^6*d^2 - 97*C*a^9*b^5*c^
3*d^5 - 11*C*a^9*b^5*c^5*d^3 + 59*C*a^10*b^4*c^2*d^6 + 16*C*a^10*b^4*c^4*d^
4 - 15*C*a^11*b^3*c^3*d^5 + 2*C*a^12*b^2*c^2*d^6 + 8*A*a*b^13*c*d^7 + A*a*b
^13*c^7*d + A*a^13*b*c*d^7 - C*a*b^13*c^7*d + 3*C*a^13*b*c*d^7)/(a^12*d^4 +
 b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4 +
a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9*c*
d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^5*c
*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^10*c
^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d^2 + 6*a
^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3) + root(480*a^11*b^7*c*d^
9*f^4 + 480*a^7*b^11*c^9*d*f^4 + 360*a^13*b^5*c*d^9*f^4 + 360*a^9*b^9*c^9*d
*f^4 + 360*a^9*b^9*c*d^9*f^4 + 360*a^5*b^13*c^9*d*f^4 + 144*a^15*b^3*c*d^9*
f^4 + 144*a^11*b^7*c^9*d*f^4 + 144*a^7*b^11*c*d^9*f^4 + 144*a^3*b^15*c^9*d*
f^4 + 48*a^17*b*c^3*d^7*f^4 + 48*a*b^17*c^7*d^3*f^4 + 24*a^17*b*c^5*d^5*f^4
 + 24*a^13*b^5*c^9*d*f^4 + 24*a^5*b^13*c*d^9*f^4 + 24*a*b^17*c^5*d^5*f^4 +
24*a^17*b*c*d^9*f^4 + 24*a*b^17*c^9*d*f^4 + 3920*a^9*b^9*c^5*d^5*f^4 - 3360
*a^10*b^8*c^4*d^6*f^4 - 3360*a^8*b^10*c^6*d^4*f^4 + 3024*a^11*b^7*c^5*d^5*f
^4 - 3024*a^10*b^8*c^6*d^4*f^4 - 3024*a^8*b^10*c^4*d^6*f^4 + 3024*a^7*b^11*
c^5*d^5*f^4 + 2320*a^9*b^9*c^7*d^3*f^4 + 2320*a^9*b^9*c^3*d^7*f^4 - 2240*a^
12*b^6*c^4*d^6*f^4 - 2240*a^6*b^12*c^6*d^4*f^4 + 2160*a^11*b^7*c^3*d^7*f^4
+ 2160*a^7*b^11*c^7*d^3*f^4 - 1624*a^12*b^6*c^6*d^4*f^4 - 1624*a^6*b^12*c^4
*d^6*f^4 + 1488*a^11*b^7*c^7*d^3*f^4 + 1488*a^7*b^11*c^3*d^7*f^4 + 1344*a^1
3*b^5*c^5*d^5*f^4 + 1344*a^5*b^13*c^5*d^5*f^4 - 1320*a^10*b^8*c^2*d^8*f^4 -
 1320*a^8*b^10*c^8*d^2*f^4 + 1200*a^13*b^5*c^3*d^7*f^4 + 1200*a^5*b^13*c^7*

d^3*f^4 - 1060*a^12*b^6*c^2*d^8*f^4 - 1060*a^6*b^12*c^8*d^2*f^4 - 948*a^10*
b^8*c^8*d^2*f^4 - 948*a^8*b^10*c^2*d^8*f^4 - 840*a^14*b^4*c^4*d^6*f^4 - 840
*a^4*b^14*c^6*d^4*f^4 + 528*a^13*b^5*c^7*d^3*f^4 + 528*a^5*b^13*c^3*d^7*f^4
 - 480*a^14*b^4*c^6*d^4*f^4 - 480*a^14*b^4*c^2*d^8*f^4 - 480*a^4*b^14*c^8*d
^2*f^4 - 480*a^4*b^14*c^4*d^6*f^4 + 368*a^15*b^3*c^3*d^7*f^4 - 368*a^12*b^6
*c^8*d^2*f^4 - 368*a^6*b^12*c^2*d^8*f^4 + 368*a^3*b^15*c^7*d^3*f^4 + 304*a^
15*b^3*c^5*d^5*f^4 + 304*a^3*b^15*c^5*d^5*f^4 - 144*a^16*b^2*c^4*d^6*f^4 -
144*a^2*b^16*c^6*d^4*f^4 - 108*a^16*b^2*c^2*d^8*f^4 - 108*a^2*b^16*c^8*d^2*
f^4 + 80*a^15*b^3*c^7*d^3*f^4 + 80*a^3*b^15*c^3*d^7*f^4 - 60*a^16*b^2*c^6*d
^4*f^4 - 60*a^14*b^4*c^8*d^2*f^4 - 60*a^4*b^14*c^2*d^8*f^4 - 60*a^2*b^16*c^
4*d^6*f^4 - 8*b^18*c^8*d^2*f^4 - 4*b^18*c^6*d^4*f^4 - 8*a^18*c^2*d^8*f^4 -
4*a^18*c^4*d^6*f^4 - 80*a^12*b^6*d^10*f^4 - 60*a^14*b^4*d^10*f^4 - 60*a^10*
b^8*d^10*f^4 - 24*a^16*b^2*d^10*f^4 - 24*a^8*b^10*d^10*f^4 - 4*a^6*b^12*d^1
0*f^4 - 80*a^6*b^12*c^10*f^4 - 60*a^8*b^10*c^10*f^4 - 60*a^4*b^14*c^10*f^4
 - 24*a^10*b^8*c^10*f^4 - 24*a^2*b^16*c^10*f^4 - 4*a^12*b^6*c^10*f^4 - 4*b^1
8*c^10*f^4 - 4*a^18*d^10*f^4 - 12*A*C*a^11*b*c*d^7*f^2 - 12*A*C*a*b^11*c^7*
d*f^2 - 912*B*C*a^5*b^7*c^4*d^4*f^2 - 792*B*C*a^8*b^4*c^3*d^5*f^2 + 792*B*C
*a^4*b^8*c^5*d^3*f^2 + 720*B*C*a^7*b^5*c^4*d^4*f^2 - 480*B*C*a^5*b^7*c^6*d^
2*f^2 - 408*B*C*a^5*b^7*c^2*d^6*f^2 + 384*B*C*a^7*b^5*c^2*d^6*f^2 - 336*B*C
*a^8*b^4*c^5*d^3*f^2 + 324*B*C*a^4*b^8*c^3*d^5*f^2 + 312*B*C*a^7*b^5*c^6*d^
2*f^2 - 248*B*C*a^3*b^9*c^6*d^2*f^2 + 216*B*C*a^9*b^3*c^2*d^6*f^2 - 196*B*C
*a^3*b^9*c^4*d^4*f^2 + 132*B*C*a^9*b^3*c^4*d^4*f^2 + 80*B*C*a^6*b^6*c^3*d^5
*f^2 - 64*B*C*a^6*b^6*c^5*d^3*f^2 - 36*B*C*a^2*b^10*c^3*d^5*f^2 - 28*B*C*a^
3*b^9*c^2*d^6*f^2 + 12*B*C*a^10*b^2*c^5*d^3*f^2 - 12*B*C*a^10*b^2*c^3*d^5*f
^2 - 12*B*C*a^2*b^10*c^5*d^3*f^2 - 4*B*C*a^9*b^3*c^6*d^2*f^2 - 1468*A*C*a^6
*b^6*c^4*d^4*f^2 + 996*A*C*a^7*b^5*c^3*d^5*f^2 + 900*A*C*a^5*b^7*c^5*d^3*f^
2 - 676*A*C*a^6*b^6*c^6*d^2*f^2 - 660*A*C*a^6*b^6*c^2*d^6*f^2 + 636*A*C*a^5
*b^7*c^3*d^5*f^2 + 540*A*C*a^7*b^5*c^5*d^3*f^2 - 236*A*C*a^3*b^9*c^5*d^3*f^
2 - 204*A*C*a^9*b^3*c^3*d^5*f^2 + 156*A*C*a^10*b^2*c^2*d^6*f^2 + 132*A*C*a^
2*b^10*c^6*d^2*f^2 - 72*A*C*a^9*b^3*c^5*d^3*f^2 - 72*A*C*a^4*b^8*c^6*d^2*f^
2 + 66*A*C*a^4*b^8*c^2*d^6*f^2 + 54*A*C*a^10*b^2*c^4*d^4*f^2 + 54*A*C*a^2*b
^10*c^4*d^4*f^2 - 48*A*C*a^8*b^4*c^2*d^6*f^2 - 48*A*C*a^4*b^8*c^4*d^4*f^2 +
 42*A*C*a^8*b^4*c^6*d^2*f^2 - 40*A*C*a^3*b^9*c^3*d^5*f^2 - 36*A*C*a^8*b^4*c
^4*d^4*f^2 + 24*A*C*a^2*b^10*c^2*d^6*f^2 + 960*A*B*a^5*b^7*c^4*d^4*f^2 - 86
4*A*B*a^4*b^8*c^5*d^3*f^2 + 756*A*B*a^8*b^4*c^3*d^5*f^2 - 744*A*B*a^7*b^5*c
^4*d^4*f^2 - 528*A*B*a^4*b^8*c^3*d^5*f^2 + 504*A*B*a^5*b^7*c^6*d^2*f^2 - 43
2*A*B*a^7*b^5*c^2*d^6*f^2 + 432*A*B*a^5*b^7*c^2*d^6*f^2 + 348*A*B*a^8*b^4*c
^5*d^3*f^2 - 312*A*B*a^7*b^5*c^6*d^2*f^2 - 284*A*B*a^9*b^3*c^2*d^6*f^2 + 28
0*A*B*a^3*b^9*c^6*d^2*f^2 + 264*A*B*a^3*b^9*c^4*d^4*f^2 - 240*A*B*a^6*b^6*c
^3*d^5*f^2 - 172*A*B*a^9*b^3*c^4*d^4*f^2 + 68*A*B*a^3*b^9*c^2*d^6*f^2 - 60*
A*B*a^2*b^10*c^3*d^5*f^2 + 24*A*B*a^6*b^6*c^5*d^3*f^2 - 24*A*B*a^2*b^10*c^5
*d^3*f^2 + 12*A*B*a^10*b^2*c^3*d^5*f^2 + 360*B*C*a^4*b^8*c^7*d*f^2 - 336*B*
C*a^8*b^4*c*d^7*f^2 + 168*B*C*a^6*b^6*c*d^7*f^2 - 136*B*C*a^6*b^6*c^7*d*f^2
 - 36*B*C*a^11*b*c^2*d^6*f^2 + 36*B*C*a*b^11*c^6*d^2*f^2 + 24*B*C*a^10*b^2*
c*d^7*f^2 - 24*B*C*a^2*b^10*c^7*d*f^2 - 12*B*C*a^11*b*c^4*d^4*f^2 + 12*B*C*

a^4*b^8*c*d^7*f^2 + 12*B*C*a*b^11*c^4*d^4*f^2 + 444*A*C*a^7*b^5*c*d^7*f^2 +
348*A*C*a^5*b^7*c^7*d*f^2 - 164*A*C*a^3*b^9*c^7*d*f^2 - 132*A*C*a^9*b^3*c*
d^7*f^2 + 84*A*C*a^5*b^7*c*d^7*f^2 + 32*A*C*a^3*b^9*c*d^7*f^2 - 12*A*C*a^11
*b*c^3*d^5*f^2 - 12*A*C*a^7*b^5*c^7*d*f^2 - 12*A*C*a*b^11*c^5*d^3*f^2 - 360
*A*B*a^4*b^8*c^7*d*f^2 + 288*A*B*a^8*b^4*c*d^7*f^2 - 288*A*B*a^6*b^6*c*d^7*
f^2 - 144*A*B*a^4*b^8*c*d^7*f^2 + 136*A*B*a^6*b^6*c^7*d*f^2 - 60*A*B*a^2*b^
10*c*d^7*f^2 - 36*A*B*a^10*b^2*c*d^7*f^2 + 24*A*B*a^2*b^10*c^7*d*f^2 - 24*A
*B*a*b^11*c^6*d^2*f^2 + 12*A*B*a^11*b*c^2*d^6*f^2 + 12*A*B*a*b^11*c^4*d^4*f
^2 + 12*A*B*a*b^11*c^2*d^6*f^2 - 8*B*C*b^12*c^5*d^3*f^2 - 8*B*C*b^12*c^3*d^
5*f^2 + 8*A*C*b^12*c^2*d^6*f^2 - 4*B*C*a^12*c^3*d^5*f^2 + 4*A*C*b^12*c^4*d^
4*f^2 - 2*A*C*b^12*c^6*d^2*f^2 + 80*B*C*a^9*b^3*d^8*f^2 - 24*B*C*a^7*b^5*d^
8*f^2 + 6*A*C*a^12*c^2*d^6*f^2 + 4*A*B*b^12*c^5*d^3*f^2 - 4*A*B*b^12*c^3*d^
5*f^2 - 90*A*C*a^8*b^4*d^8*f^2 - 80*B*C*a^3*b^9*c^8*f^2 + 54*A*C*a^10*b^2*d
^8*f^2 - 30*A*C*a^6*b^6*d^8*f^2 + 24*B*C*a^5*b^7*c^8*f^2 - 12*A*C*a^4*b^8*d
^8*f^2 - 112*A*B*a^9*b^3*d^8*f^2 - 66*A*C*a^4*b^8*c^8*f^2 + 54*A*C*a^2*b^10
*c^8*f^2 + 4*A*B*a^3*b^9*d^8*f^2 + 2*A*C*a^6*b^6*c^8*f^2 + 80*A*B*a^3*b^9*c
^8*f^2 - 24*A*B*a^5*b^7*c^8*f^2 + 726*C^2*a^6*b^6*c^4*d^4*f^2 - 402*C^2*a^7
*b^5*c^3*d^5*f^2 - 402*C^2*a^5*b^7*c^5*d^3*f^2 + 322*C^2*a^6*b^6*c^6*d^2*f^
2 + 322*C^2*a^6*b^6*c^2*d^6*f^2 - 222*C^2*a^7*b^5*c^5*d^3*f^2 - 222*C^2*a^5
*b^7*c^3*d^5*f^2 + 134*C^2*a^9*b^3*c^3*d^5*f^2 + 134*C^2*a^3*b^9*c^5*d^3*f^
2 - 66*C^2*a^10*b^2*c^2*d^6*f^2 - 66*C^2*a^2*b^10*c^6*d^2*f^2 + 52*C^2*a^9*
b^3*c^5*d^3*f^2 + 52*C^2*a^3*b^9*c^3*d^5*f^2 - 27*C^2*a^8*b^4*c^6*d^2*f^2 -
27*C^2*a^4*b^8*c^2*d^6*f^2 + 24*C^2*a^8*b^4*c^4*d^4*f^2 + 24*C^2*a^8*b^4*c
^2*d^6*f^2 + 24*C^2*a^4*b^8*c^6*d^2*f^2 + 24*C^2*a^4*b^8*c^4*d^4*f^2 - 15*C
^2*a^10*b^2*c^4*d^4*f^2 - 15*C^2*a^2*b^10*c^4*d^4*f^2 - 570*B^2*a^6*b^6*c^4
*d^4*f^2 + 366*B^2*a^7*b^5*c^3*d^5*f^2 + 318*B^2*a^5*b^7*c^5*d^3*f^2 - 262*
B^2*a^6*b^6*c^6*d^2*f^2 - 222*B^2*a^6*b^6*c^2*d^6*f^2 - 210*B^2*a^3*b^9*c^5
*d^3*f^2 + 186*B^2*a^7*b^5*c^5*d^3*f^2 + 162*B^2*a^5*b^7*c^3*d^5*f^2 - 142*
B^2*a^9*b^3*c^3*d^5*f^2 + 132*B^2*a^4*b^8*c^4*d^4*f^2 + 117*B^2*a^4*b^8*c^2
*d^6*f^2 + 102*B^2*a^2*b^10*c^6*d^2*f^2 - 96*B^2*a^3*b^9*c^3*d^5*f^2 + 90*B
^2*a^10*b^2*c^2*d^6*f^2 + 81*B^2*a^2*b^10*c^4*d^4*f^2 - 56*B^2*a^9*b^3*c^5*
d^3*f^2 + 48*B^2*a^8*b^4*c^4*d^4*f^2 + 48*B^2*a^4*b^8*c^6*d^2*f^2 + 45*B^2*
a^8*b^4*c^6*d^2*f^2 + 36*B^2*a^8*b^4*c^2*d^6*f^2 + 36*B^2*a^2*b^10*c^2*d^6*
f^2 + 33*B^2*a^10*b^2*c^4*d^4*f^2 + 822*A^2*a^6*b^6*c^4*d^4*f^2 - 594*A^2*a
^7*b^5*c^3*d^5*f^2 + 498*A^2*a^6*b^6*c^2*d^6*f^2 - 498*A^2*a^5*b^7*c^5*d^3*
f^2 - 414*A^2*a^5*b^7*c^3*d^5*f^2 + 354*A^2*a^6*b^6*c^6*d^2*f^2 - 318*A^2*a
^7*b^5*c^5*d^3*f^2 + 144*A^2*a^8*b^4*c^2*d^6*f^2 + 102*A^2*a^3*b^9*c^5*d^3*
f^2 + 84*A^2*a^4*b^8*c^4*d^4*f^2 + 81*A^2*a^4*b^8*c^2*d^6*f^2 + 72*A^2*a^8*
b^4*c^4*d^4*f^2 + 70*A^2*a^9*b^3*c^3*d^5*f^2 - 66*A^2*a^2*b^10*c^6*d^2*f^2
+ 48*A^2*a^4*b^8*c^6*d^2*f^2 - 42*A^2*a^10*b^2*c^2*d^6*f^2 + 24*A^2*a^2*b^1
0*c^2*d^6*f^2 + 20*A^2*a^9*b^3*c^5*d^3*f^2 - 15*A^2*a^10*b^2*c^4*d^4*f^2 -
15*A^2*a^8*b^4*c^6*d^2*f^2 - 15*A^2*a^2*b^10*c^4*d^4*f^2 - 12*A^2*a^3*b^9*c
^3*d^5*f^2 - 8*B*C*b^12*c^7*d*f^2 + 4*B*C*a^12*c*d^7*f^2 - 24*B*C*a^11*b*d^
8*f^2 + 8*A*B*b^12*c^7*d*f^2 - 8*A*B*b^12*c*d^7*f^2 + 24*B*C*a*b^11*c^8*f^2
- 8*A*B*a^12*c*d^7*f^2 + 12*A*B*a^11*b*d^8*f^2 - 24*A*B*a*b^11*c^8*f^2 - 1

74*C^2*a^7*b^5*c*d^7*f^2 - 174*C^2*a^5*b^7*c^7*d*f^2 + 82*C^2*a^9*b^3*c*d^7
*f^2 + 82*C^2*a^3*b^9*c^7*d*f^2 + 6*C^2*a^11*b*c^3*d^5*f^2 + 6*C^2*a^7*b^5*
c^7*d*f^2 + 6*C^2*a^5*b^7*c*d^7*f^2 + 6*C^2*a*b^11*c^5*d^3*f^2 + 162*B^2*a^
7*b^5*c*d^7*f^2 + 138*B^2*a^5*b^7*c^7*d*f^2 - 118*B^2*a^3*b^9*c^7*d*f^2 - 8
6*B^2*a^9*b^3*c*d^7*f^2 - 30*B^2*a*b^11*c^5*d^3*f^2 - 18*B^2*a^7*b^5*c^7*d*
f^2 - 18*B^2*a^5*b^7*c*d^7*f^2 - 12*B^2*a*b^11*c^3*d^5*f^2 - 6*B^2*a^11*b*c
^3*d^5*f^2 - 4*B^2*a^3*b^9*c*d^7*f^2 - 270*A^2*a^7*b^5*c*d^7*f^2 - 174*A^2*
a^5*b^7*c^7*d*f^2 - 90*A^2*a^5*b^7*c*d^7*f^2 + 82*A^2*a^3*b^9*c^7*d*f^2 + 5
0*A^2*a^9*b^3*c*d^7*f^2 - 32*A^2*a^3*b^9*c*d^7*f^2 + 6*A^2*a^11*b*c^3*d^5*f
^2 + 6*A^2*a^7*b^5*c^7*d*f^2 + 6*A^2*a*b^11*c^5*d^3*f^2 + 6*C^2*a^11*b*c*d^
7*f^2 + 6*C^2*a*b^11*c^7*d*f^2 - 18*B^2*a*b^11*c^7*d*f^2 - 6*B^2*a^11*b*c*d
^7*f^2 + 6*A^2*a^11*b*c*d^7*f^2 + 6*A^2*a*b^11*c^7*d*f^2 - 6*A*C*b^12*c^8*f
^2 - 2*A*C*a^12*d^8*f^2 + 4*C^2*b^12*c^4*d^4*f^2 + 3*C^2*b^12*c^6*d^2*f^2 +
 4*C^2*a^12*c^4*d^4*f^2 + 4*B^2*b^12*c^4*d^4*f^2 + 4*B^2*b^12*c^2*d^6*f^2 +
 3*C^2*a^12*c^2*d^6*f^2 + 3*B^2*b^12*c^6*d^2*f^2 + 33*C^2*a^8*b^4*d^8*f^2 -
 27*C^2*a^10*b^2*d^8*f^2 - 4*A^2*b^12*c^4*d^4*f^2 + 3*B^2*a^12*c^2*d^6*f^2
- C^2*a^6*b^6*d^8*f^2 - A^2*b^12*c^6*d^2*f^2 + 33*C^2*a^4*b^8*c^8*f^2 + 33*
B^2*a^10*b^2*d^8*f^2 - 27*C^2*a^2*b^10*c^8*f^2 - 27*B^2*a^8*b^4*d^8*f^2 + 3
*B^2*a^6*b^6*d^8*f^2 - C^2*a^6*b^6*c^8*f^2 - A^2*a^12*c^2*d^6*f^2 + 117*A^2
*a^8*b^4*d^8*f^2 + 111*A^2*a^6*b^6*d^8*f^2 + 72*A^2*a^4*b^8*d^8*f^2 + 33*B^
2*a^2*b^10*c^8*f^2 - 27*B^2*a^4*b^8*c^8*f^2 + 24*A^2*a^2*b^10*d^8*f^2 + 3*B
^2*a^6*b^6*c^8*f^2 - 3*A^2*a^10*b^2*d^8*f^2 + 33*A^2*a^4*b^8*c^8*f^2 - 27*A
^2*a^2*b^10*c^8*f^2 - A^2*a^6*b^6*c^8*f^2 + 3*C^2*b^12*c^8*f^2 + 3*C^2*a^12
*d^8*f^2 + 4*A^2*b^12*d^8*f^2 - B^2*b^12*c^8*f^2 - B^2*a^12*d^8*f^2 + 3*A^2
*b^12*c^8*f^2 + 3*A^2*a^12*d^8*f^2 - 24*A*B*C*a*b^8*c*d^6*f + 342*A*B*C*a^4
*b^5*c^2*d^5*f - 186*A*B*C*a^5*b^4*c^3*d^4*f - 66*A*B*C*a^2*b^7*c^4*d^3*f +
 48*A*B*C*a^2*b^7*c^2*d^5*f + 42*A*B*C*a^6*b^3*c^2*d^5*f + 26*A*B*C*a^3*b^6
*c^5*d^2*f + 24*A*B*C*a^6*b^3*c^4*d^3*f - 18*A*B*C*a^7*b^2*c^3*d^4*f - 18*A
*B*C*a^4*b^5*c^4*d^3*f - 8*A*B*C*a^3*b^6*c^3*d^4*f + 6*A*B*C*a^5*b^4*c^5*d^
2*f - 128*A*B*C*a^3*b^6*c*d^6*f + 126*A*B*C*a^7*b^2*c*d^6*f + 72*A*B*C*a*b^
8*c^3*d^4*f - 36*A*B*C*a^8*b*c^2*d^5*f - 36*A*B*C*a*b^8*c^5*d^2*f + 30*A*B*
C*a^2*b^7*c^6*d*f - 12*A*B*C*a^5*b^4*c*d^6*f - 12*A*B*C*a^4*b^5*c^6*d*f - 2
1*B^2*C*a^8*b*c*d^6*f - 3*B^2*C*a*b^8*c^6*d*f + 21*A^2*C*a^8*b*c*d^6*f - 21
*A*C^2*a^8*b*c*d^6*f - 9*A^2*C*a*b^8*c^6*d*f + 9*A*C^2*a*b^8*c^6*d*f + 36*A
^2*B*a*b^8*c*d^6*f + 21*A*B^2*a^8*b*c*d^6*f + 3*A*B^2*a*b^8*c^6*d*f + 16*A*
B*C*b^9*c^4*d^3*f - 16*A*B*C*b^9*c^2*d^5*f - 78*A*B*C*a^6*b^3*d^7*f + 24*A*
B*C*a^4*b^5*d^7*f + 2*A*B*C*a^3*b^6*c^7*f - 237*B^2*C*a^4*b^5*c^3*d^4*f + 1
65*B*C^2*a^5*b^4*c^3*d^4*f + 92*B^2*C*a^3*b^6*c^2*d^5*f - 81*B^2*C*a^7*b^2*
c^2*d^5*f + 77*B^2*C*a^3*b^6*c^4*d^3*f - 75*B*C^2*a^4*b^5*c^2*d^5*f + 69*B^
2*C*a^5*b^4*c^4*d^3*f + 69*B*C^2*a^4*b^5*c^4*d^3*f - 68*B*C^2*a^3*b^6*c^3*d
^4*f - 63*B^2*C*a^4*b^5*c^5*d^2*f - 61*B*C^2*a^6*b^3*c^2*d^5*f + 57*B*C^2*a
^2*b^7*c^4*d^3*f - 53*B*C^2*a^3*b^6*c^5*d^2*f - 44*B*C^2*a^6*b^3*c^4*d^3*f
- 36*B^2*C*a^2*b^7*c^3*d^4*f + 35*B^2*C*a^6*b^3*c^3*d^4*f - 33*B^2*C*a^5*b^
4*c^2*d^5*f + 33*B^2*C*a^2*b^7*c^5*d^2*f + 33*B*C^2*a^7*b^2*c^3*d^4*f - 12*
B^2*C*a^7*b^2*c^4*d^3*f + 9*B*C^2*a^5*b^4*c^5*d^2*f + 4*B^2*C*a^6*b^3*c^5*d

^2*f + 225*A^2*C*a^5*b^4*c^2*d^5*f - 105*A*C^2*a^5*b^4*c^2*d^5*f - 99*A^2*C
*a^4*b^5*c^3*d^4*f - 81*A^2*C*a^4*b^5*c^5*d^2*f + 67*A^2*C*a^3*b^6*c^4*d^3*
f - 59*A*C^2*a^3*b^6*c^4*d^3*f - 57*A*C^2*a^7*b^2*c^2*d^5*f + 57*A*C^2*a^2*
b^7*c^5*d^2*f + 51*A^2*C*a^5*b^4*c^4*d^3*f + 48*A^2*C*a^2*b^7*c^3*d^4*f + 4
5*A*C^2*a^4*b^5*c^5*d^2*f - 35*A^2*C*a^6*b^3*c^3*d^4*f + 33*A^2*C*a^7*b^2*c
^2*d^5*f - 33*A^2*C*a^2*b^7*c^5*d^2*f + 33*A*C^2*a^5*b^4*c^4*d^3*f + 27*A*C
^2*a^6*b^3*c^3*d^4*f + 24*A*C^2*a^3*b^6*c^2*d^5*f - 24*A*C^2*a^2*b^7*c^3*d^
4*f - 21*A*C^2*a^4*b^5*c^3*d^4*f - 16*A^2*C*a^3*b^6*c^2*d^5*f - 243*A^2*B*a
^4*b^5*c^2*d^5*f - 156*A*B^2*a^3*b^6*c^2*d^5*f + 141*A*B^2*a^4*b^5*c^3*d^4*
f + 108*A^2*B*a^3*b^6*c^3*d^4*f - 105*A*B^2*a^3*b^6*c^4*d^3*f + 84*A*B^2*a^
2*b^7*c^3*d^4*f + 81*A*B^2*a^5*b^4*c^2*d^5*f + 51*A^2*B*a^6*b^3*c^2*d^5*f -
51*A^2*B*a^4*b^5*c^4*d^3*f - 48*A^2*B*a^2*b^7*c^2*d^5*f + 45*A^2*B*a^5*b^4
*c^3*d^4*f + 39*A*B^2*a^4*b^5*c^5*d^2*f - 35*A*B^2*a^6*b^3*c^3*d^4*f + 33*A
*B^2*a^7*b^2*c^2*d^5*f + 27*A^2*B*a^3*b^6*c^5*d^2*f - 21*A*B^2*a^5*b^4*c^4*
d^3*f + 20*A^2*B*a^6*b^3*c^4*d^3*f - 15*A^2*B*a^7*b^2*c^3*d^4*f - 15*A^2*B*
a^5*b^4*c^5*d^2*f + 9*A^2*B*a^2*b^7*c^4*d^3*f + 3*A*B^2*a^2*b^7*c^5*d^2*f +
2*A*B*C*b^9*c^6*d*f - 6*A*B*C*a^9*c*d^6*f + 18*A*B*C*a^8*b*d^7*f - 6*A*B*C
*a*b^8*c^7*f + 63*B^2*C*a^6*b^3*c*d^6*f - 48*B^2*C*a*b^8*c^4*d^3*f + 42*B*C
^2*a^8*b*c^2*d^5*f + 42*B*C^2*a^5*b^4*c*d^6*f - 39*B*C^2*a^7*b^2*c*d^6*f +
30*B*C^2*a*b^8*c^5*d^2*f - 24*B^2*C*a^4*b^5*c*d^6*f - 24*B*C^2*a*b^8*c^3*d^
4*f + 17*B^2*C*a^3*b^6*c^6*d*f - 15*B*C^2*a^2*b^7*c^6*d*f + 12*B^2*C*a^8*b*
c^3*d^4*f + 12*B^2*C*a*b^8*c^2*d^5*f + 6*B*C^2*a^4*b^5*c^6*d*f - 192*A^2*C*
a^4*b^5*c*d^6*f - 99*A^2*C*a^6*b^3*c*d^6*f + 84*A*C^2*a^4*b^5*c*d^6*f + 59*
A*C^2*a^6*b^3*c*d^6*f + 51*A^2*C*a^3*b^6*c^6*d*f - 51*A*C^2*a^3*b^6*c^6*d*f
- 36*A^2*C*a*b^8*c^2*d^5*f - 24*A*C^2*a*b^8*c^4*d^3*f + 24*A*C^2*a*b^8*c^2
*d^5*f + 12*A^2*C*a*b^8*c^4*d^3*f + 12*A*C^2*a^8*b*c^3*d^4*f + 160*A^2*B*a^
3*b^6*c*d^6*f - 99*A*B^2*a^6*b^3*c*d^6*f - 87*A^2*B*a^7*b^2*c*d^6*f - 72*A*
B^2*a^4*b^5*c*d^6*f - 48*A*B^2*a*b^8*c^2*d^5*f - 36*A^2*B*a*b^8*c^3*d^4*f +
24*A*B^2*a*b^8*c^4*d^3*f - 17*A*B^2*a^3*b^6*c^6*d*f - 15*A^2*B*a^2*b^7*c^6
*d*f + 12*A*B^2*a^2*b^7*c*d^6*f + 6*A^2*B*a^8*b*c^2*d^5*f - 6*A^2*B*a^5*b^4
*c*d^6*f + 6*A^2*B*a^4*b^5*c^6*d*f + 6*A^2*B*a*b^8*c^5*d^2*f + 12*B^2*C*b^9
*c^3*d^4*f - 12*B*C^2*b^9*c^4*d^3*f - 12*A^2*C*b^9*c^3*d^4*f - 8*A*C^2*b^9*
c^5*d^2*f + 8*A*C^2*b^9*c^3*d^4*f + 4*B^2*C*a^9*c^2*d^5*f + 4*A^2*C*b^9*c^5
*d^2*f - 4*B*C^2*a^9*c^3*d^4*f + 12*A^2*B*b^9*c^2*d^5*f - 8*A*B^2*b^9*c^3*d
^4*f - 4*A^2*B*b^9*c^4*d^3*f + 4*A*C^2*a^9*c^2*d^5*f + 3*B^2*C*a^7*b^2*d^7*
f - B*C^2*a^6*b^3*d^7*f + 96*A^2*C*a^5*b^4*d^7*f - 39*A^2*C*a^7*b^2*d^7*f -
36*A*C^2*a^5*b^4*d^7*f + 32*A^2*C*a^3*b^6*d^7*f + 15*A*C^2*a^7*b^2*d^7*f -
3*B^2*C*a^2*b^7*c^7*f - B*C^2*a^3*b^6*c^7*f + 111*A^2*B*a^6*b^3*d^7*f - 39
*A*B^2*a^7*b^2*d^7*f + 24*A*B^2*a^5*b^4*d^7*f - 9*A^2*C*a^2*b^7*c^7*f + 9*A
*C^2*a^2*b^7*c^7*f - 4*A*B^2*a^3*b^6*d^7*f + 3*A*B^2*a^2*b^7*c^7*f - A^2*B*
a^3*b^6*c^7*f + 3*C^3*a^8*b*c*d^6*f - 3*C^3*a*b^8*c^6*d*f - 3*A^3*a^8*b*c*d
^6*f + 3*A^3*a*b^8*c^6*d*f - B*C^2*b^9*c^6*d*f + 4*A^2*C*b^9*c*d^6*f + 3*B*
C^2*a^9*c*d^6*f + 8*A*B^2*b^9*c*d^6*f + 3*B*C^2*a^8*b*d^7*f - A^2*B*b^9*c^6
*d*f + 12*A^2*C*a*b^8*d^7*f + 3*B*C^2*a*b^8*c^7*f - A^2*B*a^9*c*d^6*f - 9*A
^2*B*a^8*b*d^7*f + 3*A^2*B*a*b^8*c^7*f - 39*C^3*a^5*b^4*c^4*d^3*f + 39*C^3*

a^4*b^5*c^3*d^4*f + 27*C^3*a^7*b^2*c^2*d^5*f - 27*C^3*a^2*b^7*c^5*d^2*f - 1
7*C^3*a^6*b^3*c^3*d^4*f + 17*C^3*a^3*b^6*c^4*d^3*f + 3*C^3*a^5*b^4*c^2*d^5*
f - 3*C^3*a^4*b^5*c^5*d^2*f - 63*B^3*a^5*b^4*c^3*d^4*f + 57*B^3*a^4*b^5*c^2
*d^5*f - 51*B^3*a^2*b^7*c^4*d^3*f + 48*B^3*a^3*b^6*c^3*d^4*f + 31*B^3*a^6*b
^3*c^2*d^5*f + 27*B^3*a^3*b^6*c^5*d^2*f + 16*B^3*a^6*b^3*c^4*d^3*f - 15*B^3
*a^5*b^4*c^5*d^2*f - 12*B^3*a^2*b^7*c^2*d^5*f + 9*B^3*a^4*b^5*c^4*d^3*f - 3
*B^3*a^7*b^2*c^3*d^4*f - 123*A^3*a^5*b^4*c^2*d^5*f + 81*A^3*a^4*b^5*c^3*d^4
*f - 45*A^3*a^5*b^4*c^4*d^3*f + 39*A^3*a^4*b^5*c^5*d^2*f + 25*A^3*a^6*b^3*c
^3*d^4*f - 25*A^3*a^3*b^6*c^4*d^3*f - 24*A^3*a^2*b^7*c^3*d^4*f - 8*A^3*a^3*
b^6*c^2*d^5*f - 3*A^3*a^7*b^2*c^2*d^5*f + 3*A^3*a^2*b^7*c^5*d^2*f - 17*C^3*
a^6*b^3*c*d^6*f + 17*C^3*a^3*b^6*c^6*d*f - 12*C^3*a^8*b*c^3*d^4*f + 12*C^3*
a*b^8*c^4*d^3*f + 24*B^3*a*b^8*c^3*d^4*f + 21*B^3*a^7*b^2*c*d^6*f - 18*B^3*
a^5*b^4*c*d^6*f - 15*B^3*a^2*b^7*c^6*d*f - 6*B^3*a^8*b*c^2*d^5*f + 6*B^3*a^
4*b^5*c^6*d*f + 6*B^3*a*b^8*c^5*d^2*f + 4*B^3*a^3*b^6*c*d^6*f + 108*A^3*a^4
*b^5*c*d^6*f + 57*A^3*a^6*b^3*c*d^6*f - 17*A^3*a^3*b^6*c^6*d*f + 12*A^3*a*b
^8*c^2*d^5*f + 4*C^3*b^9*c^5*d^2*f - 4*C^3*a^9*c^2*d^5*f - 4*B^3*b^9*c^2*d^
5*f + 4*A^3*b^9*c^3*d^4*f + 3*C^3*a^7*b^2*d^7*f - 3*C^3*a^2*b^7*c^7*f - B^3
*a^6*b^3*d^7*f - 60*A^3*a^5*b^4*d^7*f - 32*A^3*a^3*b^6*d^7*f + 21*A^3*a^7*b
^2*d^7*f - B^3*a^3*b^6*c^7*f + 3*A^3*a^2*b^7*c^7*f - B^3*b^9*c^6*d*f - 4*A^
3*b^9*c*d^6*f - B^3*a^9*c*d^6*f + 3*B^3*a^8*b*d^7*f - 12*A^3*a*b^8*d^7*f +
3*B^3*a*b^8*c^7*f - B^2*C*a^9*d^7*f - 4*A^2*B*b^9*d^7*f + 3*A^2*C*b^9*c^7*f
 - 3*A*C^2*b^9*c^7*f - A*C^2*a^9*d^7*f - A*B^2*b^9*c^7*f - C^3*a^9*d^7*f -
A^3*b^9*c^7*f + B^2*C*b^9*c^7*f + A^2*C*a^9*d^7*f + A*B^2*a^9*d^7*f + C^3*b
^9*c^7*f + A^3*a^9*d^7*f - 6*A*B^2*C*a^5*b*c*d^5 - 21*A^2*B*C*a^3*b^3*c^2*d
^4 + 21*A*B*C^2*a^3*b^3*c^2*d^4 + 12*A*B^2*C*a^4*b^2*c^2*d^4 - 12*A*B^2*C*a
^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^4*b^2*c^3*d^3 + 3
*A^2*B*C*a^4*b^2*c^3*d^3 + 3*A^2*B*C*a^2*b^4*c^3*d^3 + 3*A*B^2*C*a^2*b^4*c^
4*d^2 + 3*A*B*C^2*a^2*b^4*c^3*d^3 + 2*A*B*C^2*a^3*b^3*c^4*d^2 - A^2*B*C*a^3
*b^3*c^4*d^2 + 18*A^2*B*C*a^2*b^4*c*d^5 + 10*A*B^2*C*a^3*b^3*c*d^5 + 9*A^2*
B*C*a^4*b^2*c*d^5 - 9*A*B*C^2*a^4*b^2*c*d^5 - 9*A*B*C^2*a^2*b^4*c*d^5 - 6*A
^2*B*C*a*b^5*c^2*d^4 + 6*A*B^2*C*a*b^5*c^3*d^3 + 6*A*B*C^2*a^5*b*c^2*d^4 -
6*A*B*C^2*a*b^5*c^4*d^2 - 3*A^2*B*C*a^5*b*c^2*d^4 + 3*A^2*B*C*a*b^5*c^4*d^2
 + 3*A*B*C^2*a*b^5*c^2*d^4 - 3*B^3*C*a^5*b*c^2*d^4 + 3*B^3*C*a^4*b^2*c*d^5
+ 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a^5*b*c*d^5 - 3*B*C^3*a^5*b*c^2*d^4 + 3
*B*C^3*a^4*b^2*c*d^5 + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a^3*b^3*c*d^5 + 8*A
*C^3*a^3*b^3*c*d^5 - 9*A^3*B*a^2*b^4*c*d^5 - 9*A*B^3*a^2*b^4*c*d^5 - 3*A^3*
B*a^4*b^2*c*d^5 + 3*A^3*B*a*b^5*c^2*d^4 + 3*A^2*B^2*a^5*b*c*d^5 - 3*A*B^3*a
^4*b^2*c*d^5 + 3*A*B^3*a*b^5*c^2*d^4 + 5*A*B*C^2*b^6*c^3*d^3 - 4*A^2*B*C*b^
6*c^3*d^3 - A*B^2*C*b^6*c^4*d^2 - 3*A*B^2*C*a^4*b^2*d^6 - 2*A^2*B*C*a^3*b^3
*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^4*b^2*c^2*d^4 + 6*B^2*C^2*a^
2*b^4*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 24*A^2*C^2*a^3*b^3*c^3*d^3 - 15
*A^2*C^2*a^4*b^2*c^2*d^4 - 9*A^2*C^2*a^2*b^4*c^4*d^2 + 3*A^2*C^2*a^2*b^4*c^
2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^4*b^2*c^2*d^4 + 4*A^2*B*C*b
^6*c*d^5 - 2*A*B*C^2*b^6*c*d^5 + 2*A*B*C^2*a^6*c*d^5 - A^2*B*C*a^6*c*d^5 +
6*A^2*B*C*a^5*b*d^6 - 3*A*B*C^2*a^5*b*d^6 - 7*B^3*C*a^3*b^3*c^2*d^4 - 7*B*C

^3*a^3*b^3*c^2*d^4 + 3*B^3*C*a^4*b^2*c^3*d^3 - 3*B^3*C*a^2*b^4*c^3*d^3 - 3*B^2*C^2*a*b^5*c^3*d^3 + 3*B*C^3*a^4*b^2*c^3*d^3 - 3*B*C^3*a^2*b^4*c^3*d^3 - B^3*C*a^3*b^3*c^4*d^2 - B^2*C^2*a^3*b^3*c*d^5 - B*C^3*a^3*b^3*c^4*d^2 - 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^4*b^2*c^2*d^4 + 9*A*C^3*a^2*b^4*c^4*d^2 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^4*b^2*c^2*d^4 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^2*b^4*c^4*d^2 - 9*A^2*B^2*a^3*b^3*c*d^5 + 7*A^3*B*a^3*b^3*c^2*d^4 + 7*A*B^3*a^3*b^3*c^2*d^4 - 3*A^3*B*a^2*b^4*c^3*d^3 - 3*A^2*B^2*a*b^5*c^3*d^3 - 3*A*B^3*a^2*b^4*c^3*d^3 - 5*A^2*C^2*b^6*c^2*d^4 + 3*A^2*C^2*b^6*c^4*d^2 + 12*A^2*C^2*a^4*b^2*d^6 + 3*A^2*C^2*a^2*b^4*d^6 + 6*A^2*B^2*a^4*b^2*d^6 + 3*A^2*B^2*a^2*b^4*d^6 + A*B*C^2*a^3*b^3*d^6 - 3*B^4*a*b^5*c^3*d^3 - B^4*a^3*b^3*c*d^5 + A^2*B^2*a^3*b^3*c^3*d^3 - 8*A^4*a^3*b^3*c*d^5 - 2*B^3*C*b^6*c^3*d^3 - 2*B*C^3*b^6*c^3*d^3 + 4*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 + 2*A*C^3*b^6*c^2*d^4 - A^3*C*b^6*c^4*d^2 - 2*A*C^3*a^6*c^2*d^4 - 15*A^3*C*a^4*b^2*d^6 - 6*A^3*C*a^2*b^4*d^6 - 3*A*C^3*a^4*b^2*d^6 + 3*B^4*a^5*b*c*d^5 - B^3*C*a^6*c*d^5 - B*C^3*a^6*c*d^5 - 2*A^3*B*b^6*c*d^5 - 2*A*B^3*b^6*c*d^5 - 3*A^3*B*a^5*b*d^6 - 3*A*B^3*a^5*b*d^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2 + 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^4*b^2*c^2*d^4 + 3*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*b^6*c^4*d^2 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^6*c^2*d^4 + A^2*C^2*a^6*c^2*d^4 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 - A^4*b^6*c^2*d^4 + 6*A^4*a^4*b^2*d^6 + 3*A^4*a^2*b^4*d^6 - 2*A^2*C^2*a^6*d^6 + A*B^2*C*a^6*d^6 + B^4*a^3*b^3*c^3*d^3 + A^3*C*a^6*d^6 + A*C^3*a^6*d^6 + C^4*b^6*c^4*d^2 + C^4*a^6*c^2*d^4 + B^4*b^6*c^2*d^4 + A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k)*((4*a^5*b^12*d^9 + 12*a^7*b^10*d^9 + 8*a^9*b^8*d^9 - 8*a^11*b^6*d^9 - 12*a^13*b^4*d^9 - 4*a^15*b^2*d^9 + 4*b^17*c^5*d^4 - 4*b^17*c^7*d^2 - 12*a*b^16*c^4*d^5 + 28*a*b^16*c^6*d^3 + 32*a^3*b^14*c^8*d - 12*a^4*b^13*c*d^8 + 48*a^5*b^12*c^8*d - 20*a^6*b^11*c*d^8 + 32*a^7*b^10*c^8*d + 48*a^8*b^9*c*d^8 + 8*a^9*b^8*c^8*d + 152*a^10*b^7*c*d^8 + 148*a^12*b^5*c*d^8 + 60*a^14*b^3*c*d^8 + 8*a^2*b^15*c^3*d^6 - 44*a^2*b^15*c^5*d^4 - 52*a^2*b^15*c^7*d^2 + 8*a^3*b^14*c^2*d^7 - 12*a^3*b^14*c^4*d^5 + 172*a^3*b^14*c^6*d^3 + 68*a^4*b^13*c^3*d^6 - 248*a^4*b^13*c^5*d^4 - 168*a^4*b^13*c^7*d^2 - 28*a^5*b^12*c^2*d^7 + 40*a^5*b^12*c^4*d^5 + 408*a^5*b^12*c^6*d^3 + 252*a^6*b^11*c^3*d^6 - 472*a^6*b^11*c^5*d^4 - 232*a^6*b^11*c^7*d^2 - 228*a^7*b^10*c^2*d^7 + 40*a^7*b^10*c^4*d^5 + 472*a^7*b^10*c^6*d^3 + 488*a^8*b^9*c^3*d^6 - 428*a^8*b^9*c^5*d^4 - 148*a^8*b^9*c^7*d^2 - 472*a^9*b^8*c^2*d^7 - 60*a^9*b^8*c^4*d^5 + 268*a^9*b^8*c^6*d^3 + 512*a^10*b^7*c^3*d^6 - 188*a^10*b^7*c^5*d^4 - 36*a^10*b^7*c^7*d^2 - 448*a^11*b^6*c^2*d^7 - 92*a^11*b^6*c^4*d^5 + 60*a^11*b^6*c^6*d^3 + 276*a^12*b^5*c^3*d^6 - 32*a^12*b^5*c^5*d^4 - 204*a^13*b^4*c^2*d^7 - 32*a^13*b^4*c^4*d^5 + 60*a^14*b^3*c^3*d^6 - 36*a^15*b^2*c^2*d^7 + 8*a*b^16*c^8*d + 8*a^16*b*c*d^8)/(a^12*d^4 + b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4 + a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9*c*d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^5*c*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^10*c^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b

^4*c^2*d^2 + 6*a^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3) + (tan(e + f*x)*(6*a^16*b*d^9 + 6*b^17*c^8*d + 8*a^4*b^13*d^9 + 38*a^6*b^11*d^9 + 78*a^8*b^9*d^9 + 92*a^10*b^7*d^9 + 68*a^12*b^5*d^9 + 30*a^14*b^3*d^9 + 8*b^17*c^4*d^5 + 6*b^17*c^6*d^3 - 32*a*b^16*c^3*d^6 - 20*a*b^16*c^5*d^4 - 20*a*b^16*c^7*d^2 + 22*a^2*b^15*c^8*d - 32*a^3*b^14*c*d^8 + 28*a^4*b^13*c^8*d - 148*a^5*b^12*c*d^8 + 12*a^6*b^11*c^8*d - 292*a^7*b^10*c*d^8 - 2*a^8*b^9*c^8*d - 328*a^9*b^8*c*d^8 - 2*a^10*b^7*c^8*d - 232*a^11*b^6*c*d^8 - 100*a^13*b^4*c*d^8 - 20*a^15*b^2*c*d^8 - 2*a^16*b*c^2*d^7 + 48*a^2*b^15*c^2*d^7 + 58*a^2*b^15*c^4*d^5 + 32*a^2*b^15*c^6*d^3 - 152*a^3*b^14*c^3*d^6 - 28*a^3*b^14*c^5*d^4 - 68*a^3*b^14*c^7*d^2 + 218*a^4*b^13*c^2*d^7 + 60*a^4*b^13*c^4*d^5 + 38*a^4*b^13*c^6*d^3 - 236*a^5*b^12*c^3*d^6 + 128*a^5*b^12*c^5*d^4 - 72*a^5*b^12*c^7*d^2 + 400*a^6*b^11*c^2*d^7 - 210*a^6*b^11*c^4*d^5 - 48*a^6*b^11*c^6*d^3 - 52*a^7*b^10*c^3*d^6 + 392*a^7*b^10*c^5*d^4 - 8*a^7*b^10*c^7*d^2 + 378*a^8*b^9*c^2*d^7 - 560*a^8*b^9*c^4*d^5 - 142*a^8*b^9*c^6*d^3 + 232*a^9*b^8*c^3*d^6 + 428*a^9*b^8*c^5*d^4 + 28*a^9*b^8*c^7*d^2 + 192*a^10*b^7*c^2*d^7 - 522*a^10*b^7*c^4*d^5 - 112*a^10*b^7*c^6*d^3 + 256*a^11*b^6*c^3*d^6 + 212*a^11*b^6*c^5*d^4 + 12*a^11*b^6*c^7*d^2 + 46*a^12*b^5*c^2*d^7 - 212*a^12*b^5*c^4*d^5 - 30*a^12*b^5*c^6*d^3 + 100*a^13*b^4*c^3*d^6 + 40*a^13*b^4*c^5*d^4 - 30*a^14*b^3*c^4*d^5 + 12*a^15*b^2*c^3*d^6))/(a^12*d^4 + b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4 + a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9*c*d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^5*c*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^10*c^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d^2 + 6*a^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3)) + (tan(e + f*x)*(3*A*a^13*b*d^8 - 3*A*b^14*c^7*d + C*a^13*b*d^8 + 3*C*b^14*c^7*d + 8*A*a^3*b^11*d^8 + 24*A*a^5*b^9*d^8 + 51*A*a^7*b^7*d^8 + 65*A*a^9*b^5*d^8 + 33*A*a^11*b^3*d^8 - 4*B*a^4*b^10*d^8 + 7*B*a^6*b^8*d^8 + 21*B*a^8*b^6*d^8 + 5*B*a^10*b^4*d^8 - 5*B*a^12*b^2*d^8 + 8*A*b^14*c^3*d^5 - 8*A*b^14*c^5*d^3 + 12*C*a^5*b^9*d^8 + 13*C*a^7*b^7*d^8 - 9*C*a^9*b^5*d^8 - 9*C*a^11*b^3*d^8 - 12*B*b^14*c^4*d^4 - B*b^14*c^6*d^2 + 12*C*b^14*c^5*d^3 - 8*A*a*b^13*c^2*d^6 + 8*A*a*b^13*c^4*d^4 + 13*A*a*b^13*c^6*d^2 - 8*A*a^2*b^12*c*d^7 - A*a^2*b^12*c^7*d + 8*A*a^4*b^10*c*d^7 + 7*A*a^4*b^10*c^7*d + 3*A*a^6*b^8*c*d^7 + 5*A*a^6*b^8*c^7*d - 63*A*a^8*b^6*c*d^7 - 63*A*a^10*b^4*c*d^7 - 13*A*a^12*b^2*c*d^7 + 24*B*a*b^13*c^3*d^5 + 30*B*a*b^13*c^5*d^3 + 8*B*a^3*b^11*c*d^7 + 13*B*a^3*b^11*c^7*d - 50*B*a^5*b^9*c*d^7 + 5*B*a^5*b^9*c^7*d - 143*B*a^7*b^7*c*d^7 - B*a^7*b^7*c^7*d - 105*B*a^9*b^5*c*d^7 - 21*B*a^11*b^3*c*d^7 - 12*C*a*b^13*c^4*d^4 - 13*C*a*b^13*c^6*d^2 + C*a^2*b^12*c^7*d - 44*C*a^4*b^10*c*d^7 - 7*C*a^4*b^10*c^7*d - 67*C*a^6*b^8*c*d^7 - 5*C*a^6*b^8*c^7*d + 7*C*a^8*b^6*c*d^7 + 39*C*a^10*b^4*c*d^7 + 9*C*a^12*b^2*c*d^7 + 64*A*a^2*b^12*c^3*d^5 - 7*A*a^2*b^12*c^5*d^3 - 96*A*a^3*b^11*c^2*d^6 - 87*A*a^3*b^11*c^4*d^4 - A*a^3*b^11*c^6*d^2 + 263*A*a^4*b^10*c^3*d^5 + 67*A*a^4*b^10*c^5*d^3 - 233*A*a^5*b^9*c^2*d^6 - 253*A*a^5*b^9*c^4*d^4 - 41*A*a^5*b^9*c^6*d^2 + 381*A*a^6*b^8*c^3*d^5 + 123*A*a^6*b^8*c^5*d^3 - 195*A*a^7*b^7*c^2*d^6 - 213*A*a^7*b^7*c^4*d^4 - 27*A*a^7*b^7*c^6*d^2 + 189*A*a^8*b^6*c^3*d^5 + 57*A*a^8*b^6*c^5*d^3 - 35*A*a^9*b^5*c^2*d^

6 - 55*A*a^9*b^5*c^4*d^4 + 15*A*a^10*b^4*c^3*d^5 + 15*A*a^11*b^3*c^2*d^6 -
16*B*a^2*b^12*c^2*d^6 - 119*B*a^2*b^12*c^4*d^4 - 37*B*a^2*b^12*c^6*d^2 + 11
6*B*a^3*b^11*c^3*d^5 + 115*B*a^3*b^11*c^5*d^3 + 17*B*a^4*b^10*c^2*d^6 - 209
*B*a^4*b^10*c^4*d^4 - 65*B*a^4*b^10*c^6*d^2 + 85*B*a^5*b^9*c^3*d^5 + 125*B*
a^5*b^9*c^5*d^3 + 161*B*a^6*b^8*c^2*d^6 - 89*B*a^6*b^8*c^4*d^4 - 23*B*a^6*b
^8*c^6*d^2 - 97*B*a^7*b^7*c^3*d^5 + 25*B*a^7*b^7*c^5*d^3 + 213*B*a^8*b^6*c^
2*d^6 + 33*B*a^8*b^6*c^4*d^4 + 6*B*a^8*b^6*c^6*d^2 - 105*B*a^9*b^5*c^3*d^5
- 15*B*a^9*b^5*c^5*d^3 + 91*B*a^10*b^4*c^2*d^6 + 20*B*a^10*b^4*c^4*d^4 - 15
*B*a^11*b^3*c^3*d^5 + 6*B*a^12*b^2*c^2*d^6 - 32*C*a^2*b^12*c^3*d^5 + 23*C*a
^2*b^12*c^5*d^3 + 64*C*a^3*b^11*c^2*d^6 + 71*C*a^3*b^11*c^4*d^4 + C*a^3*b^1
1*c^6*d^2 - 215*C*a^4*b^10*c^3*d^5 - 43*C*a^4*b^10*c^5*d^3 + 185*C*a^5*b^9*
c^2*d^6 + 229*C*a^5*b^9*c^4*d^4 + 41*C*a^5*b^9*c^6*d^2 - 349*C*a^6*b^8*c^3*
d^5 - 107*C*a^6*b^8*c^5*d^3 + 163*C*a^7*b^7*c^2*d^6 + 197*C*a^7*b^7*c^4*d^4
 + 27*C*a^7*b^7*c^6*d^2 - 181*C*a^8*b^6*c^3*d^5 - 53*C*a^8*b^6*c^5*d^3 + 27
*C*a^9*b^5*c^2*d^6 + 51*C*a^9*b^5*c^4*d^4 - 15*C*a^10*b^4*c^3*d^5 - 15*C*a^
11*b^3*c^2*d^6 + 7*B*a*b^13*c^7*d - B*a^13*b*c*d^7))/(a^12*d^4 + b^12*c^4 +
 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4 + a^4*b^8*d^4
 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9*c*d^3 - 16*a^
3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^5*c*d^3 - 16*a
^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^10*c^2*d^2 + 24
*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d^2 + 6*a^10*b^2*c^2
*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3)) - (A^2*a^9*b^2*d^7 - 45*A^2*a^5*b^
6*d^7 - 24*A^2*a^7*b^4*d^7 - 28*A^2*a^3*b^8*d^7 - B^2*a^5*b^6*d^7 - 3*B^2*a
^9*b^2*d^7 + 4*A^2*b^11*c^3*d^4 - A^2*b^11*c^5*d^2 - C^2*a^5*b^6*d^7 - 4*C^
2*a^7*b^4*d^7 + C^2*a^9*b^2*d^7 - B^2*b^11*c^5*d^2 - C^2*b^11*c^5*d^2 - 4*A
^2*a*b^10*d^7 - 4*A^2*b^11*c*d^6 - 26*A^2*a^2*b^9*c^3*d^4 + 10*A^2*a^2*b^9*
c^5*d^2 + 14*A^2*a^3*b^8*c^2*d^5 - 24*A^2*a^3*b^8*c^4*d^3 + 72*A^2*a^4*b^7*
c^3*d^4 - 13*A^2*a^4*b^7*c^5*d^2 - 154*A^2*a^5*b^6*c^2*d^5 + 33*A^2*a^5*b^6
*c^4*d^3 - 42*A^2*a^6*b^5*c^3*d^4 + 28*A^2*a^7*b^4*c^2*d^5 + 34*B^2*a^2*b^9
*c^3*d^4 - 14*B^2*a^2*b^9*c^5*d^2 - 46*B^2*a^3*b^8*c^2*d^5 + 36*B^2*a^3*b^8
*c^4*d^3 - 68*B^2*a^4*b^7*c^3*d^4 + 11*B^2*a^4*b^7*c^5*d^2 + 102*B^2*a^5*b^
6*c^2*d^5 - 27*B^2*a^5*b^6*c^4*d^3 + 42*B^2*a^6*b^5*c^3*d^4 - 52*B^2*a^7*b^
4*c^2*d^5 - 22*C^2*a^2*b^9*c^3*d^4 + 10*C^2*a^2*b^9*c^5*d^2 + 10*C^2*a^3*b^
8*c^2*d^5 - 24*C^2*a^3*b^8*c^4*d^3 + 92*C^2*a^4*b^7*c^3*d^4 - 13*C^2*a^4*b^
7*c^5*d^2 - 134*C^2*a^5*b^6*c^2*d^5 + 33*C^2*a^5*b^6*c^4*d^3 - 30*C^2*a^6*b
^5*c^3*d^4 + 48*C^2*a^7*b^4*c^2*d^5 + 4*C^2*a^9*b^2*c^2*d^5 - 4*A*B*a^2*b^9
*d^7 + 4*A*B*a^4*b^7*d^7 + 19*A*B*a^6*b^5*d^7 + 18*A*B*a^8*b^3*d^7 + 12*A*C
*a^3*b^8*d^7 + 22*A*C*a^5*b^6*d^7 + 12*A*C*a^7*b^4*d^7 - 6*A*C*a^9*b^2*d^7
+ 4*A*B*b^11*c^2*d^5 + B*C*a^6*b^5*d^7 - 6*B*C*a^8*b^3*d^7 - 4*A*C*b^11*c^3
*d^4 + 2*A*C*b^11*c^5*d^2 - 2*A^2*a*b^10*c^6*d + 2*B^2*a*b^10*c^6*d - 2*C^2
*a*b^10*c^6*d + 4*C^2*a^10*b*c*d^6 + 8*A^2*a*b^10*c^2*d^5 + 3*A^2*a*b^10*c^
4*d^3 + 8*A^2*a^2*b^9*c*d^6 + 2*A^2*a^3*b^8*c^6*d + 63*A^2*a^4*b^7*c*d^6 +
130*A^2*a^6*b^5*c*d^6 - 9*A^2*a^8*b^3*c*d^6 - 12*B^2*a*b^10*c^2*d^5 + 3*B^2
*a*b^10*c^4*d^3 + 4*B^2*a^2*b^9*c*d^6 - 2*B^2*a^3*b^8*c^6*d + 3*B^2*a^4*b^7
*c*d^6 - 50*B^2*a^6*b^5*c*d^6 + 39*B^2*a^8*b^3*c*d^6 + 3*C^2*a*b^10*c^4*d^3

+ 2*C^2*a^3*b^8*c^6*d + 3*C^2*a^4*b^7*c*d^6 + 54*C^2*a^6*b^5*c*d^6 - 33*C^
2*a^8*b^3*c*d^6 - A*B*a^10*b*d^7 - A*B*b^11*c^6*d + B*C*a^10*b*d^7 + B*C*b^
11*c^6*d + 16*A*B*a*b^10*c*d^6 + 4*A*C*a*b^10*c^6*d - 24*A*B*a*b^10*c^3*d^4
+ 6*A*B*a*b^10*c^5*d^2 + 6*A*B*a^2*b^9*c^6*d + 56*A*B*a^3*b^8*c*d^6 - A*B*
a^4*b^7*c^6*d + 70*A*B*a^5*b^6*c*d^6 - 140*A*B*a^7*b^4*c*d^6 + 6*A*B*a^9*b^
2*c*d^6 - 4*A*C*a*b^10*c^2*d^5 - 6*A*C*a*b^10*c^4*d^3 - 20*A*C*a^2*b^9*c*d^
6 - 4*A*C*a^3*b^8*c^6*d - 74*A*C*a^4*b^7*c*d^6 - 176*A*C*a^6*b^5*c*d^6 + 54
*A*C*a^8*b^3*c*d^6 + 12*B*C*a*b^10*c^3*d^4 - 6*B*C*a*b^10*c^5*d^2 - 6*B*C*a
^2*b^9*c^6*d - 12*B*C*a^3*b^8*c*d^6 + B*C*a^4*b^7*c^6*d - 50*B*C*a^5*b^6*c*
d^6 + 112*B*C*a^7*b^4*c*d^6 - 26*B*C*a^9*b^2*c*d^6 - 20*A*B*a^2*b^9*c^2*d^5
- 15*A*B*a^2*b^9*c^4*d^3 + 100*A*B*a^3*b^8*c^3*d^4 - 36*A*B*a^3*b^8*c^5*d^
2 - 195*A*B*a^4*b^7*c^2*d^5 + 90*A*B*a^4*b^7*c^4*d^3 - 144*A*B*a^5*b^6*c^3*
d^4 + 6*A*B*a^5*b^6*c^5*d^2 + 190*A*B*a^6*b^5*c^2*d^5 - 15*A*B*a^6*b^5*c^4*
d^3 + 20*A*B*a^7*b^4*c^3*d^4 - 15*A*B*a^8*b^3*c^2*d^5 + 48*A*C*a^2*b^9*c^3*
d^4 - 20*A*C*a^2*b^9*c^5*d^2 - 8*A*C*a^3*b^8*c^2*d^5 + 48*A*C*a^3*b^8*c^4*d
^3 - 164*A*C*a^4*b^7*c^3*d^4 + 26*A*C*a^4*b^7*c^5*d^2 + 312*A*C*a^5*b^6*c^2
*d^5 - 66*A*C*a^5*b^6*c^4*d^3 + 72*A*C*a^6*b^5*c^3*d^4 - 60*A*C*a^7*b^4*c^2
*d^5 + 16*B*C*a^2*b^9*c^2*d^5 + 15*B*C*a^2*b^9*c^4*d^3 - 120*B*C*a^3*b^8*c^
3*d^4 + 36*B*C*a^3*b^8*c^5*d^2 + 175*B*C*a^4*b^7*c^2*d^5 - 90*B*C*a^4*b^7*c
^4*d^3 + 140*B*C*a^5*b^6*c^3*d^4 - 6*B*C*a^5*b^6*c^5*d^2 - 202*B*C*a^6*b^5*
c^2*d^5 + 15*B*C*a^6*b^5*c^4*d^3 - 16*B*C*a^7*b^4*c^3*d^4 + 15*B*C*a^8*b^3*
c^2*d^5)/(a^12*d^4 + b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*
c^4 + a^8*b^4*c^4 + a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^
2*d^4 - 4*a^3*b^9*c*d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*
c^3*d - 24*a^7*b^5*c*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*
c^3*d + 6*a^2*b^10*c^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a
^8*b^4*c^2*d^2 + 6*a^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3) + (t
an(e + f*x)*(2*A^2*b^11*d^7 + 6*A^2*a^2*b^9*d^7 - 12*A^2*a^4*b^7*d^7 - 66*A
^2*a^6*b^5*d^7 + 18*A^2*a^8*b^3*d^7 - 2*B^2*a^4*b^7*d^7 + 29*B^2*a^6*b^5*d^
7 - 36*B^2*a^8*b^3*d^7 - 6*A^2*b^11*c^2*d^5 + 2*A^2*b^11*c^4*d^3 + 2*C^2*a^
4*b^7*d^7 - 32*C^2*a^6*b^5*d^7 + 30*C^2*a^8*b^3*d^7 + 2*B^2*b^11*c^2*d^5 +
2*B^2*b^11*c^4*d^3 + 4*C^2*b^11*c^4*d^3 + B^2*a^10*b*d^7 - 4*C^2*a^10*b*d^7
+ B^2*b^11*c^6*d + 38*A^2*a^2*b^9*c^2*d^5 + 4*A^2*a^2*b^9*c^4*d^3 - 16*A^2
*a^3*b^8*c^3*d^4 - 24*A^2*a^3*b^8*c^5*d^2 - 2*A^2*a^4*b^7*c^2*d^5 + 62*A^2*
a^4*b^7*c^4*d^3 - 88*A^2*a^5*b^6*c^3*d^4 + 78*A^2*a^6*b^5*c^2*d^5 - 8*B^2*a
^2*b^9*c^2*d^5 + 19*B^2*a^2*b^9*c^4*d^3 - 46*B^2*a^3*b^8*c^3*d^4 + 12*B^2*a
^3*b^8*c^5*d^2 + 83*B^2*a^4*b^7*c^2*d^5 - 28*B^2*a^4*b^7*c^4*d^3 + 30*B^2*a
^5*b^6*c^3*d^4 - 6*B^2*a^5*b^6*c^5*d^2 - 22*B^2*a^6*b^5*c^2*d^5 + 15*B^2*a^
6*b^5*c^4*d^3 - 18*B^2*a^7*b^4*c^3*d^4 + 9*B^2*a^8*b^3*c^2*d^5 + 12*C^2*a^2
*b^9*c^2*d^5 + 2*C^2*a^2*b^9*c^4*d^3 - 24*C^2*a^3*b^8*c^5*d^2 - 82*C^2*a^4*
b^7*c^2*d^5 + 52*C^2*a^4*b^7*c^4*d^3 - 56*C^2*a^5*b^6*c^3*d^4 + 22*C^2*a^6*
b^5*c^2*d^5 - 6*C^2*a^6*b^5*c^4*d^3 + 16*C^2*a^7*b^4*c^3*d^4 - 6*C^2*a^8*b^
3*c^2*d^5 - 6*A*B*a^3*b^8*d^7 - 18*A*B*a^5*b^6*d^7 + 114*A*B*a^7*b^4*d^7 -
10*A*B*a^9*b^2*d^7 + 14*A*C*a^4*b^7*d^7 + 94*A*C*a^6*b^5*d^7 - 54*A*C*a^8*b
^3*d^7 + 2*A*B*b^11*c^3*d^4 + 24*B*C*a^5*b^6*d^7 - 84*B*C*a^7*b^4*d^7 + 28*

B*C*a^9*b^2*d^7 + 4*A*C*b^11*c^2*d^5 - 6*A*C*b^11*c^4*d^3 - 4*B*C*b^11*c^3*
d^4 - 8*A^2*a*b^10*c*d^6 - 8*A^2*a*b^10*c^3*d^4 + 4*A^2*a^2*b^9*c^6*d - 40*
A^2*a^3*b^8*c*d^6 + 72*A^2*a^5*b^6*c*d^6 - 48*A^2*a^7*b^4*c*d^6 - 14*B^2*a*
b^10*c^3*d^4 - 6*B^2*a*b^10*c^5*d^2 - 2*B^2*a^2*b^9*c^6*d + 14*B^2*a^3*b^8*
c*d^6 + B^2*a^4*b^7*c^6*d - 100*B^2*a^5*b^6*c*d^6 + 38*B^2*a^7*b^4*c*d^6 -
8*C^2*a*b^10*c^3*d^4 + 4*C^2*a^2*b^9*c^6*d - 8*C^2*a^3*b^8*c*d^6 + 104*C^2*
a^5*b^6*c*d^6 - 48*C^2*a^7*b^4*c*d^6 - 8*C^2*a^9*b^2*c*d^6 + 2*C^2*a^10*b*c
^2*d^5 + 2*A*C*a^10*b*d^7 - 4*A*B*b^11*c*d^6 + 4*A*B*a*b^10*c^6*d - 4*B*C*a
*b^10*c^6*d - 2*B*C*a^10*b*c*d^6 + 30*A*B*a*b^10*c^2*d^5 - 10*A*B*a^2*b^9*c
*d^6 - 4*A*B*a^3*b^8*c^6*d + 114*A*B*a^4*b^7*c*d^6 - 166*A*B*a^6*b^5*c*d^6
 + 18*A*B*a^8*b^3*c*d^6 + 16*A*C*a*b^10*c^3*d^4 - 8*A*C*a^2*b^9*c^6*d + 16*A
*C*a^3*b^8*c*d^6 - 224*A*C*a^5*b^6*c*d^6 + 64*A*C*a^7*b^4*c*d^6 + 6*B*C*a*b
^10*c^4*d^3 + 4*B*C*a^3*b^8*c^6*d - 106*B*C*a^4*b^7*c*d^6 + 194*B*C*a^6*b^5
*c*d^6 - 6*B*C*a^8*b^3*c*d^6 - 2*A*B*a^2*b^9*c^3*d^4 - 24*A*B*a^2*b^9*c^5*d
^2 - 54*A*B*a^3*b^8*c^2*d^5 + 60*A*B*a^3*b^8*c^4*d^3 - 90*A*B*a^4*b^7*c^3*d
^4 + 24*A*B*a^4*b^7*c^5*d^2 + 118*A*B*a^5*b^6*c^2*d^5 - 60*A*B*a^5*b^6*c^4*
d^3 + 74*A*B*a^6*b^5*c^3*d^4 - 46*A*B*a^7*b^4*c^2*d^5 - 56*A*C*a^2*b^9*c^2*
d^5 - 6*A*C*a^2*b^9*c^4*d^3 + 16*A*C*a^3*b^8*c^3*d^4 + 48*A*C*a^3*b^8*c^5*d
^2 + 80*A*C*a^4*b^7*c^2*d^5 - 114*A*C*a^4*b^7*c^4*d^3 + 144*A*C*a^5*b^6*c^3
*d^4 - 96*A*C*a^6*b^5*c^2*d^5 + 6*A*C*a^6*b^5*c^4*d^3 - 16*A*C*a^7*b^4*c^3*
d^4 + 12*A*C*a^8*b^3*c^2*d^5 - 14*B*C*a^2*b^9*c^3*d^4 + 24*B*C*a^2*b^9*c^5*
d^2 + 106*B*C*a^3*b^8*c^2*d^5 - 50*B*C*a^3*b^8*c^4*d^3 + 70*B*C*a^4*b^7*c^3
*d^4 - 24*B*C*a^4*b^7*c^5*d^2 - 110*B*C*a^5*b^6*c^2*d^5 + 62*B*C*a^5*b^6*c^
4*d^3 - 74*B*C*a^6*b^5*c^3*d^4 + 26*B*C*a^7*b^4*c^2*d^5 - 2*B*C*a^7*b^4*c^4
*d^3 + 6*B*C*a^8*b^3*c^3*d^4 - 6*B*C*a^9*b^2*c^2*d^5))/(a^12*d^4 + b^12*c^4
 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^4*c^4 + a^4*b^8*d
^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^3*b^9*c*d^3 - 16*
a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a^7*b^5*c*d^3 - 16
*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^2*b^10*c^2*d^2 +
24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d^2 + 6*a^10*b^2*c
^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3)) - (tan(e + f*x)*(B^3*a^4*b^4*d^6
 - A^3*a^3*b^5*d^6 - 3*B^3*a^6*b^2*d^6 - 3*C^3*a^5*b^3*d^6 - B^3*b^8*c^2*d^
4 - A^2*B*b^8*d^6 - A^3*a*b^7*d^6 + A^3*b^8*c*d^5 + C^3*a^7*b*d^6 + 2*B^3*a
^2*b^6*c^2*d^4 - B^3*a^4*b^4*c^2*d^4 + 4*C^3*a^2*b^6*c^3*d^3 - 12*C^3*a^3*b
^5*c^2*d^4 - A*C^2*a^7*b*d^6 + A^2*C*a*b^7*d^6 + 2*A*B^2*b^8*c*d^5 + B^2*C*
a^7*b*d^6 - A^2*C*b^8*c*d^5 + A*B^2*a^3*b^5*d^6 + 9*A*B^2*a^5*b^3*d^6 - 3*A
^2*B*a^2*b^6*d^6 - 6*A^2*B*a^4*b^4*d^6 + 2*A*C^2*a^3*b^5*d^6 + 9*A*C^2*a^5*
b^3*d^6 - A^2*C*a^3*b^5*d^6 - 6*A^2*C*a^5*b^3*d^6 + B*C^2*a^4*b^4*d^6 - 3*B
*C^2*a^6*b^2*d^6 - 3*B^2*C*a^5*b^3*d^6 + B^2*C*b^8*c^3*d^3 + A^3*a^2*b^6*c*
d^5 - 5*B^3*a^3*b^5*c*d^5 + 3*B^3*a^5*b^3*c*d^5 + 11*C^3*a^4*b^4*c*d^5 - C^
3*a^6*b^2*c*d^5 + 4*A*B^2*a^3*b^5*c^2*d^4 - 4*A^2*B*a^2*b^6*c^2*d^4 - 8*A*C
^2*a^2*b^6*c^3*d^3 + 24*A*C^2*a^3*b^5*c^2*d^4 + 4*A^2*C*a^2*b^6*c^3*d^3 - 1
2*A^2*C*a^3*b^5*c^2*d^4 + 8*B*C^2*a^2*b^6*c^2*d^4 + 4*B*C^2*a^3*b^5*c^3*d^3
 - 12*B*C^2*a^4*b^4*c^2*d^4 - 2*B^2*C*a^2*b^6*c^3*d^3 + 2*B^2*C*a^3*b^5*c^2
*d^4 + B^2*C*a^4*b^4*c^3*d^3 - 3*B^2*C*a^5*b^3*c^2*d^4 + 2*A*B*C*a^4*b^4*d^

6 + 2*A*B*C*a^6*b^2*d^6 + A^2*B*a*b^7*c*d^5 - B*C^2*a^7*b*c*d^5 - 4*A*B^2*a
*b^7*c^2*d^4 + 7*A*B^2*a^2*b^6*c*d^5 - 11*A*B^2*a^4*b^4*c*d^5 + 9*A^2*B*a^3
*b^5*c*d^5 - 2*A*C^2*a^2*b^6*c*d^5 - 25*A*C^2*a^4*b^4*c*d^5 + A*C^2*a^6*b^2
*c*d^5 + A^2*C*a^2*b^6*c*d^5 + 14*A^2*C*a^4*b^4*c*d^5 - 4*B*C^2*a*b^7*c^3*d
^3 - 6*B*C^2*a^3*b^5*c*d^5 + 9*B*C^2*a^5*b^3*c*d^5 + B^2*C*a*b^7*c^2*d^4 +
7*B^2*C*a^4*b^4*c*d^5 + 3*B^2*C*a^6*b^2*c*d^5 - 4*A*B*C*a^2*b^6*c^2*d^4 - 4
*A*B*C*a^3*b^5*c^3*d^3 + 12*A*B*C*a^4*b^4*c^2*d^4 - 2*A*B*C*a*b^7*c*d^5 + 4
*A*B*C*a*b^7*c^3*d^3 - 6*A*B*C*a^3*b^5*c*d^5 - 12*A*B*C*a^5*b^3*c*d^5))/(a^
12*d^4 + b^12*c^4 + 4*a^2*b^10*c^4 + 6*a^4*b^8*c^4 + 4*a^6*b^6*c^4 + a^8*b^
4*c^4 + a^4*b^8*d^4 + 4*a^6*b^6*d^4 + 6*a^8*b^4*d^4 + 4*a^10*b^2*d^4 - 4*a^
3*b^9*c*d^3 - 16*a^3*b^9*c^3*d - 16*a^5*b^7*c*d^3 - 24*a^5*b^7*c^3*d - 24*a
^7*b^5*c*d^3 - 16*a^7*b^5*c^3*d - 16*a^9*b^3*c*d^3 - 4*a^9*b^3*c^3*d + 6*a^
2*b^10*c^2*d^2 + 24*a^4*b^8*c^2*d^2 + 36*a^6*b^6*c^2*d^2 + 24*a^8*b^4*c^2*d
^2 + 6*a^10*b^2*c^2*d^2 - 4*a*b^11*c^3*d - 4*a^11*b*c*d^3))*root(480*a^11*b
^7*c*d^9*f^4 + 480*a^7*b^11*c^9*d*f^4 + 360*a^13*b^5*c*d^9*f^4 + 360*a^9*b^
9*c^9*d*f^4 + 360*a^9*b^9*c*d^9*f^4 + 360*a^5*b^13*c^9*d*f^4 + 144*a^15*b^3
*c*d^9*f^4 + 144*a^11*b^7*c^9*d*f^4 + 144*a^7*b^11*c*d^9*f^4 + 144*a^3*b^15
*c^9*d*f^4 + 48*a^17*b*c^3*d^7*f^4 + 48*a*b^17*c^7*d^3*f^4 + 24*a^17*b*c^5*
d^5*f^4 + 24*a^13*b^5*c^9*d*f^4 + 24*a^5*b^13*c*d^9*f^4 + 24*a*b^17*c^5*d^5
*f^4 + 24*a^17*b*c*d^9*f^4 + 24*a*b^17*c^9*d*f^4 + 3920*a^9*b^9*c^5*d^5*f^4
 - 3360*a^10*b^8*c^4*d^6*f^4 - 3360*a^8*b^10*c^6*d^4*f^4 + 3024*a^11*b^7*c^
5*d^5*f^4 - 3024*a^10*b^8*c^6*d^4*f^4 - 3024*a^8*b^10*c^4*d^6*f^4 + 3024*a^
7*b^11*c^5*d^5*f^4 + 2320*a^9*b^9*c^7*d^3*f^4 + 2320*a^9*b^9*c^3*d^7*f^4 -
2240*a^12*b^6*c^4*d^6*f^4 - 2240*a^6*b^12*c^6*d^4*f^4 + 2160*a^11*b^7*c^3*d
^7*f^4 + 2160*a^7*b^11*c^7*d^3*f^4 - 1624*a^12*b^6*c^6*d^4*f^4 - 1624*a^6*b
^12*c^4*d^6*f^4 + 1488*a^11*b^7*c^7*d^3*f^4 + 1488*a^7*b^11*c^3*d^7*f^4 + 1
344*a^13*b^5*c^5*d^5*f^4 + 1344*a^5*b^13*c^5*d^5*f^4 - 1320*a^10*b^8*c^2*d^
8*f^4 - 1320*a^8*b^10*c^8*d^2*f^4 + 1200*a^13*b^5*c^3*d^7*f^4 + 1200*a^5*b^
13*c^7*d^3*f^4 - 1060*a^12*b^6*c^2*d^8*f^4 - 1060*a^6*b^12*c^8*d^2*f^4 - 94
8*a^10*b^8*c^8*d^2*f^4 - 948*a^8*b^10*c^2*d^8*f^4 - 840*a^14*b^4*c^4*d^6*f^
4 - 840*a^4*b^14*c^6*d^4*f^4 + 528*a^13*b^5*c^7*d^3*f^4 + 528*a^5*b^13*c^3*
d^7*f^4 - 480*a^14*b^4*c^6*d^4*f^4 - 480*a^14*b^4*c^2*d^8*f^4 - 480*a^4*b^1
4*c^8*d^2*f^4 - 480*a^4*b^14*c^4*d^6*f^4 + 368*a^15*b^3*c^3*d^7*f^4 - 368*a
^12*b^6*c^8*d^2*f^4 - 368*a^6*b^12*c^2*d^8*f^4 + 368*a^3*b^15*c^7*d^3*f^4 +
 304*a^15*b^3*c^5*d^5*f^4 + 304*a^3*b^15*c^5*d^5*f^4 - 144*a^16*b^2*c^4*d^6
*f^4 - 144*a^2*b^16*c^6*d^4*f^4 - 108*a^16*b^2*c^2*d^8*f^4 - 108*a^2*b^16*c
^8*d^2*f^4 + 80*a^15*b^3*c^7*d^3*f^4 + 80*a^3*b^15*c^3*d^7*f^4 - 60*a^16*b^
2*c^6*d^4*f^4 - 60*a^14*b^4*c^8*d^2*f^4 - 60*a^4*b^14*c^2*d^8*f^4 - 60*a^2*
b^16*c^4*d^6*f^4 - 8*b^18*c^8*d^2*f^4 - 4*b^18*c^6*d^4*f^4 - 8*a^18*c^2*d^8
*f^4 - 4*a^18*c^4*d^6*f^4 - 80*a^12*b^6*d^10*f^4 - 60*a^14*b^4*d^10*f^4 - 6
0*a^10*b^8*d^10*f^4 - 24*a^16*b^2*d^10*f^4 - 24*a^8*b^10*d^10*f^4 - 4*a^6*b
^12*d^10*f^4 - 80*a^6*b^12*c^10*f^4 - 60*a^8*b^10*c^10*f^4 - 60*a^4*b^14*c^
10*f^4 - 24*a^10*b^8*c^10*f^4 - 24*a^2*b^16*c^10*f^4 - 4*a^12*b^6*c^10*f^4
 - 4*b^18*c^10*f^4 - 4*a^18*d^10*f^4 - 12*A*C*a^11*b*c*d^7*f^2 - 12*A*C*a*b^
11*c^7*d*f^2 - 912*B*C*a^5*b^7*c^4*d^4*f^2 - 792*B*C*a^8*b^4*c^3*d^5*f^2 +

792*B*C*a^4*b^8*c^5*d^3*f^2 + 720*B*C*a^7*b^5*c^4*d^4*f^2 - 480*B*C*a^5*b^7
*c^6*d^2*f^2 - 408*B*C*a^5*b^7*c^2*d^6*f^2 + 384*B*C*a^7*b^5*c^2*d^6*f^2 -
336*B*C*a^8*b^4*c^5*d^3*f^2 + 324*B*C*a^4*b^8*c^3*d^5*f^2 + 312*B*C*a^7*b^5
*c^6*d^2*f^2 - 248*B*C*a^3*b^9*c^6*d^2*f^2 + 216*B*C*a^9*b^3*c^2*d^6*f^2 -
196*B*C*a^3*b^9*c^4*d^4*f^2 + 132*B*C*a^9*b^3*c^4*d^4*f^2 + 80*B*C*a^6*b^6*
c^3*d^5*f^2 - 64*B*C*a^6*b^6*c^5*d^3*f^2 - 36*B*C*a^2*b^10*c^3*d^5*f^2 - 28
*B*C*a^3*b^9*c^2*d^6*f^2 + 12*B*C*a^10*b^2*c^5*d^3*f^2 - 12*B*C*a^10*b^2*c^
3*d^5*f^2 - 12*B*C*a^2*b^10*c^5*d^3*f^2 - 4*B*C*a^9*b^3*c^6*d^2*f^2 - 1468*
A*C*a^6*b^6*c^4*d^4*f^2 + 996*A*C*a^7*b^5*c^3*d^5*f^2 + 900*A*C*a^5*b^7*c^5
*d^3*f^2 - 676*A*C*a^6*b^6*c^6*d^2*f^2 - 660*A*C*a^6*b^6*c^2*d^6*f^2 + 636*
A*C*a^5*b^7*c^3*d^5*f^2 + 540*A*C*a^7*b^5*c^5*d^3*f^2 - 236*A*C*a^3*b^9*c^5
*d^3*f^2 - 204*A*C*a^9*b^3*c^3*d^5*f^2 + 156*A*C*a^10*b^2*c^2*d^6*f^2 + 132
*A*C*a^2*b^10*c^6*d^2*f^2 - 72*A*C*a^9*b^3*c^5*d^3*f^2 - 72*A*C*a^4*b^8*c^6
*d^2*f^2 + 66*A*C*a^4*b^8*c^2*d^6*f^2 + 54*A*C*a^10*b^2*c^4*d^4*f^2 + 54*A*
C*a^2*b^10*c^4*d^4*f^2 - 48*A*C*a^8*b^4*c^2*d^6*f^2 - 48*A*C*a^4*b^8*c^4*d^
4*f^2 + 42*A*C*a^8*b^4*c^6*d^2*f^2 - 40*A*C*a^3*b^9*c^3*d^5*f^2 - 36*A*C*a^
8*b^4*c^4*d^4*f^2 + 24*A*C*a^2*b^10*c^2*d^6*f^2 + 960*A*B*a^5*b^7*c^4*d^4*f
^2 - 864*A*B*a^4*b^8*c^5*d^3*f^2 + 756*A*B*a^8*b^4*c^3*d^5*f^2 - 744*A*B*a^
7*b^5*c^4*d^4*f^2 - 528*A*B*a^4*b^8*c^3*d^5*f^2 + 504*A*B*a^5*b^7*c^6*d^2*f
^2 - 432*A*B*a^7*b^5*c^2*d^6*f^2 + 432*A*B*a^5*b^7*c^2*d^6*f^2 + 348*A*B*a^
8*b^4*c^5*d^3*f^2 - 312*A*B*a^7*b^5*c^6*d^2*f^2 - 284*A*B*a^9*b^3*c^2*d^6*f
^2 + 280*A*B*a^3*b^9*c^6*d^2*f^2 + 264*A*B*a^3*b^9*c^4*d^4*f^2 - 240*A*B*a^
6*b^6*c^3*d^5*f^2 - 172*A*B*a^9*b^3*c^4*d^4*f^2 + 68*A*B*a^3*b^9*c^2*d^6*f^
2 - 60*A*B*a^2*b^10*c^3*d^5*f^2 + 24*A*B*a^6*b^6*c^5*d^3*f^2 - 24*A*B*a^2*b
^10*c^5*d^3*f^2 + 12*A*B*a^10*b^2*c^3*d^5*f^2 + 360*B*C*a^4*b^8*c^7*d*f^2 -
336*B*C*a^8*b^4*c*d^7*f^2 + 168*B*C*a^6*b^6*c*d^7*f^2 - 136*B*C*a^6*b^6*c^
7*d*f^2 - 36*B*C*a^11*b*c^2*d^6*f^2 + 36*B*C*a*b^11*c^6*d^2*f^2 + 24*B*C*a^
10*b^2*c*d^7*f^2 - 24*B*C*a^2*b^10*c^7*d*f^2 - 12*B*C*a^11*b*c^4*d^4*f^2 +
12*B*C*a^4*b^8*c*d^7*f^2 + 12*B*C*a*b^11*c^4*d^4*f^2 + 444*A*C*a^7*b^5*c*d^
7*f^2 + 348*A*C*a^5*b^7*c^7*d*f^2 - 164*A*C*a^3*b^9*c^7*d*f^2 - 132*A*C*a^9
*b^3*c*d^7*f^2 + 84*A*C*a^5*b^7*c*d^7*f^2 + 32*A*C*a^3*b^9*c*d^7*f^2 - 12*A
*C*a^11*b*c^3*d^5*f^2 - 12*A*C*a^7*b^5*c^7*d*f^2 - 12*A*C*a*b^11*c^5*d^3*f^
2 - 360*A*B*a^4*b^8*c^7*d*f^2 + 288*A*B*a^8*b^4*c*d^7*f^2 - 288*A*B*a^6*b^6
*c*d^7*f^2 - 144*A*B*a^4*b^8*c*d^7*f^2 + 136*A*B*a^6*b^6*c^7*d*f^2 - 60*A*B
*a^2*b^10*c*d^7*f^2 - 36*A*B*a^10*b^2*c*d^7*f^2 + 24*A*B*a^2*b^10*c^7*d*f^2
 - 24*A*B*a*b^11*c^6*d^2*f^2 + 12*A*B*a^11*b*c^2*d^6*f^2 + 12*A*B*a*b^11*c^
4*d^4*f^2 + 12*A*B*a*b^11*c^2*d^6*f^2 - 8*B*C*b^12*c^5*d^3*f^2 - 8*B*C*b^12
*c^3*d^5*f^2 + 8*A*C*b^12*c^2*d^6*f^2 - 4*B*C*a^12*c^3*d^5*f^2 + 4*A*C*b^12
*c^4*d^4*f^2 - 2*A*C*b^12*c^6*d^2*f^2 + 80*B*C*a^9*b^3*d^8*f^2 - 24*B*C*a^7
*b^5*d^8*f^2 + 6*A*C*a^12*c^2*d^6*f^2 + 4*A*B*b^12*c^5*d^3*f^2 - 4*A*B*b^12
*c^3*d^5*f^2 - 90*A*C*a^8*b^4*d^8*f^2 - 80*B*C*a^3*b^9*c^8*f^2 + 54*A*C*a^1
0*b^2*d^8*f^2 - 30*A*C*a^6*b^6*d^8*f^2 + 24*B*C*a^5*b^7*c^8*f^2 - 12*A*C*a^
4*b^8*d^8*f^2 - 112*A*B*a^9*b^3*d^8*f^2 - 66*A*C*a^4*b^8*c^8*f^2 + 54*A*C*a
^2*b^10*c^8*f^2 + 4*A*B*a^3*b^9*d^8*f^2 + 2*A*C*a^6*b^6*c^8*f^2 + 80*A*B*a^
3*b^9*c^8*f^2 - 24*A*B*a^5*b^7*c^8*f^2 + 726*C^2*a^6*b^6*c^4*d^4*f^2 - 402*

C^2*a^7*b^5*c^3*d^5*f^2 - 402*C^2*a^5*b^7*c^5*d^3*f^2 + 322*C^2*a^6*b^6*c^6
*d^2*f^2 + 322*C^2*a^6*b^6*c^2*d^6*f^2 - 222*C^2*a^7*b^5*c^5*d^3*f^2 - 222*
C^2*a^5*b^7*c^3*d^5*f^2 + 134*C^2*a^9*b^3*c^3*d^5*f^2 + 134*C^2*a^3*b^9*c^5
*d^3*f^2 - 66*C^2*a^10*b^2*c^2*d^6*f^2 - 66*C^2*a^2*b^10*c^6*d^2*f^2 + 52*C
^2*a^9*b^3*c^5*d^3*f^2 + 52*C^2*a^3*b^9*c^3*d^5*f^2 - 27*C^2*a^8*b^4*c^6*d^
2*f^2 - 27*C^2*a^4*b^8*c^2*d^6*f^2 + 24*C^2*a^8*b^4*c^4*d^4*f^2 + 24*C^2*a^
8*b^4*c^2*d^6*f^2 + 24*C^2*a^4*b^8*c^6*d^2*f^2 + 24*C^2*a^4*b^8*c^4*d^4*f^2
 - 15*C^2*a^10*b^2*c^4*d^4*f^2 - 15*C^2*a^2*b^10*c^4*d^4*f^2 - 570*B^2*a^6*
b^6*c^4*d^4*f^2 + 366*B^2*a^7*b^5*c^3*d^5*f^2 + 318*B^2*a^5*b^7*c^5*d^3*f^2
 - 262*B^2*a^6*b^6*c^6*d^2*f^2 - 222*B^2*a^6*b^6*c^2*d^6*f^2 - 210*B^2*a^3*
b^9*c^5*d^3*f^2 + 186*B^2*a^7*b^5*c^5*d^3*f^2 + 162*B^2*a^5*b^7*c^3*d^5*f^2
 - 142*B^2*a^9*b^3*c^3*d^5*f^2 + 132*B^2*a^4*b^8*c^4*d^4*f^2 + 117*B^2*a^4*
b^8*c^2*d^6*f^2 + 102*B^2*a^2*b^10*c^6*d^2*f^2 - 96*B^2*a^3*b^9*c^3*d^5*f^2
 + 90*B^2*a^10*b^2*c^2*d^6*f^2 + 81*B^2*a^2*b^10*c^4*d^4*f^2 - 56*B^2*a^9*b
^3*c^5*d^3*f^2 + 48*B^2*a^8*b^4*c^4*d^4*f^2 + 48*B^2*a^4*b^8*c^6*d^2*f^2 +
45*B^2*a^8*b^4*c^6*d^2*f^2 + 36*B^2*a^8*b^4*c^2*d^6*f^2 + 36*B^2*a^2*b^10*c
^2*d^6*f^2 + 33*B^2*a^10*b^2*c^4*d^4*f^2 + 822*A^2*a^6*b^6*c^4*d^4*f^2 - 59
4*A^2*a^7*b^5*c^3*d^5*f^2 + 498*A^2*a^6*b^6*c^2*d^6*f^2 - 498*A^2*a^5*b^7*c
^5*d^3*f^2 - 414*A^2*a^5*b^7*c^3*d^5*f^2 + 354*A^2*a^6*b^6*c^6*d^2*f^2 - 31
8*A^2*a^7*b^5*c^5*d^3*f^2 + 144*A^2*a^8*b^4*c^2*d^6*f^2 + 102*A^2*a^3*b^9*c
^5*d^3*f^2 + 84*A^2*a^4*b^8*c^4*d^4*f^2 + 81*A^2*a^4*b^8*c^2*d^6*f^2 + 72*A
^2*a^8*b^4*c^4*d^4*f^2 + 70*A^2*a^9*b^3*c^3*d^5*f^2 - 66*A^2*a^2*b^10*c^6*d
^2*f^2 + 48*A^2*a^4*b^8*c^6*d^2*f^2 - 42*A^2*a^10*b^2*c^2*d^6*f^2 + 24*A^2*
a^2*b^10*c^2*d^6*f^2 + 20*A^2*a^9*b^3*c^5*d^3*f^2 - 15*A^2*a^10*b^2*c^4*d^4
*f^2 - 15*A^2*a^8*b^4*c^6*d^2*f^2 - 15*A^2*a^2*b^10*c^4*d^4*f^2 - 12*A^2*a^
3*b^9*c^3*d^5*f^2 - 8*B*C*b^12*c^7*d*f^2 + 4*B*C*a^12*c*d^7*f^2 - 24*B*C*a^
11*b*d^8*f^2 + 8*A*B*b^12*c^7*d*f^2 - 8*A*B*b^12*c*d^7*f^2 + 24*B*C*a*b^11*
c^8*f^2 - 8*A*B*a^12*c*d^7*f^2 + 12*A*B*a^11*b*d^8*f^2 - 24*A*B*a*b^11*c^8*
f^2 - 174*C^2*a^7*b^5*c*d^7*f^2 - 174*C^2*a^5*b^7*c^7*d*f^2 + 82*C^2*a^9*b^
3*c*d^7*f^2 + 82*C^2*a^3*b^9*c^7*d*f^2 + 6*C^2*a^11*b*c^3*d^5*f^2 + 6*C^2*a
^7*b^5*c^7*d*f^2 + 6*C^2*a^5*b^7*c*d^7*f^2 + 6*C^2*a*b^11*c^5*d^3*f^2 + 162
*B^2*a^7*b^5*c*d^7*f^2 + 138*B^2*a^5*b^7*c^7*d*f^2 - 118*B^2*a^3*b^9*c^7*d*
f^2 - 86*B^2*a^9*b^3*c*d^7*f^2 - 30*B^2*a*b^11*c^5*d^3*f^2 - 18*B^2*a^7*b^5
*c^7*d*f^2 - 18*B^2*a^5*b^7*c*d^7*f^2 - 12*B^2*a*b^11*c^3*d^5*f^2 - 6*B^2*a
^11*b*c^3*d^5*f^2 - 4*B^2*a^3*b^9*c*d^7*f^2 - 270*A^2*a^7*b^5*c*d^7*f^2 - 1
74*A^2*a^5*b^7*c^7*d*f^2 - 90*A^2*a^5*b^7*c*d^7*f^2 + 82*A^2*a^3*b^9*c^7*d*
f^2 + 50*A^2*a^9*b^3*c*d^7*f^2 - 32*A^2*a^3*b^9*c*d^7*f^2 + 6*A^2*a^11*b*c^
3*d^5*f^2 + 6*A^2*a^7*b^5*c^7*d*f^2 + 6*A^2*a*b^11*c^5*d^3*f^2 + 6*C^2*a^11
*b*c*d^7*f^2 + 6*C^2*a*b^11*c^7*d*f^2 - 18*B^2*a*b^11*c^7*d*f^2 - 6*B^2*a^1
1*b*c*d^7*f^2 + 6*A^2*a^11*b*c*d^7*f^2 + 6*A^2*a*b^11*c^7*d*f^2 - 6*A*C*b^1
2*c^8*f^2 - 2*A*C*a^12*d^8*f^2 + 4*C^2*b^12*c^4*d^4*f^2 + 3*C^2*b^12*c^6*d^
2*f^2 + 4*C^2*a^12*c^4*d^4*f^2 + 4*B^2*b^12*c^4*d^4*f^2 + 4*B^2*b^12*c^2*d^
6*f^2 + 3*C^2*a^12*c^2*d^6*f^2 + 3*B^2*b^12*c^6*d^2*f^2 + 33*C^2*a^8*b^4*d^
8*f^2 - 27*C^2*a^10*b^2*d^8*f^2 - 4*A^2*b^12*c^4*d^4*f^2 + 3*B^2*a^12*c^2*d
^6*f^2 - C^2*a^6*b^6*d^8*f^2 - A^2*b^12*c^6*d^2*f^2 + 33*C^2*a^4*b^8*c^8*f^

2 + 33*B^2*a^10*b^2*d^8*f^2 - 27*C^2*a^2*b^10*c^8*f^2 - 27*B^2*a^8*b^4*d^8*
f^2 + 3*B^2*a^6*b^6*d^8*f^2 - C^2*a^6*b^6*c^8*f^2 - A^2*a^12*c^2*d^6*f^2 +
117*A^2*a^8*b^4*d^8*f^2 + 111*A^2*a^6*b^6*d^8*f^2 + 72*A^2*a^4*b^8*d^8*f^2
+ 33*B^2*a^2*b^10*c^8*f^2 - 27*B^2*a^4*b^8*c^8*f^2 + 24*A^2*a^2*b^10*d^8*f^
2 + 3*B^2*a^6*b^6*c^8*f^2 - 3*A^2*a^10*b^2*d^8*f^2 + 33*A^2*a^4*b^8*c^8*f^2
 - 27*A^2*a^2*b^10*c^8*f^2 - A^2*a^6*b^6*c^8*f^2 + 3*C^2*b^12*c^8*f^2 + 3*C
^2*a^12*d^8*f^2 + 4*A^2*b^12*d^8*f^2 - B^2*b^12*c^8*f^2 - B^2*a^12*d^8*f^2
+ 3*A^2*b^12*c^8*f^2 + 3*A^2*a^12*d^8*f^2 - 24*A*B*C*a*b^8*c*d^6*f + 342*A*
B*C*a^4*b^5*c^2*d^5*f - 186*A*B*C*a^5*b^4*c^3*d^4*f - 66*A*B*C*a^2*b^7*c^4*
d^3*f + 48*A*B*C*a^2*b^7*c^2*d^5*f + 42*A*B*C*a^6*b^3*c^2*d^5*f + 26*A*B*C*
a^3*b^6*c^5*d^2*f + 24*A*B*C*a^6*b^3*c^4*d^3*f - 18*A*B*C*a^7*b^2*c^3*d^4*f
 - 18*A*B*C*a^4*b^5*c^4*d^3*f - 8*A*B*C*a^3*b^6*c^3*d^4*f + 6*A*B*C*a^5*b^4
*c^5*d^2*f - 128*A*B*C*a^3*b^6*c*d^6*f + 126*A*B*C*a^7*b^2*c*d^6*f + 72*A*B
*C*a*b^8*c^3*d^4*f - 36*A*B*C*a^8*b*c^2*d^5*f - 36*A*B*C*a*b^8*c^5*d^2*f +
30*A*B*C*a^2*b^7*c^6*d*f - 12*A*B*C*a^5*b^4*c*d^6*f - 12*A*B*C*a^4*b^5*c^6*
d*f - 21*B^2*C*a^8*b*c*d^6*f - 3*B^2*C*a*b^8*c^6*d*f + 21*A^2*C*a^8*b*c*d^6
*f - 21*A*C^2*a^8*b*c*d^6*f - 9*A^2*C*a*b^8*c^6*d*f + 9*A*C^2*a*b^8*c^6*d*f
 + 36*A^2*B*a*b^8*c*d^6*f + 21*A*B^2*a^8*b*c*d^6*f + 3*A*B^2*a*b^8*c^6*d*f
+ 16*A*B*C*b^9*c^4*d^3*f - 16*A*B*C*b^9*c^2*d^5*f - 78*A*B*C*a^6*b^3*d^7*f
+ 24*A*B*C*a^4*b^5*d^7*f + 2*A*B*C*a^3*b^6*c^7*f - 237*B^2*C*a^4*b^5*c^3*d^
4*f + 165*B*C^2*a^5*b^4*c^3*d^4*f + 92*B^2*C*a^3*b^6*c^2*d^5*f - 81*B^2*C*a
^7*b^2*c^2*d^5*f + 77*B^2*C*a^3*b^6*c^4*d^3*f - 75*B*C^2*a^4*b^5*c^2*d^5*f
+ 69*B^2*C*a^5*b^4*c^4*d^3*f + 69*B*C^2*a^4*b^5*c^4*d^3*f - 68*B*C^2*a^3*b^
6*c^3*d^4*f - 63*B^2*C*a^4*b^5*c^5*d^2*f - 61*B*C^2*a^6*b^3*c^2*d^5*f + 57*
B*C^2*a^2*b^7*c^4*d^3*f - 53*B*C^2*a^3*b^6*c^5*d^2*f - 44*B*C^2*a^6*b^3*c^4
*d^3*f - 36*B^2*C*a^2*b^7*c^3*d^4*f + 35*B^2*C*a^6*b^3*c^3*d^4*f - 33*B^2*C
*a^5*b^4*c^2*d^5*f + 33*B^2*C*a^2*b^7*c^5*d^2*f + 33*B*C^2*a^7*b^2*c^3*d^4*
f - 12*B^2*C*a^7*b^2*c^4*d^3*f + 9*B*C^2*a^5*b^4*c^5*d^2*f + 4*B^2*C*a^6*b^
3*c^5*d^2*f + 225*A^2*C*a^5*b^4*c^2*d^5*f - 105*A*C^2*a^5*b^4*c^2*d^5*f - 9
9*A^2*C*a^4*b^5*c^3*d^4*f - 81*A^2*C*a^4*b^5*c^5*d^2*f + 67*A^2*C*a^3*b^6*c
^4*d^3*f - 59*A*C^2*a^3*b^6*c^4*d^3*f - 57*A*C^2*a^7*b^2*c^2*d^5*f + 57*A*C
^2*a^2*b^7*c^5*d^2*f + 51*A^2*C*a^5*b^4*c^4*d^3*f + 48*A^2*C*a^2*b^7*c^3*d^
4*f + 45*A*C^2*a^4*b^5*c^5*d^2*f - 35*A^2*C*a^6*b^3*c^3*d^4*f + 33*A^2*C*a^
7*b^2*c^2*d^5*f - 33*A^2*C*a^2*b^7*c^5*d^2*f + 33*A*C^2*a^5*b^4*c^4*d^3*f +
 27*A*C^2*a^6*b^3*c^3*d^4*f + 24*A*C^2*a^3*b^6*c^2*d^5*f - 24*A*C^2*a^2*b^7
*c^3*d^4*f - 21*A*C^2*a^4*b^5*c^3*d^4*f - 16*A^2*C*a^3*b^6*c^2*d^5*f - 243*
A^2*B*a^4*b^5*c^2*d^5*f - 156*A*B^2*a^3*b^6*c^2*d^5*f + 141*A*B^2*a^4*b^5*c
^3*d^4*f + 108*A^2*B*a^3*b^6*c^3*d^4*f - 105*A*B^2*a^3*b^6*c^4*d^3*f + 84*A
*B^2*a^2*b^7*c^3*d^4*f + 81*A*B^2*a^5*b^4*c^2*d^5*f + 51*A^2*B*a^6*b^3*c^2*
d^5*f - 51*A^2*B*a^4*b^5*c^4*d^3*f - 48*A^2*B*a^2*b^7*c^2*d^5*f + 45*A^2*B*
a^5*b^4*c^3*d^4*f + 39*A*B^2*a^4*b^5*c^5*d^2*f - 35*A*B^2*a^6*b^3*c^3*d^4*f
 + 33*A*B^2*a^7*b^2*c^2*d^5*f + 27*A^2*B*a^3*b^6*c^5*d^2*f - 21*A*B^2*a^5*b
^4*c^4*d^3*f + 20*A^2*B*a^6*b^3*c^4*d^3*f - 15*A^2*B*a^7*b^2*c^3*d^4*f - 15
*A^2*B*a^5*b^4*c^5*d^2*f + 9*A^2*B*a^2*b^7*c^4*d^3*f + 3*A*B^2*a^2*b^7*c^5*
d^2*f + 2*A*B*C*b^9*c^6*d*f - 6*A*B*C*a^9*c*d^6*f + 18*A*B*C*a^8*b*d^7*f -

6*A*B*C*a*b^8*c^7*f + 63*B^2*C*a^6*b^3*c*d^6*f - 48*B^2*C*a*b^8*c^4*d^3*f +
42*B*C^2*a^8*b*c^2*d^5*f + 42*B*C^2*a^5*b^4*c*d^6*f - 39*B*C^2*a^7*b^2*c*d
^6*f + 30*B*C^2*a*b^8*c^5*d^2*f - 24*B^2*C*a^4*b^5*c*d^6*f - 24*B*C^2*a*b^8
*c^3*d^4*f + 17*B^2*C*a^3*b^6*c^6*d*f - 15*B*C^2*a^2*b^7*c^6*d*f + 12*B^2*C
*a^8*b*c^3*d^4*f + 12*B^2*C*a*b^8*c^2*d^5*f + 6*B*C^2*a^4*b^5*c^6*d*f - 192
*A^2*C*a^4*b^5*c*d^6*f - 99*A^2*C*a^6*b^3*c*d^6*f + 84*A*C^2*a^4*b^5*c*d^6*
f + 59*A*C^2*a^6*b^3*c*d^6*f + 51*A^2*C*a^3*b^6*c^6*d*f - 51*A*C^2*a^3*b^6*
c^6*d*f - 36*A^2*C*a*b^8*c^2*d^5*f - 24*A*C^2*a*b^8*c^4*d^3*f + 24*A*C^2*a*
b^8*c^2*d^5*f + 12*A^2*C*a*b^8*c^4*d^3*f + 12*A*C^2*a^8*b*c^3*d^4*f + 160*A
^2*B*a^3*b^6*c*d^6*f - 99*A*B^2*a^6*b^3*c*d^6*f - 87*A^2*B*a^7*b^2*c*d^6*f
- 72*A*B^2*a^4*b^5*c*d^6*f - 48*A*B^2*a*b^8*c^2*d^5*f - 36*A^2*B*a*b^8*c^3*
d^4*f + 24*A*B^2*a*b^8*c^4*d^3*f - 17*A*B^2*a^3*b^6*c^6*d*f - 15*A^2*B*a^2*
b^7*c^6*d*f + 12*A*B^2*a^2*b^7*c*d^6*f + 6*A^2*B*a^8*b*c^2*d^5*f - 6*A^2*B*
a^5*b^4*c*d^6*f + 6*A^2*B*a^4*b^5*c^6*d*f + 6*A^2*B*a*b^8*c^5*d^2*f + 12*B^
2*C*b^9*c^3*d^4*f - 12*B*C^2*b^9*c^4*d^3*f - 12*A^2*C*b^9*c^3*d^4*f - 8*A*C
^2*b^9*c^5*d^2*f + 8*A*C^2*b^9*c^3*d^4*f + 4*B^2*C*a^9*c^2*d^5*f + 4*A^2*C*
b^9*c^5*d^2*f - 4*B*C^2*a^9*c^3*d^4*f + 12*A^2*B*b^9*c^2*d^5*f - 8*A*B^2*b^
9*c^3*d^4*f - 4*A^2*B*b^9*c^4*d^3*f + 4*A*C^2*a^9*c^2*d^5*f + 3*B^2*C*a^7*b
^2*d^7*f - B*C^2*a^6*b^3*d^7*f + 96*A^2*C*a^5*b^4*d^7*f - 39*A^2*C*a^7*b^2*
d^7*f - 36*A*C^2*a^5*b^4*d^7*f + 32*A^2*C*a^3*b^6*d^7*f + 15*A*C^2*a^7*b^2*
d^7*f - 3*B^2*C*a^2*b^7*c^7*f - B*C^2*a^3*b^6*c^7*f + 111*A^2*B*a^6*b^3*d^7
*f - 39*A*B^2*a^7*b^2*d^7*f + 24*A*B^2*a^5*b^4*d^7*f - 9*A^2*C*a^2*b^7*c^7*
f + 9*A*C^2*a^2*b^7*c^7*f - 4*A*B^2*a^3*b^6*d^7*f + 3*A*B^2*a^2*b^7*c^7*f -
A^2*B*a^3*b^6*c^7*f + 3*C^3*a^8*b*c*d^6*f - 3*C^3*a*b^8*c^6*d*f - 3*A^3*a^
8*b*c*d^6*f + 3*A^3*a*b^8*c^6*d*f - B*C^2*b^9*c^6*d*f + 4*A^2*C*b^9*c*d^6*f
+ 3*B*C^2*a^9*c*d^6*f + 8*A*B^2*b^9*c*d^6*f + 3*B*C^2*a^8*b*d^7*f - A^2*B*
b^9*c^6*d*f + 12*A^2*C*a*b^8*d^7*f + 3*B*C^2*a*b^8*c^7*f - A^2*B*a^9*c*d^6*
f - 9*A^2*B*a^8*b*d^7*f + 3*A^2*B*a*b^8*c^7*f - 39*C^3*a^5*b^4*c^4*d^3*f +
39*C^3*a^4*b^5*c^3*d^4*f + 27*C^3*a^7*b^2*c^2*d^5*f - 27*C^3*a^2*b^7*c^5*d^
2*f - 17*C^3*a^6*b^3*c^3*d^4*f + 17*C^3*a^3*b^6*c^4*d^3*f + 3*C^3*a^5*b^4*c
^2*d^5*f - 3*C^3*a^4*b^5*c^5*d^2*f - 63*B^3*a^5*b^4*c^3*d^4*f + 57*B^3*a^4*
b^5*c^2*d^5*f - 51*B^3*a^2*b^7*c^4*d^3*f + 48*B^3*a^3*b^6*c^3*d^4*f + 31*B^
3*a^6*b^3*c^2*d^5*f + 27*B^3*a^3*b^6*c^5*d^2*f + 16*B^3*a^6*b^3*c^4*d^3*f -
15*B^3*a^5*b^4*c^5*d^2*f - 12*B^3*a^2*b^7*c^2*d^5*f + 9*B^3*a^4*b^5*c^4*d^
3*f - 3*B^3*a^7*b^2*c^3*d^4*f - 123*A^3*a^5*b^4*c^2*d^5*f + 81*A^3*a^4*b^5*
c^3*d^4*f - 45*A^3*a^5*b^4*c^4*d^3*f + 39*A^3*a^4*b^5*c^5*d^2*f + 25*A^3*a^
6*b^3*c^3*d^4*f - 25*A^3*a^3*b^6*c^4*d^3*f - 24*A^3*a^2*b^7*c^3*d^4*f - 8*A
^3*a^3*b^6*c^2*d^5*f - 3*A^3*a^7*b^2*c^2*d^5*f + 3*A^3*a^2*b^7*c^5*d^2*f -
17*C^3*a^6*b^3*c*d^6*f + 17*C^3*a^3*b^6*c^6*d*f - 12*C^3*a^8*b*c^3*d^4*f +
12*C^3*a*b^8*c^4*d^3*f + 24*B^3*a*b^8*c^3*d^4*f + 21*B^3*a^7*b^2*c*d^6*f -
18*B^3*a^5*b^4*c*d^6*f - 15*B^3*a^2*b^7*c^6*d*f - 6*B^3*a^8*b*c^2*d^5*f + 6
*B^3*a^4*b^5*c^6*d*f + 6*B^3*a*b^8*c^5*d^2*f + 4*B^3*a^3*b^6*c*d^6*f + 108*
A^3*a^4*b^5*c*d^6*f + 57*A^3*a^6*b^3*c*d^6*f - 17*A^3*a^3*b^6*c^6*d*f + 12*
A^3*a*b^8*c^2*d^5*f + 4*C^3*b^9*c^5*d^2*f - 4*C^3*a^9*c^2*d^5*f - 4*B^3*b^9
*c^2*d^5*f + 4*A^3*b^9*c^3*d^4*f + 3*C^3*a^7*b^2*d^7*f - 3*C^3*a^2*b^7*c^7*

f - B^3*a^6*b^3*d^7*f - 60*A^3*a^5*b^4*d^7*f - 32*A^3*a^3*b^6*d^7*f + 21*A^
3*a^7*b^2*d^7*f - B^3*a^3*b^6*c^7*f + 3*A^3*a^2*b^7*c^7*f - B^3*b^9*c^6*d*f
 - 4*A^3*b^9*c*d^6*f - B^3*a^9*c*d^6*f + 3*B^3*a^8*b*d^7*f - 12*A^3*a*b^8*d
^7*f + 3*B^3*a*b^8*c^7*f - B^2*C*a^9*d^7*f - 4*A^2*B*b^9*d^7*f + 3*A^2*C*b^
9*c^7*f - 3*A*C^2*b^9*c^7*f - A*C^2*a^9*d^7*f - A*B^2*b^9*c^7*f - C^3*a^9*d
^7*f - A^3*b^9*c^7*f + B^2*C*b^9*c^7*f + A^2*C*a^9*d^7*f + A*B^2*a^9*d^7*f
 + C^3*b^9*c^7*f + A^3*a^9*d^7*f - 6*A*B^2*C*a^5*b*c*d^5 - 21*A^2*B*C*a^3*b^
3*c^2*d^4 + 21*A*B*C^2*a^3*b^3*c^2*d^4 + 12*A*B^2*C*a^4*b^2*c^2*d^4 - 12*A*
B^2*C*a^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^4*b^2*c^3*
d^3 + 3*A^2*B*C*a^4*b^2*c^3*d^3 + 3*A^2*B*C*a^2*b^4*c^3*d^3 + 3*A*B^2*C*a^2
*b^4*c^4*d^2 + 3*A*B*C^2*a^2*b^4*c^3*d^3 + 2*A*B*C^2*a^3*b^3*c^4*d^2 - A^2*
B*C*a^3*b^3*c^4*d^2 + 18*A^2*B*C*a^2*b^4*c*d^5 + 10*A*B^2*C*a^3*b^3*c*d^5 +
 9*A^2*B*C*a^4*b^2*c*d^5 - 9*A*B*C^2*a^4*b^2*c*d^5 - 9*A*B*C^2*a^2*b^4*c*d^
5 - 6*A^2*B*C*a*b^5*c^2*d^4 + 6*A*B^2*C*a*b^5*c^3*d^3 + 6*A*B*C^2*a^5*b*c^2
*d^4 - 6*A*B*C^2*a*b^5*c^4*d^2 - 3*A^2*B*C*a^5*b*c^2*d^4 + 3*A^2*B*C*a*b^5*
c^4*d^2 + 3*A*B*C^2*a*b^5*c^2*d^4 - 3*B^3*C*a^5*b*c^2*d^4 + 3*B^3*C*a^4*b^2
*c*d^5 + 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a^5*b*c*d^5 - 3*B*C^3*a^5*b*c^2*
d^4 + 3*B*C^3*a^4*b^2*c*d^5 + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a^3*b^3*c*d^
5 + 8*A*C^3*a^3*b^3*c*d^5 - 9*A^3*B*a^2*b^4*c*d^5 - 9*A*B^3*a^2*b^4*c*d^5 -
 3*A^3*B*a^4*b^2*c*d^5 + 3*A^3*B*a*b^5*c^2*d^4 + 3*A^2*B^2*a^5*b*c*d^5 - 3*
A*B^3*a^4*b^2*c*d^5 + 3*A*B^3*a*b^5*c^2*d^4 + 5*A*B*C^2*b^6*c^3*d^3 - 4*A^2
*B*C*b^6*c^3*d^3 - A*B^2*C*b^6*c^4*d^2 - 3*A*B^2*C*a^4*b^2*d^6 - 2*A^2*B*C*
a^3*b^3*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^4*b^2*c^2*d^4 + 6*B^2
*C^2*a^2*b^4*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 24*A^2*C^2*a^3*b^3*c^3*d
^3 - 15*A^2*C^2*a^4*b^2*c^2*d^4 - 9*A^2*C^2*a^2*b^4*c^4*d^2 + 3*A^2*C^2*a^2
*b^4*c^2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^4*b^2*c^2*d^4 + 4*A^
2*B*C*b^6*c*d^5 - 2*A*B*C^2*b^6*c*d^5 + 2*A*B*C^2*a^6*c*d^5 - A^2*B*C*a^6*c
*d^5 + 6*A^2*B*C*a^5*b*d^6 - 3*A*B*C^2*a^5*b*d^6 - 7*B^3*C*a^3*b^3*c^2*d^4
 - 7*B*C^3*a^3*b^3*c^2*d^4 + 3*B^3*C*a^4*b^2*c^3*d^3 - 3*B^3*C*a^2*b^4*c^3*d
^3 - 3*B^2*C^2*a*b^5*c^3*d^3 + 3*B*C^3*a^4*b^2*c^3*d^3 - 3*B*C^3*a^2*b^4*c^
3*d^3 - B^3*C*a^3*b^3*c^4*d^2 - B^2*C^2*a^3*b^3*c*d^5 - B*C^3*a^3*b^3*c^4*d
^2 - 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^4*b^2
*c^2*d^4 + 9*A*C^3*a^2*b^4*c^4*d^2 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^4*
b^2*c^2*d^4 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^2*b^4*c^4*d^2 - 9*A^2*B^2
*a^3*b^3*c*d^5 + 7*A^3*B*a^3*b^3*c^2*d^4 + 7*A*B^3*a^3*b^3*c^2*d^4 - 3*A^3*
B*a^2*b^4*c^3*d^3 - 3*A^2*B^2*a*b^5*c^3*d^3 - 3*A*B^3*a^2*b^4*c^3*d^3 - 5*A
^2*C^2*b^6*c^2*d^4 + 3*A^2*C^2*b^6*c^4*d^2 + 12*A^2*C^2*a^4*b^2*d^6 + 3*A^2
*C^2*a^2*b^4*d^6 + 6*A^2*B^2*a^4*b^2*d^6 + 3*A^2*B^2*a^2*b^4*d^6 + A*B*C^2*
a^3*b^3*d^6 - 3*B^4*a*b^5*c^3*d^3 - B^4*a^3*b^3*c*d^5 + A^2*B^2*a^3*b^3*c^3
*d^3 - 8*A^4*a^3*b^3*c*d^5 - 2*B^3*C*b^6*c^3*d^3 - 2*B*C^3*b^6*c^3*d^3 + 4*
A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 + 2*A*C^3*b^6*c^2*d^4 - A^3*C*b^6*c
^4*d^2 - 2*A*C^3*a^6*c^2*d^4 - 15*A^3*C*a^4*b^2*d^6 - 6*A^3*C*a^2*b^4*d^6 -
 3*A*C^3*a^4*b^2*d^6 + 3*B^4*a^5*b*c*d^5 - B^3*C*a^6*c*d^5 - B*C^3*a^6*c*d^
5 - 2*A^3*B*b^6*c*d^5 - 2*A*B^3*b^6*c*d^5 - 3*A^3*B*a^5*b*d^6 - 3*A*B^3*a^5
*b*d^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*

```
d^2 + 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^4*b^2*c^2*d^4 + 3*A^4*a^2*b^4*c^2*d^4
 + B^2*C^2*b^6*c^4*d^2 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^6*c^2*d^4 + A^2*C^
2*a^6*c^2*d^4 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A
^3*B*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 - A^4*b^6*c^2*d^4 + 6*A^4*a^4*b^2*d^6
+ 3*A^4*a^2*b^4*d^6 - 2*A^2*C^2*a^6*d^6 + A*B^2*C*a^6*d^6 + B^4*a^3*b^3*c^3
*d^3 + A^3*C*a^6*d^6 + A*C^3*a^6*d^6 + C^4*b^6*c^4*d^2 + C^4*a^6*c^2*d^4 +
B^4*b^6*c^2*d^4 + A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k), k
, 1, 4))/f
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: NotImplementedError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**3/(c+d*tan(f*x
+e)),x)
```

```
[Out] Exception raised: NotImplementedError
```

3.77 $\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$

Optimal. Leaf size=579

$$\frac{\log(\cos(e+fx))\left(a^3\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+3a^2b\left(-A\left(c^2-d^2\right)-2Bcd+c^2C-Cd^2\right)-3ab^2\left(2cd(A-C)\right.\right.}{f\left(c^2+d^2\right)^2}$$

```
[Out] -(a^3*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-3*a*b^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2
-d^2))-3*a^2*b*(2*c*(A-C)*d-B*(c^2-d^2))+b^3*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(
c^2+d^2)^2+(3*a^2*b*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-b^3*(c^2*C-2*B*c*d-C*
d^2-A*(c^2-d^2))+a^3*(2*c*(A-C)*d-B*(c^2-d^2))-3*a*b^2*(2*c*(A-C)*d-B*(c^2-
d^2)))*ln(cos(f*x+e))/(c^2+d^2)^2/f+(-a*d+b*c)^2*(b*(3*c^4*C-2*B*c^3*d+c^2*
(A+5*C)*d^2-4*B*c*d^3+3*A*d^4)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*ln(c+d*tan(
f*x+e))/d^4/(c^2+d^2)^2/f+b^2*(a*d*(3*c^2*C-B*c*d+(A+2*C)*d^2)-b*(3*c^3*C-2
*B*c^2*d+c*(A+2*C)*d^2-B*d^3))*tan(f*x+e)/d^3/(c^2+d^2)/f+1/2*b*(3*c^2*C-2*
B*c*d+(2*A+C)*d^2)*(a+b*tan(f*x+e))^2/d^2/(c^2+d^2)/f-(A*d^2-B*c*d+C*c^2)*(
a+b*tan(f*x+e))^3/d/(c^2+d^2)/f/(c+d*tan(f*x+e))
```

Rubi [A] time = 2.13, antiderivative size = 579, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 7, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.156, Rules used = {3645, 3647, 3637, 3626, 3617, 31, 3475}

$$\frac{\log(\cos(e+fx))\left(3a^2b\left(-A\left(c^2-d^2\right)-2Bcd+c^2C-Cd^2\right)+a^3\left(2cd(A-C)-B\left(c^2-d^2\right)\right)-3ab^2\left(2cd(A-C)\right.\right.}{f\left(c^2+d^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^2,x]
```

```
[Out] -(((a^3*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - 3*a*b^2*(c^2*C - 2*B*c*
d - C*d^2 - A*(c^2 - d^2)) - 3*a^2*b*(2*c*(A - C)*d - B*(c^2 - d^2)) + b^3*
(2*c*(A - C)*d - B*(c^2 - d^2)))*x)/(c^2 + d^2)^2) + ((3*a^2*b*(c^2*C - 2*B
*c*d - C*d^2 - A*(c^2 - d^2)) - b^3*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2
)) + a^3*(2*c*(A - C)*d - B*(c^2 - d^2)) - 3*a*b^2*(2*c*(A - C)*d - B*(c^2
- d^2)))*Log[Cos[e + f*x]])/((c^2 + d^2)^2*f) + ((b*c - a*d)^2*(b*(3*c^4*C
- 2*B*c^3*d + c^2*(A + 5*C)*d^2 - 4*B*c*d^3 + 3*A*d^4) + a*d^2*(2*c*(A - C)
*d - B*(c^2 - d^2)))*Log[c + d*Tan[e + f*x]])/(d^4*(c^2 + d^2)^2*f) + (b^2*
(a*d*(3*c^2*C - B*c*d + (A + 2*C)*d^2) - b*(3*c^3*C - 2*B*c^2*d + c*(A + 2*
C)*d^2 - B*d^3))*Tan[e + f*x])/(d^3*(c^2 + d^2)*f) + (b*(3*c^2*C - 2*B*c*d
+ (2*A + C)*d^2)*(a + b*Tan[e + f*x])^2)/(2*d^2*(c^2 + d^2)*f) - ((c^2*C -
B*c*d + A*d^2)*(a + b*Tan[e + f*x])^3)/(d*(c^2 + d^2)*f*(c + d*Tan[e + f*x]
))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
```

```
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx = -\frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{d\left(c^2+d^2\right)f(c+d\tan(e+fx))} + \cdots$$

$$= \frac{b\left(3c^2C-2Bcd+(2A+C)d^2\right)(a+b\tan(e \cdots}{2d^2\left(c^2+d^2\right)f}$$

$$= \frac{b^2\left(ad\left(3c^2C-Bcd+(A+2C)d^2\right)-b\left(3c^3C \cdots\right.\right.}{d^3\left(c^2 \cdots\right.}$$

$$= -\frac{\left(a^3\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-3a \cdots\right.}{\cdots}$$

$$= -\frac{\left(a^3\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-3a \cdots\right.}{\cdots}$$

$$= -\frac{\left(a^3\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-3a \cdots\right.}{\cdots}$$

Mathematica [C] time = 8.55, size = 2463, normalized size = 4.25

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^2,x]
```

```
[Out] ((a^3*A*c^2 - 3*a*A*b^2*c^2 - 3*a^2*b*B*c^2 + b^3*B*c^2 - a^3*c^2*C + 3*a*b
^2*c^2*C + 6*a^2*A*b*c*d - 2*A*b^3*c*d + 2*a^3*B*c*d - 6*a*b^2*B*c*d - 6*a^
2*b*c*C*d + 2*b^3*c*C*d - a^3*A*d^2 + 3*a*A*b^2*d^2 + 3*a^2*b*B*d^2 - b^3*B
*d^2 + a^3*C*d^2 - 3*a*b^2*C*d^2)*(e + f*x)*Cos[e + f*x]*(c*Cos[e + f*x] +
d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^3)/((c - I*d)^2*(c + I*d)^2*f*(a*Cos
[e + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^2) + (((3*I)*b^3*c^11*C*
d^3 - (2*I)*b^3*B*c^10*d^4 - (6*I)*a*b^2*c^10*C*d^4 + 3*b^3*c^10*C*d^4 + I*
A*b^3*c^9*d^5 + (3*I)*a*b^2*B*c^9*d^5 - 2*b^3*B*c^9*d^5 + (3*I)*a^2*b*c^9*C
*d^5 - 6*a*b^2*c^9*C*d^5 + (8*I)*b^3*c^9*C*d^5 + A*b^3*c^8*d^6 + 3*a*b^2*B*
c^8*d^6 - (6*I)*b^3*B*c^8*d^6 + 3*a^2*b*c^8*C*d^6 - (18*I)*a*b^2*c^8*C*d^6
+ 8*b^3*c^8*C*d^6 - (3*I)*a^2*A*b*c^7*d^7 + (4*I)*A*b^3*c^7*d^7 - I*a^3*B*c
^7*d^7 + (12*I)*a*b^2*B*c^7*d^7 - 6*b^3*B*c^7*d^7 + (12*I)*a^2*b*c^7*C*d^7
- 18*a*b^2*c^7*C*d^7 + (5*I)*b^3*c^7*C*d^7 + (2*I)*a^3*A*c^6*d^8 - 3*a^2*A*
b*c^6*d^8 - (6*I)*a*A*b^2*c^6*d^8 + 4*A*b^3*c^6*d^8 - a^3*B*c^6*d^8 - (6*I)
*a^2*b*B*c^6*d^8 + 12*a*b^2*B*c^6*d^8 - (4*I)*b^3*B*c^6*d^8 - (2*I)*a^3*c^6
*C*d^8 + 12*a^2*b*c^6*C*d^8 - (12*I)*a*b^2*c^6*C*d^8 + 5*b^3*c^6*C*d^8 + 2*
a^3*A*c^5*d^9 - 6*a*A*b^2*c^5*d^9 + (3*I)*A*b^3*c^5*d^9 - 6*a^2*b*B*c^5*d^9
 + (9*I)*a*b^2*B*c^5*d^9 - 4*b^3*B*c^5*d^9 - 2*a^3*c^5*C*d^9 + (9*I)*a^2*b*
c^5*C*d^9 - 12*a*b^2*c^5*C*d^9 + (2*I)*a^3*A*c^4*d^10 - (6*I)*a*A*b^2*c^4*d
^10 + 3*A*b^3*c^4*d^10 - (6*I)*a^2*b*B*c^4*d^10 + 9*a*b^2*B*c^4*d^10 - (2*I
)*a^3*c^4*C*d^10 + 9*a^2*b*c^4*C*d^10 + 2*a^3*A*c^3*d^11 + (3*I)*a^2*A*b*c^
3*d^11 - 6*a*A*b^2*c^3*d^11 + I*a^3*B*c^3*d^11 - 6*a^2*b*B*c^3*d^11 - 2*a^3
*c^3*C*d^11 + 3*a^2*A*b*c^2*d^12 + a^3*B*c^2*d^12)*(e + f*x)*Cos[e + f*x]*(
c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^3)/(c^2*(c - I*d)^4
*(c + I*d)^3*d^7*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])
^2) - (I*(3*b^3*c^6*C - 2*b^3*B*c^5*d - 6*a*b^2*c^5*C*d + A*b^3*c^4*d^2 + 3
*a*b^2*B*c^4*d^2 + 3*a^2*b*c^4*C*d^2 + 5*b^3*c^4*C*d^2 - 4*b^3*B*c^3*d^3 -
12*a*b^2*c^3*C*d^3 - 3*a^2*A*b*c^2*d^4 + 3*A*b^3*c^2*d^4 - a^3*B*c^2*d^4 +
9*a*b^2*B*c^2*d^4 + 9*a^2*b*c^2*C*d^4 + 2*a^3*A*c*d^5 - 6*a*A*b^2*c*d^5 - 6
*a^2*b*B*c*d^5 - 2*a^3*c*C*d^5 + 3*a^2*A*b*d^6 + a^3*B*d^6)*ArcTan[Tan[e +
f*x]]*Cos[e + f*x]*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])
^3)/(d^4*(c^2 + d^2)^2*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e +
 f*x])^2) + ((-3*b^3*c^2*C + 2*b^3*B*c*d + 6*a*b^2*c*C*d - A*b^3*d^2 - 3*a*
b^2*B*d^2 - 3*a^2*b*C*d^2 + b^3*C*d^2)*Cos[e + f*x]*Log[Cos[e + f*x]]*(c*Co
s[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^3)/(d^4*f*(a*Cos[e + f*
x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^2) + ((3*b^3*c^6*C - 2*b^3*B*c^
```

```
5*d - 6*a*b^2*c^5*C*d + A*b^3*c^4*d^2 + 3*a*b^2*B*c^4*d^2 + 3*a^2*b*c^4*C*d
^2 + 5*b^3*c^4*C*d^2 - 4*b^3*B*c^3*d^3 - 12*a*b^2*c^3*C*d^3 - 3*a^2*A*b*c^2
*d^4 + 3*A*b^3*c^2*d^4 - a^3*B*c^2*d^4 + 9*a*b^2*B*c^2*d^4 + 9*a^2*b*c^2*C*
d^4 + 2*a^3*A*c*d^5 - 6*a*A*b^2*c*d^5 - 6*a^2*b*B*c*d^5 - 2*a^3*c*C*d^5 + 3
*a^2*A*b*d^6 + a^3*B*d^6)*Cos[e + f*x]*Log[(c*Cos[e + f*x] + d*Sin[e + f*x]
)^2]*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^3)/(2*d^4*(c^
2 + d^2)^2*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^2) +
(b^3*C*Sec[e + f*x]*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x]
)^3)/(2*d^2*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^2) +
 ((c*Cos[e + f*x] + d*Sin[e + f*x])^2*(-2*b^3*c*C*Sin[e + f*x] + b^3*B*d*Si
n[e + f*x] + 3*a*b^2*C*d*Sin[e + f*x])*(a + b*Tan[e + f*x])^3)/(d^3*f*(a*Co
s[e + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^2) + (Cos[e + f*x]*(c*C
os[e + f*x] + d*Sin[e + f*x])*(-(b^3*c^5*C*Sin[e + f*x]) + b^3*B*c^4*d*Sin[
e + f*x] + 3*a*b^2*c^4*C*d*Sin[e + f*x] - A*b^3*c^3*d^2*Sin[e + f*x] - 3*a*
b^2*B*c^3*d^2*Sin[e + f*x] - 3*a^2*b*c^3*C*d^2*Sin[e + f*x] + 3*a*A*b^2*c^2
*d^3*Sin[e + f*x] + 3*a^2*b*B*c^2*d^3*Sin[e + f*x] + a^3*c^2*C*d^3*Sin[e +
f*x] - 3*a^2*A*b*c*d^4*Sin[e + f*x] - a^3*B*c*d^4*Sin[e + f*x] + a^3*A*d^5*
Sin[e + f*x])*(a + b*Tan[e + f*x])^3)/(c*(c - I*d)*(c + I*d)*d^3*f*(a*Cos[e
 + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^2)
```

fricas [B] time = 2.37, size = 1477, normalized size = 2.55

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] 1/2*(3*C*b^3*c^5*d^2 - 2*A*a^3*d^7 - 2*(3*C*a*b^2 + B*b^3)*c^4*d^3 + 2*(3*C
*a^2*b + 3*B*a*b^2 + (A + C)*b^3)*c^3*d^4 - 2*(C*a^3 + 3*B*a^2*b + 3*A*a*b^
2)*c^2*d^5 + (2*B*a^3 + 6*A*a^2*b + C*b^3)*c*d^6 + (C*b^3*c^4*d^3 + 2*C*b^3
*c^2*d^5 + C*b^3*d^7)*tan(f*x + e)^3 + 2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A -
 C)*a*b^2 + B*b^3)*c^3*d^4 + 2*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A -
C)*b^3)*c^2*d^5 - ((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d^6
)*f*x - (3*C*b^3*c^5*d^2 + 6*C*b^3*c^3*d^4 + 3*C*b^3*c*d^6 - 2*(3*C*a*b^2 +
 B*b^3)*c^4*d^3 - 4*(3*C*a*b^2 + B*b^3)*c^2*d^5 - 2*(3*C*a*b^2 + B*b^3)*d^7
)*tan(f*x + e)^2 + (3*C*b^3*c^7 - 2*(3*C*a*b^2 + B*b^3)*c^6*d + (3*C*a^2*b
+ 3*B*a*b^2 + (A + 5*C)*b^3)*c^5*d^2 - 4*(3*C*a*b^2 + B*b^3)*c^4*d^3 - (B*a
^3 + 3*(A - 3*C)*a^2*b - 9*B*a*b^2 - 3*A*b^3)*c^3*d^4 + 2*((A - C)*a^3 - 3*
B*a^2*b - 3*A*a*b^2)*c^2*d^5 + (B*a^3 + 3*A*a^2*b)*c*d^6 + (3*C*b^3*c^6*d -
 2*(3*C*a*b^2 + B*b^3)*c^5*d^2 + (3*C*a^2*b + 3*B*a*b^2 + (A + 5*C)*b^3)*c^
4*d^3 - 4*(3*C*a*b^2 + B*b^3)*c^3*d^4 - (B*a^3 + 3*(A - 3*C)*a^2*b - 9*B*a*
b^2 - 3*A*b^3)*c^2*d^5 + 2*((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*c*d^6 + (B
*a^3 + 3*A*a^2*b)*d^7)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*
x + e) + c^2)/(tan(f*x + e)^2 + 1)) - (3*C*b^3*c^7 - 2*(3*C*a*b^2 + B*b^3)*
```

```
c^6*d + (3*C*a^2*b + 3*B*a*b^2 + (A + 5*C)*b^3)*c^5*d^2 - 4*(3*C*a*b^2 + B*
b^3)*c^4*d^3 + (6*C*a^2*b + 6*B*a*b^2 + (2*A + C)*b^3)*c^3*d^4 - 2*(3*C*a*b
^2 + B*b^3)*c^2*d^5 + (3*C*a^2*b + 3*B*a*b^2 + (A - C)*b^3)*c*d^6 + (3*C*b^
3*c^6*d - 2*(3*C*a*b^2 + B*b^3)*c^5*d^2 + (3*C*a^2*b + 3*B*a*b^2 + (A + 5*C
)*b^3)*c^4*d^3 - 4*(3*C*a*b^2 + B*b^3)*c^3*d^4 + (6*C*a^2*b + 6*B*a*b^2 + (
2*A + C)*b^3)*c^2*d^5 - 2*(3*C*a*b^2 + B*b^3)*c*d^6 + (3*C*a^2*b + 3*B*a*b^
2 + (A - C)*b^3)*d^7)*tan(f*x + e))*log(1/(tan(f*x + e)^2 + 1)) - (6*C*b^3*
c^6*d - C*b^3*d^7 - 4*(3*C*a*b^2 + B*b^3)*c^5*d^2 + (6*C*a^2*b + 6*B*a*b^2
+ (2*A + 7*C)*b^3)*c^4*d^3 - 2*(C*a^3 + 3*B*a^2*b + 3*(A + 2*C)*a*b^2 + 2*B
*b^3)*c^3*d^4 + 2*(B*a^3 + 3*A*a^2*b + C*b^3)*c^2*d^5 - 2*(A*a^3 + 3*C*a*b^
2 + B*b^3)*c*d^6 - 2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c
^2*d^5 + 2*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d^6 - ((A
- C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*d^7)*f*x)*tan(f*x + e))/((c
^4*d^5 + 2*c^2*d^7 + d^9)*f*tan(f*x + e) + (c^5*d^4 + 2*c^3*d^6 + c*d^8)*f)
```

giac [B] time = 3.65, size = 1355, normalized size = 2.34

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^3*c^2 - C*a^3*c^2 - 3*B*a^2*b*c^2 - 3*A*a*b^2*c^2 + 3*C*a*b^2*c
^2 + B*b^3*c^2 + 2*B*a^3*c*d + 6*A*a^2*b*c*d - 6*C*a^2*b*c*d - 6*B*a*b^2*c*
d - 2*A*b^3*c*d + 2*C*b^3*c*d - A*a^3*d^2 + C*a^3*d^2 + 3*B*a^2*b*d^2 + 3*A
*a*b^2*d^2 - 3*C*a*b^2*d^2 - B*b^3*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d^4) +
 (B*a^3*c^2 + 3*A*a^2*b*c^2 - 3*C*a^2*b*c^2 - 3*B*a*b^2*c^2 - A*b^3*c^2 + C
*b^3*c^2 - 2*A*a^3*c*d + 2*C*a^3*c*d + 6*B*a^2*b*c*d + 6*A*a*b^2*c*d - 6*C*
a*b^2*c*d - 2*B*b^3*c*d - B*a^3*d^2 - 3*A*a^2*b*d^2 + 3*C*a^2*b*d^2 + 3*B*a
*b^2*d^2 + A*b^3*d^2 - C*b^3*d^2)*log(tan(f*x + e)^2 + 1)/(c^4 + 2*c^2*d^2
+ d^4) + 2*(3*C*b^3*c^6 - 6*C*a*b^2*c^5*d - 2*B*b^3*c^5*d + 3*C*a^2*b*c^4*d
^2 + 3*B*a*b^2*c^4*d^2 + A*b^3*c^4*d^2 + 5*C*b^3*c^4*d^2 - 12*C*a*b^2*c^3*d
^3 - 4*B*b^3*c^3*d^3 - B*a^3*c^2*d^4 - 3*A*a^2*b*c^2*d^4 + 9*C*a^2*b*c^2*d^
4 + 9*B*a*b^2*c^2*d^4 + 3*A*b^3*c^2*d^4 + 2*A*a^3*c*d^5 - 2*C*a^3*c*d^5 - 6
*B*a^2*b*c*d^5 - 6*A*a*b^2*c*d^5 + B*a^3*d^6 + 3*A*a^2*b*d^6)*log(abs(d*tan
(f*x + e) + c))/(c^4*d^4 + 2*c^2*d^6 + d^8) - 2*(3*C*b^3*c^6*d*tan(f*x + e)
 - 6*C*a*b^2*c^5*d^2*tan(f*x + e) - 2*B*b^3*c^5*d^2*tan(f*x + e) + 3*C*a^2*
b*c^4*d^3*tan(f*x + e) + 3*B*a*b^2*c^4*d^3*tan(f*x + e) + A*b^3*c^4*d^3*tan
(f*x + e) + 5*C*b^3*c^4*d^3*tan(f*x + e) - 12*C*a*b^2*c^3*d^4*tan(f*x + e)
- 4*B*b^3*c^3*d^4*tan(f*x + e) - B*a^3*c^2*d^5*tan(f*x + e) - 3*A*a^2*b*c^2
*d^5*tan(f*x + e) + 9*C*a^2*b*c^2*d^5*tan(f*x + e) + 9*B*a*b^2*c^2*d^5*tan(
f*x + e) + 3*A*b^3*c^2*d^5*tan(f*x + e) + 2*A*a^3*c*d^6*tan(f*x + e) - 2*C*
a^3*c*d^6*tan(f*x + e) - 6*B*a^2*b*c*d^6*tan(f*x + e) - 6*A*a*b^2*c*d^6*tan
(f*x + e) + B*a^3*d^7*tan(f*x + e) + 3*A*a^2*b*d^7*tan(f*x + e) + 2*C*b^3*c
```

```
^7 - 3*C*a*b^2*c^6*d - B*b^3*c^6*d + 4*C*b^3*c^5*d^2 + C*a^3*c^4*d^3 + 3*B*
a^2*b*c^4*d^3 + 3*A*a*b^2*c^4*d^3 - 9*C*a*b^2*c^4*d^3 - 3*B*b^3*c^4*d^3 - 2
*B*a^3*c^3*d^4 - 6*A*a^2*b*c^3*d^4 + 6*C*a^2*b*c^3*d^4 + 6*B*a*b^2*c^3*d^4
+ 2*A*b^3*c^3*d^4 + 3*A*a^3*c^2*d^5 - C*a^3*c^2*d^5 - 3*B*a^2*b*c^2*d^5 - 3
*A*a*b^2*c^2*d^5 + A*a^3*d^7)/((c^4*d^4 + 2*c^2*d^6 + d^8)*(d*tan(f*x + e)
+ c)) + (C*b^3*d^2*tan(f*x + e)^2 - 4*C*b^3*c*d*tan(f*x + e) + 6*C*a*b^2*d^
2*tan(f*x + e) + 2*B*b^3*d^2*tan(f*x + e))/d^4)/f
```

maple [B] time = 0.29, size = 2250, normalized size = 3.89

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x
)
```

```
[Out] -4/f/d/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*b^3*c^3-2/f*d/(c^2+d^2)^2*ln(c+d*ta
n(f*x+e))*C*a^3*c+3/f/d^4/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*b^3*c^6+5/f/d^2/
(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*b^3*c^4-3/f/(c^2+d^2)^2*C*arctan(tan(f*x+e
))*a*b^2*d^2+2/f/(c^2+d^2)^2*B*arctan(tan(f*x+e))*a^3*c*d-1/f/d/(c^2+d^2)/(
c+d*tan(f*x+e))*C*c^2*a^3+1/f/d^4/(c^2+d^2)/(c+d*tan(f*x+e))*C*c^5*b^3+3/2/
f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*a^2*b*c^2-3/2/f/(c^2+d^2)^2*ln(1+tan(f*x
+e)^2)*A*a^2*b*d^2-3/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a*b^2*c^2+3/f/(c^
2+d^2)/(c+d*tan(f*x+e))*A*a^2*c*b+9/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*a*b^
2*c^2+9/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*a^2*b*c^2+2/f/(c^2+d^2)^2*C*arct
an(tan(f*x+e))*b^3*c*d-3/f/(c^2+d^2)^2*B*arctan(tan(f*x+e))*a^2*b*c^2+3/f/(
c^2+d^2)^2*B*arctan(tan(f*x+e))*a^2*b*d^2+3/f/(c^2+d^2)^2*C*arctan(tan(f*x+
e))*a*b^2*c^2+1/f/d^2/(c^2+d^2)/(c+d*tan(f*x+e))*A*c^3*b^3-1/f/(c^2+d^2)^2*
ln(1+tan(f*x+e)^2)*A*a^3*c*d-3/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*a^2*b*c^2
-2/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*b^3*c*d+3/2/f/(c^2+d^2)^2*ln(1+tan(f*
x+e)^2)*B*a*b^2*d^2-1/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*b^3*c*d+1/f/(c^2+d
^2)^2*ln(1+tan(f*x+e)^2)*C*a^3*c*d-3/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*a
^2*b*c^2-1/f/d^3/(c^2+d^2)/(c+d*tan(f*x+e))*B*c^4*b^3+3/2/f/(c^2+d^2)^2*ln(
1+tan(f*x+e)^2)*C*a^2*b*d^2-3/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a*b^2*c^2+
3/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a*b^2*d^2+2/f*d/(c^2+d^2)^2*ln(c+d*tan
(f*x+e))*A*a^3*c+3/f*d^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*a^2*b+1/f/d^2/(c^
2+d^2)^2*ln(c+d*tan(f*x+e))*A*b^3*c^4-2/f/d^3/(c^2+d^2)^2*ln(c+d*tan(f*x+e)
)*B*b^3*c^5-3/f/d/(c^2+d^2)/(c+d*tan(f*x+e))*A*c^2*a*b^2+3/f/(c^2+d^2)^2*ln
(1+tan(f*x+e)^2)*A*a*b^2*c*d+3/f/d^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*a^2*b
*c^4-6/f/d^3/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*a*b^2*c^5-12/f/d/(c^2+d^2)^2*
ln(c+d*tan(f*x+e))*C*a*b^2*c^3-6/f*d/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*a*b^2
*c+3/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a^2*b*c*d-3/f/(c^2+d^2)^2*ln(1+tan(
f*x+e)^2)*C*a*b^2*c*d+6/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a^2*b*c*d-6/f/(c
^2+d^2)^2*B*arctan(tan(f*x+e))*a*b^2*c*d-6/f/(c^2+d^2)^2*C*arctan(tan(f*x+e
))*a^2*b*c*d+3/f/d^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*a*b^2*c^4-3/f/d/(c^2+
```

```
d^2)/(c+d*tan(f*x+e))*B*c^2*a^2*b+3/f/d^2/(c^2+d^2)/(c+d*tan(f*x+e))*B*c^3*
a*b^2+3/f/d^2/(c^2+d^2)/(c+d*tan(f*x+e))*C*c^3*a^2*b-3/f/d^3/(c^2+d^2)/(c+d
*tan(f*x+e))*C*c^4*a*b^2-6/f*d/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*a^2*b*c+1/f
/(c^2+d^2)^2*B*arctan(tan(f*x+e))*b^3*c^2-1/f/(c^2+d^2)^2*B*arctan(tan(f*x+
e))*b^3*d^2-1/f/(c^2+d^2)^2*C*arctan(tan(f*x+e))*a^3*c^2+1/f/(c^2+d^2)^2*C*
arctan(tan(f*x+e))*a^3*d^2+1/f/(c^2+d^2)/(c+d*tan(f*x+e))*a^3*B*c+1/f*b^3/d
^2*B*tan(f*x+e)+1/2/f*b^3/d^2*C*tan(f*x+e)^2+3/f/(c^2+d^2)^2*ln(c+d*tan(f*x
+e))*A*b^3*c^2-1/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*a^3*c^2-1/f*d/(c^2+d^2)
/(c+d*tan(f*x+e))*A*a^3+1/f*d^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*a^3-1/2/f/
(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*b^3*c^2+1/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^
2)*A*b^3*d^2+1/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a^3*c^2-1/2/f/(c^2+d^2)
^2*ln(1+tan(f*x+e)^2)*B*a^3*d^2+3/f*b^2/d^2*C*tan(f*x+e)*a-2/f*b^3/d^3*C*c*
tan(f*x+e)+1/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*b^3*c^2-1/2/f/(c^2+d^2)^2
*ln(1+tan(f*x+e)^2)*C*b^3*d^2+1/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a^3*c^2-
1/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a^3*d^2
```

maxima [A] time = 0.73, size = 684, normalized size = 1.18

$$\frac{\frac{2\left(\left(\left((A-C)a^3-3\,Ba^2b-3\,(A-C)ab^2+Bb^3\right)c^2+2\left(Ba^3+3\,(A-C)a^2b-3\,Bab^2-(A-C)b^3\right)cd-\left((A-C)a^3-3\,Ba^2b-3\,(A-C)ab^2+Bb^3\right)d^2\right)(fx+e)}{c^4+2\,c^2d^2+d^4}+\frac{2\left(3\,Cb^3c^6-\cdots\right.}{\cdots}}{\cdots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c^2 + 2*(B*a^3
+ 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d - ((A - C)*a^3 - 3*B*a^2*b
 - 3*(A - C)*a*b^2 + B*b^3)*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d^4) + 2*(3*C
*b^3*c^6 - 2*(3*C*a*b^2 + B*b^3)*c^5*d + (3*C*a^2*b + 3*B*a*b^2 + (A + 5*C)
*b^3)*c^4*d^2 - 4*(3*C*a*b^2 + B*b^3)*c^3*d^3 - (B*a^3 + 3*(A - 3*C)*a^2*b
- 9*B*a*b^2 - 3*A*b^3)*c^2*d^4 + 2*((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*c*
d^5 + (B*a^3 + 3*A*a^2*b)*d^6)*log(d*tan(f*x + e) + c)/(c^4*d^4 + 2*c^2*d^6
 + d^8) + ((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c^2 - 2*((A
- C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d - (B*a^3 + 3*(A - C)*a^
2*b - 3*B*a*b^2 - (A - C)*b^3)*d^2)*log(tan(f*x + e)^2 + 1)/(c^4 + 2*c^2*d^
2 + d^4) + 2*(C*b^3*c^5 - A*a^3*d^5 - (3*C*a*b^2 + B*b^3)*c^4*d + (3*C*a^2*
b + 3*B*a*b^2 + A*b^3)*c^3*d^2 - (C*a^3 + 3*B*a^2*b + 3*A*a*b^2)*c^2*d^3 +
(B*a^3 + 3*A*a^2*b)*c*d^4)/(c^3*d^4 + c*d^6 + (c^2*d^5 + d^7)*tan(f*x + e))
 + (C*b^3*d*tan(f*x + e)^2 - 2*(2*C*b^3*c - (3*C*a*b^2 + B*b^3)*d)*tan(f*x
+ e))/d^3)/f
```

mupad [B] time = 16.68, size = 701, normalized size = 1.21

$$\frac{\tan\left(e+f\,x\right)\,\left(\frac{B\,b^3+3\,C\,a\,b^2}{d^2}-\frac{2\,C\,b^3\,c}{d^3}\right)}{f}-\frac{\ln\left(\tan\left(e+f\,x\right)+1{}\mathrm{i}\right)\,\left(B\,a^3-A\,b^3+C\,b^3+3\,A\,a^2\,b-3\,B\,a\,b^2-3\,C\,a^2\,b+\right.}{2\,f\,\left(-c^2+c\,d\,2{}\mathrm{i}+\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^2,x)
```

```
[Out] (tan(e + f*x)*((B*b^3 + 3*C*a*b^2)/d^2 - (2*C*b^3*c)/d^3))/f - (log(tan(e +
 f*x) + 1i)*(A*a^3*1i - A*b^3 + B*a^3 + B*b^3*1i - C*a^3*1i + C*b^3 - A*a*b
^2*3i + 3*A*a^2*b - 3*B*a*b^2 - B*a^2*b*3i + C*a*b^2*3i - 3*C*a^2*b))/(2*f*
(c*d*2i - c^2 + d^2)) + (log(c + d*tan(e + f*x))*(d^4*(3*A*b^3*c^2 - B*a^3*
c^2 - 3*A*a^2*b*c^2 + 9*B*a*b^2*c^2 + 9*C*a^2*b*c^2) - d^5*(2*C*a^3*c - 2*A
*a^3*c + 6*A*a*b^2*c + 6*B*a^2*b*c) - d^3*(4*B*b^3*c^3 + 12*C*a*b^2*c^3) +
d^6*(B*a^3 + 3*A*a^2*b) - d*(2*B*b^3*c^5 + 6*C*a*b^2*c^5) + d^2*(A*b^3*c^4
+ 5*C*b^3*c^4 + 3*B*a*b^2*c^4 + 3*C*a^2*b*c^4) + 3*C*b^3*c^6))/(f*(d^8 + 2*
c^2*d^6 + c^4*d^4)) - (log(tan(e + f*x) - 1i)*(A*a^3 - A*b^3*1i + B*a^3*1i
+ B*b^3 - C*a^3 + C*b^3*1i - 3*A*a*b^2 + A*a^2*b*3i - B*a*b^2*3i - 3*B*a^2*
b + 3*C*a*b^2 - C*a^2*b*3i))/(2*f*(2*c*d - c^2*1i + d^2*1i)) - (A*a^3*d^5 -
 C*b^3*c^5 - B*a^3*c*d^4 + B*b^3*c^4*d - A*b^3*c^3*d^2 + C*a^3*c^2*d^3 + 3*
A*a*b^2*c^2*d^3 - 3*B*a*b^2*c^3*d^2 + 3*B*a^2*b*c^2*d^3 - 3*C*a^2*b*c^3*d^2
 - 3*A*a^2*b*c*d^4 + 3*C*a*b^2*c^4*d)/(d*f*(c*d^3 + d^4*tan(e + f*x))*(c^2
+ d^2)) + (C*b^3*tan(e + f*x)^2)/(2*d^2*f)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**2,x)
```

```
[Out] Timed out
```

3.78 $\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$

Optimal. Leaf size=417

$$\frac{\log(\cos(e+fx))\left(a^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+2ab\left(-A\left(c^2-d^2\right)-2Bcd+c^2C-Cd^2\right)-b^2\left(2cd(A-C)-B\left(\right.\right.\right.}{f\left(c^2+d^2\right)^2}$$

```
[Out] -(a^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-b^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2
))-2*a*b*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(c^2+d^2)^2+(2*a*b*(c^2*C-2*B*c*d-C*d
^2-A*(c^2-d^2))+a^2*(2*c*(A-C)*d-B*(c^2-d^2))-b^2*(2*c*(A-C)*d-B*(c^2-d^2))
)*ln(cos(f*x+e))/(c^2+d^2)^2/f-(-a*d+b*c)*(b*(2*A*d^4-B*c^3*d-3*B*c*d^3+2*C
*c^4+4*C*c^2*d^2)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*ln(c+d*tan(f*x+e))/d^3/(
c^2+d^2)^2/f+b^2*(2*c^2*C-B*c*d+(A+C)*d^2)*tan(f*x+e)/d^2/(c^2+d^2)/f-(A*d^
2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^2/d/(c^2+d^2)/f/(c+d*tan(f*x+e))
```

Rubi [A] time = 1.11, antiderivative size = 417, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3645, 3637, 3626, 3617, 31, 3475}

$$\frac{\log(\cos(e+fx))\left(a^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+2ab\left(-A\left(c^2-d^2\right)-2Bcd+c^2C-Cd^2\right)-b^2\left(2cd(A-C)-B\left(\right.\right.\right.}{f\left(c^2+d^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^2,x]
```

```
[Out] -(((a^2*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C - 2*B*c*d -
C*d^2 - A*(c^2 - d^2)) - 2*a*b*(2*c*(A - C)*d - B*(c^2 - d^2)))*x)/(c^2 + d
^2)^2) + ((2*a*b*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + a^2*(2*c*(A -
C)*d - B*(c^2 - d^2)) - b^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Log[Cos[e + f*
x]])/((c^2 + d^2)^2*f) - ((b*c - a*d)*(b*(2*c^4*C - B*c^3*d + 4*c^2*C*d^2 -
 3*B*c*d^3 + 2*A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Log[c + d*Ta
n[e + f*x]])/(d^3*(c^2 + d^2)^2*f) + (b^2*(2*c^2*C - B*c*d + (A + C)*d^2)*T
an[e + f*x])/(d^2*(c^2 + d^2)*f) - ((c^2*C - B*c*d + A*d^2)*(a + b*Tan[e +
f*x])^2)/(d*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx=-\frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{d\left(c^2+d^2\right)f(c+d\tan(e+fx))}+\frac{\int}{}$$

$$=\frac{b^2\left(2c^2C-Bcd+(A+C)d^2\right)\tan(e+fx)}{d^2\left(c^2+d^2\right)f}-\frac{(}{}$$

$$=-\frac{\left(a^2\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(\right.\right.}{}$$

$$=-\frac{\left(a^2\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(\right.\right.}{}$$

$$=-\frac{\left(a^2\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(\right.\right.}{}$$

<u>**Mathematica**</u> [C] time = 7.94, size = 2636, normalized size = 6.32

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^2,x]
```

```
[Out] (((-2*I)*b^2*c^10*C*d^2 + I*b^2*B*c^9*d^3 + (2*I)*a*b*c^9*C*d^3 - 2*b^2*c^9
*C*d^3 + b^2*B*c^8*d^4 + 2*a*b*c^8*C*d^4 - (6*I)*b^2*c^8*C*d^4 - (2*I)*a*A*
b*c^7*d^5 - I*a^2*B*c^7*d^5 + (4*I)*b^2*B*c^7*d^5 + (8*I)*a*b*c^7*C*d^5 - 6
*b^2*c^7*C*d^5 + (2*I)*a^2*A*c^6*d^6 - 2*a*A*b*c^6*d^6 - (2*I)*A*b^2*c^6*d^
6 - a^2*B*c^6*d^6 - (4*I)*a*b*B*c^6*d^6 + 4*b^2*B*c^6*d^6 - (2*I)*a^2*c^6*C
*d^6 + 8*a*b*c^6*C*d^6 - (4*I)*b^2*c^6*C*d^6 + 2*a^2*A*c^5*d^7 - 2*A*b^2*c^
5*d^7 - 4*a*b*B*c^5*d^7 + (3*I)*b^2*B*c^5*d^7 - 2*a^2*c^5*C*d^7 + (6*I)*a*b
*c^5*C*d^7 - 4*b^2*c^5*C*d^7 + (2*I)*a^2*A*c^4*d^8 - (2*I)*A*b^2*c^4*d^8 -
(4*I)*a*b*B*c^4*d^8 + 3*b^2*B*c^4*d^8 - (2*I)*a^2*c^4*C*d^8 + 6*a*b*c^4*C*d
^8 + 2*a^2*A*c^3*d^9 + (2*I)*a*A*b*c^3*d^9 - 2*A*b^2*c^3*d^9 + I*a^2*B*c^3*
d^9 - 4*a*b*B*c^3*d^9 - 2*a^2*c^3*C*d^9 + 2*a*A*b*c^2*d^10 + a^2*B*c^2*d^10
)*(e + f*x)*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^2)/(c^
2*(c - I*d)^4*(c + I*d)^3*d^5*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*
Tan[e + f*x])^2) - (I*(-2*b^2*c^5*C + b^2*B*c^4*d + 2*a*b*c^4*C*d - 4*b^2*c
^3*C*d^2 - 2*a*A*b*c^2*d^3 - a^2*B*c^2*d^3 + 3*b^2*B*c^2*d^3 + 6*a*b*c^2*C*
```

```
d^3 + 2*a^2*A*c*d^4 - 2*A*b^2*c*d^4 - 4*a*b*B*c*d^4 - 2*a^2*c*C*d^4 + 2*a*A
*b*d^5 + a^2*B*d^5)*ArcTan[Tan[e + f*x]]*(c*Cos[e + f*x] + d*Sin[e + f*x])^
2*(a + b*Tan[e + f*x])^2)/(d^3*(c^2 + d^2)^2*f*(a*Cos[e + f*x] + b*Sin[e +
f*x])^2*(c + d*Tan[e + f*x])^2) + ((2*b^2*c*C - b^2*B*d - 2*a*b*C*d)*Log[Co
s[e + f*x]]*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*Tan[e + f*x])^2)/(d^
3*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^2) + ((-2*b^2*
c^5*C + b^2*B*c^4*d + 2*a*b*c^4*C*d - 4*b^2*c^3*C*d^2 - 2*a*A*b*c^2*d^3 - a
^2*B*c^2*d^3 + 3*b^2*B*c^2*d^3 + 6*a*b*c^2*C*d^3 + 2*a^2*A*c*d^4 - 2*A*b^2*
c*d^4 - 4*a*b*B*c*d^4 - 2*a^2*c*C*d^4 + 2*a*A*b*d^5 + a^2*B*d^5)*Log[(c*Cos
[e + f*x] + d*Sin[e + f*x])^2]*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(a + b*T
an[e + f*x])^2)/(2*d^3*(c^2 + d^2)^2*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*
(c + d*Tan[e + f*x])^2) + (Sec[e + f*x]*(c*Cos[e + f*x] + d*Sin[e + f*x])*(
b^2*c^5*C*d + 2*b^2*c^3*C*d^3 + b^2*c*C*d^5 + a^2*A*c^4*d^2*(e + f*x) - A*b
^2*c^4*d^2*(e + f*x) - 2*a*b*B*c^4*d^2*(e + f*x) - a^2*c^4*C*d^2*(e + f*x)
+ b^2*c^4*C*d^2*(e + f*x) + 4*a*A*b*c^3*d^3*(e + f*x) + 2*a^2*B*c^3*d^3*(e
+ f*x) - 2*b^2*B*c^3*d^3*(e + f*x) - 4*a*b*c^3*C*d^3*(e + f*x) - a^2*A*c^2*
d^4*(e + f*x) + A*b^2*c^2*d^4*(e + f*x) + 2*a*b*B*c^2*d^4*(e + f*x) + a^2*c
^2*C*d^4*(e + f*x) - b^2*c^2*C*d^4*(e + f*x) - b^2*c^5*C*d*Cos[2*(e + f*x)]
 - 2*b^2*c^3*C*d^3*Cos[2*(e + f*x)] - b^2*c*C*d^5*Cos[2*(e + f*x)] + a^2*A*
c^4*d^2*(e + f*x)*Cos[2*(e + f*x)] - A*b^2*c^4*d^2*(e + f*x)*Cos[2*(e + f*x
)] - 2*a*b*B*c^4*d^2*(e + f*x)*Cos[2*(e + f*x)] - a^2*c^4*C*d^2*(e + f*x)*C
os[2*(e + f*x)] + b^2*c^4*C*d^2*(e + f*x)*Cos[2*(e + f*x)] + 4*a*A*b*c^3*d^
3*(e + f*x)*Cos[2*(e + f*x)] + 2*a^2*B*c^3*d^3*(e + f*x)*Cos[2*(e + f*x)] -
 2*b^2*B*c^3*d^3*(e + f*x)*Cos[2*(e + f*x)] - 4*a*b*c^3*C*d^3*(e + f*x)*Cos
[2*(e + f*x)] - a^2*A*c^2*d^4*(e + f*x)*Cos[2*(e + f*x)] + A*b^2*c^2*d^4*(e
 + f*x)*Cos[2*(e + f*x)] + 2*a*b*B*c^2*d^4*(e + f*x)*Cos[2*(e + f*x)] + a^2
*c^2*C*d^4*(e + f*x)*Cos[2*(e + f*x)] - b^2*c^2*C*d^4*(e + f*x)*Cos[2*(e +
f*x)] + 2*b^2*c^6*C*Sin[2*(e + f*x)] - b^2*B*c^5*d*Sin[2*(e + f*x)] - 2*a*b
*c^5*C*d*Sin[2*(e + f*x)] + A*b^2*c^4*d^2*Sin[2*(e + f*x)] + 2*a*b*B*c^4*d^
2*Sin[2*(e + f*x)] + a^2*c^4*C*d^2*Sin[2*(e + f*x)] + 3*b^2*c^4*C*d^2*Sin[2
*(e + f*x)] - 2*a*A*b*c^3*d^3*Sin[2*(e + f*x)] - a^2*B*c^3*d^3*Sin[2*(e + f
*x)] - b^2*B*c^3*d^3*Sin[2*(e + f*x)] - 2*a*b*c^3*C*d^3*Sin[2*(e + f*x)] +
a^2*A*c^2*d^4*Sin[2*(e + f*x)] + A*b^2*c^2*d^4*Sin[2*(e + f*x)] + 2*a*b*B*c
^2*d^4*Sin[2*(e + f*x)] + a^2*c^2*C*d^4*Sin[2*(e + f*x)] + b^2*c^2*C*d^4*Si
n[2*(e + f*x)] - 2*a*A*b*c*d^5*Sin[2*(e + f*x)] - a^2*B*c*d^5*Sin[2*(e + f*
x)] + a^2*A*d^6*Sin[2*(e + f*x)] + a^2*A*c^3*d^3*(e + f*x)*Sin[2*(e + f*x)]
 - A*b^2*c^3*d^3*(e + f*x)*Sin[2*(e + f*x)] - 2*a*b*B*c^3*d^3*(e + f*x)*Sin
[2*(e + f*x)] - a^2*c^3*C*d^3*(e + f*x)*Sin[2*(e + f*x)] + b^2*c^3*C*d^3*(e
 + f*x)*Sin[2*(e + f*x)] + 4*a*A*b*c^2*d^4*(e + f*x)*Sin[2*(e + f*x)] + 2*a
^2*B*c^2*d^4*(e + f*x)*Sin[2*(e + f*x)] - 2*b^2*B*c^2*d^4*(e + f*x)*Sin[2*(
e + f*x)] - 4*a*b*c^2*C*d^4*(e + f*x)*Sin[2*(e + f*x)] - a^2*A*c*d^5*(e + f
*x)*Sin[2*(e + f*x)] + A*b^2*c*d^5*(e + f*x)*Sin[2*(e + f*x)] + 2*a*b*B*c*d
^5*(e + f*x)*Sin[2*(e + f*x)] + a^2*c*C*d^5*(e + f*x)*Sin[2*(e + f*x)] - b^
2*c*C*d^5*(e + f*x)*Sin[2*(e + f*x)])*(a + b*Tan[e + f*x])^2)/(2*c*(c - I*d
)^2*(c + I*d)^2*d^2*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*
```

x])^2)

fricas [B] time = 1.28, size = 939, normalized size = 2.25

$$-\frac{2\,Cb^2c^4d^2 + 2\,Aa^2d^6 - 2\left(2\,Cab + Bb^2\right)c^3d^3 + 2\left(Ca^2 + 2\,Bab + Ab^2\right)c^2d^4 - 2\left(Ba^2 + 2\,Aab\right)cd^5 - 2\left(\left((A-C)a^2\right.\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] -1/2*(2*C*b^2*c^4*d^2 + 2*A*a^2*d^6 - 2*(2*C*a*b + B*b^2)*c^3*d^3 + 2*(C*a^
2 + 2*B*a*b + A*b^2)*c^2*d^4 - 2*(B*a^2 + 2*A*a*b)*c*d^5 - 2*(((A - C)*a^2
- 2*B*a*b - (A - C)*b^2)*c^3*d^3 + 2*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2*d^
4 - ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^5)*f*x - 2*(C*b^2*c^4*d^2 + 2
*C*b^2*c^2*d^4 + C*b^2*d^6)*tan(f*x + e)^2 + (2*C*b^2*c^6 + 4*C*b^2*c^4*d^2
 - (2*C*a*b + B*b^2)*c^5*d + (B*a^2 + 2*(A - 3*C)*a*b - 3*B*b^2)*c^3*d^3 -
2*((A - C)*a^2 - 2*B*a*b - A*b^2)*c^2*d^4 - (B*a^2 + 2*A*a*b)*c*d^5 + (2*C*
b^2*c^5*d + 4*C*b^2*c^3*d^3 - (2*C*a*b + B*b^2)*c^4*d^2 + (B*a^2 + 2*(A - 3
*C)*a*b - 3*B*b^2)*c^2*d^4 - 2*((A - C)*a^2 - 2*B*a*b - A*b^2)*c*d^5 - (B*a
^2 + 2*A*a*b)*d^6)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x +
e) + c^2)/(tan(f*x + e)^2 + 1)) - (2*C*b^2*c^6 + 4*C*b^2*c^4*d^2 + 2*C*b^2*
c^2*d^4 - (2*C*a*b + B*b^2)*c^5*d - 2*(2*C*a*b + B*b^2)*c^3*d^3 - (2*C*a*b
+ B*b^2)*c*d^5 + (2*C*b^2*c^5*d + 4*C*b^2*c^3*d^3 + 2*C*b^2*c*d^5 - (2*C*a*
b + B*b^2)*c^4*d^2 - 2*(2*C*a*b + B*b^2)*c^2*d^4 - (2*C*a*b + B*b^2)*d^6)*t
an(f*x + e))*log(1/(tan(f*x + e)^2 + 1)) - 2*(2*C*b^2*c^5*d - (2*C*a*b + B*
b^2)*c^4*d^2 + (C*a^2 + 2*B*a*b + (A + 2*C)*b^2)*c^3*d^3 - (B*a^2 + 2*A*a*b
)*c^2*d^4 + (A*a^2 + C*b^2)*c*d^5 + (((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*
c^2*d^4 + 2*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d^5 - ((A - C)*a^2 - 2*B*a*b
- (A - C)*b^2)*d^6)*f*x)*tan(f*x + e))/((c^4*d^4 + 2*c^2*d^6 + d^8)*f*tan(f
*x + e) + (c^5*d^3 + 2*c^3*d^5 + c*d^7)*f)
```

giac [B] time = 3.09, size = 912, normalized size = 2.19

$$\frac{\frac{2\,Cb^2\tan\left(fx+e\right)}{d^2} + \frac{2\left(Aa^2c^2 - Ca^2c^2 - 2\,Babc^2 - Ab^2c^2 + Cb^2c^2 + 2\,Ba^2cd + 4\,Aabcd - 4\,Cabcd - 2\,Bb^2cd - Aa^2d^2 + Ca^2d^2 + 2\,Babd^2 + Ab^2d^2 - Cb^2d^2\right)\left(fx+e\right)}{c^4 + 2\,c^2d^2 + d^4} + }{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*b^2*tan(f*x + e)/d^2 + 2*(A*a^2*c^2 - C*a^2*c^2 - 2*B*a*b*c^2 - A*
b^2*c^2 + C*b^2*c^2 + 2*B*a^2*c*d + 4*A*a*b*c*d - 4*C*a*b*c*d - 2*B*b^2*c*d
 - A*a^2*d^2 + C*a^2*d^2 + 2*B*a*b*d^2 + A*b^2*d^2 - C*b^2*d^2)*(f*x + e)/(
c^4 + 2*c^2*d^2 + d^4) + (B*a^2*c^2 + 2*A*a*b*c^2 - 2*C*a*b*c^2 - B*b^2*c^2
 - 2*A*a^2*c*d + 2*C*a^2*c*d + 4*B*a*b*c*d + 2*A*b^2*c*d - 2*C*b^2*c*d - B*
a^2*d^2 - 2*A*a*b*d^2 + 2*C*a*b*d^2 + B*b^2*d^2)*log(tan(f*x + e)^2 + 1)/(c
^4 + 2*c^2*d^2 + d^4) - 2*(2*C*b^2*c^5 - 2*C*a*b*c^4*d - B*b^2*c^4*d + 4*C*
b^2*c^3*d^2 + B*a^2*c^2*d^3 + 2*A*a*b*c^2*d^3 - 6*C*a*b*c^2*d^3 - 3*B*b^2*c
^2*d^3 - 2*A*a^2*c*d^4 + 2*C*a^2*c*d^4 + 4*B*a*b*c*d^4 + 2*A*b^2*c*d^4 - B*
a^2*d^5 - 2*A*a*b*d^5)*log(abs(d*tan(f*x + e) + c))/(c^4*d^3 + 2*c^2*d^5 +
d^7) + 2*(2*C*b^2*c^5*d*tan(f*x + e) - 2*C*a*b*c^4*d^2*tan(f*x + e) - B*b^2
*c^4*d^2*tan(f*x + e) + 4*C*b^2*c^3*d^3*tan(f*x + e) + B*a^2*c^2*d^4*tan(f*
x + e) + 2*A*a*b*c^2*d^4*tan(f*x + e) - 6*C*a*b*c^2*d^4*tan(f*x + e) - 3*B*
b^2*c^2*d^4*tan(f*x + e) - 2*A*a^2*c*d^5*tan(f*x + e) + 2*C*a^2*c*d^5*tan(f
*x + e) + 4*B*a*b*c*d^5*tan(f*x + e) + 2*A*b^2*c*d^5*tan(f*x + e) - B*a^2*d
^6*tan(f*x + e) - 2*A*a*b*d^6*tan(f*x + e) + C*b^2*c^6 - C*a^2*c^4*d^2 - 2*
B*a*b*c^4*d^2 - A*b^2*c^4*d^2 + 3*C*b^2*c^4*d^2 + 2*B*a^2*c^3*d^3 + 4*A*a*b
*c^3*d^3 - 4*C*a*b*c^3*d^3 - 2*B*b^2*c^3*d^3 - 3*A*a^2*c^2*d^4 + C*a^2*c^2*
d^4 + 2*B*a*b*c^2*d^4 + A*b^2*c^2*d^4 - A*a^2*d^6)/((c^4*d^3 + 2*c^2*d^5 +
d^7)*(d*tan(f*x + e) + c)))/f
```

maple [B] time = 0.27, size = 1554, normalized size = 3.73

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x
)
```

```
[Out] 2/f/d^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*a*b*c^4+2/f/(c^2+d^2)^2*ln(1+tan(f
*x+e)^2)*B*a*b*c*d+2/f/d^2/(c^2+d^2)/(c+d*tan(f*x+e))*C*c^3*a*b-4/f*d/(c^2+
d^2)^2*ln(c+d*tan(f*x+e))*B*a*b*c-2/f/d/(c^2+d^2)/(c+d*tan(f*x+e))*B*a*b*c^
2+4/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a*b*c*d-4/f/(c^2+d^2)^2*C*arctan(tan
(f*x+e))*a*b*c*d-1/f/d/(c^2+d^2)/(c+d*tan(f*x+e))*C*a^2*c^2-1/f/d^3/(c^2+d^
2)/(c+d*tan(f*x+e))*C*c^4*b^2+2/f*d/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*a^2*c+
2/f*d^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*a*b-2/f*d/(c^2+d^2)^2*ln(c+d*tan(f
*x+e))*A*b^2*c+1/f/d^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*b^2*c^4-2/f/(c^2+d^
2)^2*B*arctan(tan(f*x+e))*a*b*c^2+2/f/(c^2+d^2)^2*B*arctan(tan(f*x+e))*a*b*
d^2-2/f*d/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*a^2*c-2/f/d^3/(c^2+d^2)^2*ln(c+d
*tan(f*x+e))*C*b^2*c^5-4/f/d/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*b^2*c^3-1/f/(
c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*a^2*c*d-1/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*
C*b^2*c*d+2/f/(c^2+d^2)/(c+d*tan(f*x+e))*A*a*b*c+6/f/(c^2+d^2)^2*ln(c+d*tan
(f*x+e))*C*a*b*c^2-2/f/(c^2+d^2)^2*B*arctan(tan(f*x+e))*b^2*c*d+1/f/(c^2+d^
2)^2*ln(1+tan(f*x+e)^2)*A*b^2*c*d+1/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*a^2*
c*d-1/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*a*b*c^2+1/f/(c^2+d^2)^2*ln(1+tan(f
```

```
*x+e)^2)*C*a*b*d^2+2/f/(c^2+d^2)^2*B*arctan(tan(f*x+e))*a^2*c*d-1/f/d/(c^2+
d^2)/(c+d*tan(f*x+e))*A*b^2*c^2+1/f/d^2/(c^2+d^2)/(c+d*tan(f*x+e))*B*c^3*b^
2-2/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*a*b*c^2+1/f/(c^2+d^2)^2*ln(1+tan(f*x
+e)^2)*A*a*b*c^2-1/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*a*b*d^2-1/2/f/(c^2+d^
2)^2*ln(1+tan(f*x+e)^2)*B*b^2*c^2+1/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*b^
2*d^2-1/f*d/(c^2+d^2)/(c+d*tan(f*x+e))*A*a^2+1/f*d^2/(c^2+d^2)^2*ln(c+d*tan
(f*x+e))*B*a^2+1/f/(c^2+d^2)/(c+d*tan(f*x+e))*B*a^2*c-1/f/(c^2+d^2)^2*ln(c+
d*tan(f*x+e))*B*a^2*c^2+3/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*b^2*c^2-1/f/(c
^2+d^2)^2*A*arctan(tan(f*x+e))*a^2*d^2-1/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))
*b^2*c^2+1/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*b^2*d^2-1/f/(c^2+d^2)^2*C*arc
tan(tan(f*x+e))*a^2*c^2+1/f/(c^2+d^2)^2*C*arctan(tan(f*x+e))*a^2*d^2+1/f/(c
^2+d^2)^2*A*arctan(tan(f*x+e))*a^2*c^2+1/f/(c^2+d^2)^2*C*arctan(tan(f*x+e))
*b^2*c^2-1/f/(c^2+d^2)^2*C*arctan(tan(f*x+e))*b^2*d^2+1/2/f/(c^2+d^2)^2*ln(
1+tan(f*x+e)^2)*B*a^2*c^2-1/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a^2*d^2+1/
f*b^2*C/d^2*tan(f*x+e)
```

maxima [A] time = 0.54, size = 493, normalized size = 1.18

$$\frac{\frac{2\,Cb^2\tan\left(fx+e\right)}{d^2}+\frac{2\left(\left(\left(A-C\right)a^2-2\,Bab-\left(A-C\right)b^2\right)c^2+2\left(Ba^2+2\left(A-C\right)ab-Bb^2\right)cd-\left(\left(A-C\right)a^2-2\,Bab-\left(A-C\right)b^2\right)d^2\right)\left(fx+e\right)}{c^4+2\,c^2d^2+d^4}-\frac{2\left(2\,Cb^2c^5+4\,Cb^2c^3d^2-\left(\right.\right.}{}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*b^2*tan(f*x + e)/d^2 + 2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^
2 + 2*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d - ((A - C)*a^2 - 2*B*a*b - (A - C
)*b^2)*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d^4) - 2*(2*C*b^2*c^5 + 4*C*b^2*c^
3*d^2 - (2*C*a*b + B*b^2)*c^4*d + (B*a^2 + 2*(A - 3*C)*a*b - 3*B*b^2)*c^2*d
^3 - 2*((A - C)*a^2 - 2*B*a*b - A*b^2)*c*d^4 - (B*a^2 + 2*A*a*b)*d^5)*log(d
*tan(f*x + e) + c)/(c^4*d^3 + 2*c^2*d^5 + d^7) + ((B*a^2 + 2*(A - C)*a*b -
B*b^2)*c^2 - 2*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d - (B*a^2 + 2*(A -
C)*a*b - B*b^2)*d^2)*log(tan(f*x + e)^2 + 1)/(c^4 + 2*c^2*d^2 + d^4) - 2*(C
*b^2*c^4 + A*a^2*d^4 - (2*C*a*b + B*b^2)*c^3*d + (C*a^2 + 2*B*a*b + A*b^2)*
c^2*d^2 - (B*a^2 + 2*A*a*b)*c*d^3)/(c^3*d^3 + c*d^5 + (c^2*d^4 + d^6)*tan(f
*x + e)))/f
```

mupad [B] time = 35.26, size = 3958, normalized size = 9.49

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^2,x)
```

```
[Out] (log((2*C^2*b^4*c^5 - 2*C^2*a^2*b^2*c^5 + 4*C^2*b^4*c^3*d^2 - A*B*a^4*d^5 -
 2*A*C*b^4*c^5 + B*C*a^4*d^5 + 2*A^2*a*b^3*d^5 - 2*A^2*a^3*b*d^5 - A^2*a^4*
c*d^4 + 2*B^2*a^3*b*d^5 - A^2*b^4*c*d^4 + B^2*a^4*c*d^4 + B^2*b^4*c*d^4 - C
^2*a^4*c*d^4 + C^2*b^4*c*d^4 - 4*C^2*a^2*b^2*c^3*d^2 + 5*A*B*a^2*b^2*d^5 +
2*A*C*a^2*b^2*c^5 + A*B*a^4*c^2*d^3 + 3*A*B*b^4*c^2*d^3 - B*C*a^2*b^2*d^5 -
 4*A*C*b^4*c^3*d^2 - B*C*a^4*c^2*d^3 - 3*B*C*b^4*c^2*d^3 + 2*B^2*a*b^3*c^4*
d - 2*C^2*a*b^3*c^4*d + 2*C^2*a^3*b*c^4*d - 2*A^2*a*b^3*c^2*d^3 + 6*A^2*a^2
*b^2*c*d^4 + 2*A^2*a^3*b*c^2*d^3 + 6*B^2*a*b^3*c^2*d^3 - 6*B^2*a^2*b^2*c*d^
4 - 2*B^2*a^3*b*c^2*d^3 - 6*C^2*a*b^3*c^2*d^3 + 4*C^2*a^2*b^2*c*d^4 + 6*C^2
*a^3*b*c^2*d^3 - 2*A*C*a*b^3*d^5 + 2*A*C*a^3*b*d^5 - 4*B*C*a*b^3*c^5 + A*B*
b^4*c^4*d + 2*A*C*a^4*c*d^4 - B*C*b^4*c^4*d - 8*A*B*a*b^3*c*d^4 + 8*A*B*a^3
*b*c*d^4 + 2*A*C*a*b^3*c^4*d - 2*A*C*a^3*b*c^4*d + 4*B*C*a*b^3*c*d^4 - 8*B*
C*a^3*b*c*d^4 - A*B*a^2*b^2*c^4*d + 8*A*C*a*b^3*c^2*d^3 - 10*A*C*a^2*b^2*c*
d^4 - 8*A*C*a^3*b*c^2*d^3 - 8*B*C*a*b^3*c^3*d^2 + 5*B*C*a^2*b^2*c^4*d - 8*A
*B*a^2*b^2*c^2*d^3 + 4*A*C*a^2*b^2*c^3*d^2 + 16*B*C*a^2*b^2*c^2*d^3)/(d^2*(
c^2 + d^2)^2) + ((a*1i - b)^2*((A*b^2*d^2 - A*a^2*d^2 + C*a^2*d^2 - 8*C*b^2
*c^2 - C*b^2*d^2 + 2*B*a*b*d^2 + 4*B*b^2*c*d + 8*C*a*b*c*d)/d - (tan(e + f*
x)*(3*B*a^2*d^5 - 5*B*b^2*d^5 - 4*C*b^2*c^5 + 6*A*a*b*d^5 - 10*C*a*b*d^5 +
4*A*a^2*c*d^4 - 4*A*b^2*c*d^4 + 2*B*b^2*c^4*d - 4*C*a^2*c*d^4 + 8*C*b^2*c*d
^4 - B*a^2*c^2*d^3 + B*b^2*c^2*d^3 - 8*B*a*b*c*d^4 + 4*C*a*b*c^4*d - 2*A*a*
b*c^2*d^3 + 2*C*a*b*c^2*d^3))/(d^2*(c^2 + d^2)) + (d*(a*1i - b)^2*(4*c*d -
c^2*tan(e + f*x) + 3*d^2*tan(e + f*x))*(A + B*1i - C)*1i)/(c*1i - d)^2)*(A
+ B*1i - C)*1i)/(2*(c*1i - d)^2) + (tan(e + f*x)*(A^2*a^4*d^5 + A^2*b^4*d^5
 + B^2*b^4*d^5 + C^2*a^4*d^5 + C^2*b^4*d^5 - 2*A^2*a^2*b^2*d^5 + 3*B^2*a^2*
b^2*d^5 + B^2*a^4*c^2*d^3 + 2*C^2*a^2*b^2*d^5 + 3*B^2*b^4*c^2*d^3 - 2*A*C*a
^4*d^5 - 2*A*C*b^4*d^5 - 2*B*C*b^4*c^5 - 4*C^2*a*b^3*c^5 + B^2*b^4*c^4*d +
4*A^2*a^2*b^2*c^2*d^3 - 4*B^2*a^2*b^2*c^2*d^3 + 12*C^2*a^2*b^2*c^2*d^3 + 2*
B*C*a^2*b^2*c^5 - 4*B*C*b^4*c^3*d^2 + 4*A^2*a*b^3*c*d^4 - 4*A^2*a^3*b*c*d^4
 - 4*B^2*a*b^3*c*d^4 + 4*B^2*a^3*b*c*d^4 - 4*C^2*a^3*b*c*d^4 - B^2*a^2*b^2*
c^4*d - 8*C^2*a*b^3*c^3*d^2 + 4*C^2*a^2*b^2*c^4*d + 2*A*B*a*b^3*d^5 - 4*A*B
*a^3*b*d^5 + 4*A*C*a*b^3*c^5 - 2*A*B*a^4*c*d^4 - 2*A*B*b^4*c*d^4 + 2*B*C*a^
3*b*d^5 + 2*B*C*a^4*c*d^4 - 2*A*B*a*b^3*c^4*d - 4*A*C*a*b^3*c*d^4 + 8*A*C*a
^3*b*c*d^4 + 4*B*C*a*b^3*c^4*d - 2*B*C*a^3*b*c^4*d - 8*A*B*a*b^3*c^2*d^3 +
12*A*B*a^2*b^2*c*d^4 + 4*A*B*a^3*b*c^2*d^3 + 8*A*C*a*b^3*c^3*d^2 - 4*A*C*a^
2*b^2*c^4*d + 12*B*C*a*b^3*c^2*d^3 - 10*B*C*a^2*b^2*c*d^4 - 8*B*C*a^3*b*c^2
*d^3 - 16*A*C*a^2*b^2*c^2*d^3 + 4*B*C*a^2*b^2*c^3*d^2))/(d^2*(c^2 + d^2)^2)
)*(A*b^2 - A*a^2 - B*a^2*1i + B*b^2*1i + C*a^2 - C*b^2 - A*a*b*2i + 2*B*a*b
 + C*a*b*2i))/(2*f*(2*c*d - c^2*1i + d^2*1i)) + (log((2*C^2*b^4*c^5 - 2*C^2
*a^2*b^2*c^5 + 4*C^2*b^4*c^3*d^2 - A*B*a^4*d^5 - 2*A*C*b^4*c^5 + B*C*a^4*d^
5 + 2*A^2*a*b^3*d^5 - 2*A^2*a^3*b*d^5 - A^2*a^4*c*d^4 + 2*B^2*a^3*b*d^5 - A
^2*b^4*c*d^4 + B^2*a^4*c*d^4 + B^2*b^4*c*d^4 - C^2*a^4*c*d^4 + C^2*b^4*c*d^
4 - 4*C^2*a^2*b^2*c^3*d^2 + 5*A*B*a^2*b^2*d^5 + 2*A*C*a^2*b^2*c^5 + A*B*a^4
*c^2*d^3 + 3*A*B*b^4*c^2*d^3 - B*C*a^2*b^2*d^5 - 4*A*C*b^4*c^3*d^2 - B*C*a^
4*c^2*d^3 - 3*B*C*b^4*c^2*d^3 + 2*B^2*a*b^3*c^4*d - 2*C^2*a*b^3*c^4*d + 2*C
^2*a^3*b*c^4*d - 2*A^2*a*b^3*c^2*d^3 + 6*A^2*a^2*b^2*c*d^4 + 2*A^2*a^3*b*c^
```

```
2*d^3 + 6*B^2*a*b^3*c^2*d^3 - 6*B^2*a^2*b^2*c*d^4 - 2*B^2*a^3*b*c^2*d^3 - 6
*C^2*a*b^3*c^2*d^3 + 4*C^2*a^2*b^2*c*d^4 + 6*C^2*a^3*b*c^2*d^3 - 2*A*C*a*b^
3*d^5 + 2*A*C*a^3*b*d^5 - 4*B*C*a*b^3*c^5 + A*B*b^4*c^4*d + 2*A*C*a^4*c*d^4
 - B*C*b^4*c^4*d - 8*A*B*a*b^3*c*d^4 + 8*A*B*a^3*b*c*d^4 + 2*A*C*a*b^3*c^4*
d - 2*A*C*a^3*b*c^4*d + 4*B*C*a*b^3*c*d^4 - 8*B*C*a^3*b*c*d^4 - A*B*a^2*b^2
*c^4*d + 8*A*C*a*b^3*c^2*d^3 - 10*A*C*a^2*b^2*c*d^4 - 8*A*C*a^3*b*c^2*d^3 -
 8*B*C*a*b^3*c^3*d^2 + 5*B*C*a^2*b^2*c^4*d - 8*A*B*a^2*b^2*c^2*d^3 + 4*A*C*
a^2*b^2*c^3*d^2 + 16*B*C*a^2*b^2*c^2*d^3)/(d^2*(c^2 + d^2)^2) + (tan(e + f*
x)*(A^2*a^4*d^5 + A^2*b^4*d^5 + B^2*b^4*d^5 + C^2*a^4*d^5 + C^2*b^4*d^5 - 2
*A^2*a^2*b^2*d^5 + 3*B^2*a^2*b^2*d^5 + B^2*a^4*c^2*d^3 + 2*C^2*a^2*b^2*d^5
+ 3*B^2*b^4*c^2*d^3 - 2*A*C*a^4*d^5 - 2*A*C*b^4*d^5 - 2*B*C*b^4*c^5 - 4*C^2
*a*b^3*c^5 + B^2*b^4*c^4*d + 4*A^2*a^2*b^2*c^2*d^3 - 4*B^2*a^2*b^2*c^2*d^3
+ 12*C^2*a^2*b^2*c^2*d^3 + 2*B*C*a^2*b^2*c^5 - 4*B*C*b^4*c^3*d^2 + 4*A^2*a*
b^3*c*d^4 - 4*A^2*a^3*b*c*d^4 - 4*B^2*a*b^3*c*d^4 + 4*B^2*a^3*b*c*d^4 - 4*C
^2*a^3*b*c*d^4 - B^2*a^2*b^2*c^4*d - 8*C^2*a*b^3*c^3*d^2 + 4*C^2*a^2*b^2*c^
4*d + 2*A*B*a*b^3*d^5 - 4*A*B*a^3*b*d^5 + 4*A*C*a*b^3*c^5 - 2*A*B*a^4*c*d^4
 - 2*A*B*b^4*c*d^4 + 2*B*C*a^3*b*d^5 + 2*B*C*a^4*c*d^4 - 2*A*B*a*b^3*c^4*d
- 4*A*C*a*b^3*c*d^4 + 8*A*C*a^3*b*c*d^4 + 4*B*C*a*b^3*c^4*d - 2*B*C*a^3*b*c
^4*d - 8*A*B*a*b^3*c^2*d^3 + 12*A*B*a^2*b^2*c*d^4 + 4*A*B*a^3*b*c^2*d^3 + 8
*A*C*a*b^3*c^3*d^2 - 4*A*C*a^2*b^2*c^4*d + 12*B*C*a*b^3*c^2*d^3 - 10*B*C*a^
2*b^2*c*d^4 - 8*B*C*a^3*b*c^2*d^3 - 16*A*C*a^2*b^2*c^2*d^3 + 4*B*C*a^2*b^2*
c^3*d^2))/(d^2*(c^2 + d^2)^2) + ((a*1i + b)^2*((tan(e + f*x)*(3*B*a^2*d^5 -
 5*B*b^2*d^5 - 4*C*b^2*c^5 + 6*A*a*b*d^5 - 10*C*a*b*d^5 + 4*A*a^2*c*d^4 - 4
*A*b^2*c*d^4 + 2*B*b^2*c^4*d - 4*C*a^2*c*d^4 + 8*C*b^2*c*d^4 - B*a^2*c^2*d^
3 + B*b^2*c^2*d^3 - 8*B*a*b*c*d^4 + 4*C*a*b*c^4*d - 2*A*a*b*c^2*d^3 + 2*C*a
*b*c^2*d^3))/(d^2*(c^2 + d^2)) - (A*b^2*d^2 - A*a^2*d^2 + C*a^2*d^2 - 8*C*b
^2*c^2 - C*b^2*d^2 + 2*B*a*b*d^2 + 4*B*b^2*c*d + 8*C*a*b*c*d)/d + (d*(a*1i
+ b)^2*(4*c*d - c^2*tan(e + f*x) + 3*d^2*tan(e + f*x))*(A*1i + B - C*1i))/(
c*1i + d)^2)*(A*1i + B - C*1i))/(2*(c*1i + d)^2))*(A*b^2*1i - A*a^2*1i - B*
a^2 + B*b^2 + C*a^2*1i - C*b^2*1i - 2*A*a*b + B*a*b*2i + 2*C*a*b))/(2*f*(c*
d*2i - c^2 + d^2)) - (log(c + d*tan(e + f*x))*(d^3*(B*a^2*c^2 - 3*B*b^2*c^2
 + 2*A*a*b*c^2 - 6*C*a*b*c^2) - d^5*(B*a^2 + 2*A*a*b) - d*(B*b^2*c^4 + 2*C*
a*b*c^4) + d^4*(2*A*b^2*c - 2*A*a^2*c + 2*C*a^2*c + 4*B*a*b*c) + 2*C*b^2*c^
5 + 4*C*b^2*c^3*d^2))/(f*(d^7 + 2*c^2*d^5 + c^4*d^3)) + (C*b^2*tan(e + f*x)
)/(d^2*f) - (A*a^2*d^4 + C*b^2*c^4 - B*a^2*c*d^3 - B*b^2*c^3*d + A*b^2*c^2*
d^2 + C*a^2*c^2*d^2 - 2*A*a*b*c*d^3 - 2*C*a*b*c^3*d + 2*B*a*b*c^2*d^2)/(d*f
*(c*d^2 + d^3*tan(e + f*x))*(c^2 + d^2))
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**2,x)
```

```
[Out] Timed out
```

3.79 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$

Optimal. Leaf size=292

$$\frac{(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{d^2f\left(c^2+d^2\right)(c+d\tan(e+fx))}-\frac{\log(\cos(e+fx))\left(-A\left(2acd-b\left(c^2-d^2\right)\right)+a\left(Bc^2-Bd^2+2cCd\right)-b\left(-2Bcd\right.\right.}{f\left(c^2+d^2\right)^2}$$

```
[Out] -(a*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-b*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(c^2+d
^2)^2-(a*(B*c^2-B*d^2+2*C*c*d)-b*(-2*B*c*d+C*c^2-C*d^2)-A*(2*a*c*d-b*(c^2-d
^2)))*ln(cos(f*x+e))/(c^2+d^2)^2/f+(b*(c^4*C-c^2*(A-3*C)*d^2-2*B*c*d^3+A*d^
4)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*ln(c+d*tan(f*x+e))/d^2/(c^2+d^2)^2/f+(-
a*d+b*c)*(A*d^2-B*c*d+C*c^2)/d^2/(c^2+d^2)/f/(c+d*tan(f*x+e))
```

Rubi [A] time = 0.55, antiderivative size = 288, normalized size of antiderivative = 0.99, number of steps used = 5, number of rules used = 5, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3635, 3626, 3617, 31, 3475}

$$\frac{(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{d^2f\left(c^2+d^2\right)(c+d\tan(e+fx))}+\frac{\left(ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c^4C\right)\right)\text{lo}}{d^2f\left(c^2+d^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d*T
an[e + f*x])^2,x]
```

```
[Out] -(((a*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b*(2*c*(A - C)*d - B*(c^2
 - d^2)))*x)/(c^2 + d^2)^2) + ((2*a*A*c*d - 2*a*c*C*d - A*b*(c^2 - d^2) - a
*B*(c^2 - d^2) + b*(c^2*C - 2*B*c*d - C*d^2))*Log[Cos[e + f*x]])/((c^2 + d^
2)^2*f) + ((b*(c^4*C - c^2*(A - 3*C)*d^2 - 2*B*c*d^3 + A*d^4) + a*d^2*(2*c*
(A - C)*d - B*(c^2 - d^2)))*Log[c + d*Tan[e + f*x]])/(d^2*(c^2 + d^2)^2*f)
+ ((b*c - a*d)*(c^2*C - B*c*d + A*d^2))/(d^2*(c^2 + d^2)*f*(c + d*Tan[e + f
*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx = \frac{(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))} + \frac{\int \frac{ad(Ac-cC+}{}}{}$$

$$= -\frac{\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2c(\right.\right.}{\left(c^2+d^2\right)^2}$$

$$= -\frac{\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2c(\right.\right.}{\left(c^2+d^2\right)^2}$$

$$= -\frac{\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b\left(2c(\right.\right.}{\left(c^2+d^2\right)^2}$$

Mathematica [C] time = 6.78, size = 606, normalized size = 2.08

$$\frac{-2ic\tan^{-1}(\tan(e+fx))(c+d\tan(e+fx))\left(ad^2\left(2cd(A-C)+B\left(d^2-c^2\right)\right)+b\left(-c^2d^2(A-3C)+Ad^4-2Bcd^3+\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^2,x]
```

```
[Out] (c^2*(2*(c + I*d)^2*(a*(A - I*B - C)*d^2 + b*(I*c^2*C + 2*c*C*d + ((-I)*A -
 B)*d^2))*(e + f*x) - 2*b*C*(c^2 + d^2)^2*Log[Cos[e + f*x]] + (b*(c^4*C - c
^2*(A - 3*C)*d^2 - 2*B*c*d^3 + A*d^4) + a*d^2*(2*c*(A - C)*d + B*(-c^2 + d^
2)))*Log[(c*Cos[e + f*x] + d*Sin[e + f*x])^2]) + d*(2*(c + I*d)*(b*c*(I*c^3
*C*(I + e + f*x) + d^3*((-I)*B*(e + f*x) + A*(I + e + f*x)) - I*c*d^2*(-2*C
*(e + f*x) + A*(-I + e + f*x) - I*B*(I + e + f*x)) + c^2*d*(B + C*(I + e +
f*x))) + a*d*(c^3*C - I*A*d^3 + c*d^2*(A*(1 + I*e + I*f*x) - I*C*(e + f*x)
+ B*(I + e + f*x)) - c^2*d*(B*(1 + I*e + I*f*x) - A*(e + f*x) + C*(I + e +
f*x)))) - 2*b*c*C*(c^2 + d^2)^2*Log[Cos[e + f*x]] + c*(b*(c^4*C - c^2*(A -
3*C)*d^2 - 2*B*c*d^3 + A*d^4) + a*d^2*(2*c*(A - C)*d + B*(-c^2 + d^2)))*Log
[(c*Cos[e + f*x] + d*Sin[e + f*x])^2])*Tan[e + f*x] - (2*I)*c*(b*(c^4*C - c
^2*(A - 3*C)*d^2 - 2*B*c*d^3 + A*d^4) + a*d^2*(2*c*(A - C)*d + B*(-c^2 + d^
2)))*ArcTan[Tan[e + f*x]]*(c + d*Tan[e + f*x]))/(2*c*d^2*(c^2 + d^2)^2*f*(c
 + d*Tan[e + f*x]))
```

fricas [A] time = 0.65, size = 505, normalized size = 1.73

$$2\,Cbc^3d^2 - 2\,Aad^5 - 2\,(Ca + Bb)c^2d^3 + 2\,(Ba + Ab)cd^4 + 2\,\Big(((A - C)a - Bb)c^3d^2 + 2\,(Ba + (A - C)b)c^2d^3 - ((A$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^2,x, algorithm="fricas")
```

```
[Out] 1/2*(2*C*b*c^3*d^2 - 2*A*a*d^5 - 2*(C*a + B*b)*c^2*d^3 + 2*(B*a + A*b)*c*d^
4 + 2*(((A - C)*a - B*b)*c^3*d^2 + 2*(B*a + (A - C)*b)*c^2*d^3 - ((A - C)*a
 - B*b)*c*d^4)*f*x + (C*b*c^5 - (B*a + (A - 3*C)*b)*c^3*d^2 + 2*((A - C)*a
- B*b)*c^2*d^3 + (B*a + A*b)*c*d^4 + (C*b*c^4*d - (B*a + (A - 3*C)*b)*c^2*d
^3 + 2*((A - C)*a - B*b)*c*d^4 + (B*a + A*b)*d^5)*tan(f*x + e))*log((d^2*ta
n(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) - (C*b*c^5 +
 2*C*b*c^3*d^2 + C*b*c*d^4 + (C*b*c^4*d + 2*C*b*c^2*d^3 + C*b*d^5)*tan(f*x
+ e))*log(1/(tan(f*x + e)^2 + 1)) - 2*(C*b*c^4*d - A*a*c*d^4 - (C*a + B*b)*
c^3*d^2 + (B*a + A*b)*c^2*d^3 - (((A - C)*a - B*b)*c^2*d^3 + 2*(B*a + (A -
C)*b)*c*d^4 - ((A - C)*a - B*b)*d^5)*f*x)*tan(f*x + e))/((c^4*d^3 + 2*c^2*d
^5 + d^7)*f*tan(f*x + e) + (c^5*d^2 + 2*c^3*d^4 + c*d^6)*f)
```

giac [A] time = 6.68, size = 528, normalized size = 1.81

$$\frac{2\left(Aac^2 - Cac^2 - Bbc^2 + 2\,Bacd + 2\,Abcd - 2\,Cbcd - Aad^2 + Cad^2 + Bbd^2\right)(fx + e)}{c^4 + 2\,c^2d^2 + d^4} + \frac{\left(Bac^2 + Abc^2 - Cbc^2 - 2\,Aacd + 2\,Cacd + 2\,Bbcd - Bad^2 - Abd^2 + Cbd^2\right)\log\Big(\mathrm{ta}}{c^4 + 2\,c^2d^2 + d^4}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c^2 - C*a*c^2 - B*b*c^2 + 2*B*a*c*d + 2*A*b*c*d - 2*C*b*c*d - A
*a*d^2 + C*a*d^2 + B*b*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d^4) + (B*a*c^2 +
A*b*c^2 - C*b*c^2 - 2*A*a*c*d + 2*C*a*c*d + 2*B*b*c*d - B*a*d^2 - A*b*d^2 +
 C*b*d^2)*log(tan(f*x + e)^2 + 1)/(c^4 + 2*c^2*d^2 + d^4) + 2*(C*b*c^4 - B*
a*c^2*d^2 - A*b*c^2*d^2 + 3*C*b*c^2*d^2 + 2*A*a*c*d^3 - 2*C*a*c*d^3 - 2*B*b
*c*d^3 + B*a*d^4 + A*b*d^4)*log(abs(d*tan(f*x + e) + c))/(c^4*d^2 + 2*c^2*d
^4 + d^6) - 2*(C*b*c^4*tan(f*x + e) - B*a*c^2*d^2*tan(f*x + e) - A*b*c^2*d^
2*tan(f*x + e) + 3*C*b*c^2*d^2*tan(f*x + e) + 2*A*a*c*d^3*tan(f*x + e) - 2*
C*a*c*d^3*tan(f*x + e) - 2*B*b*c*d^3*tan(f*x + e) + B*a*d^4*tan(f*x + e) +
A*b*d^4*tan(f*x + e) + C*a*c^4 + B*b*c^4 - 2*B*a*c^3*d - 2*A*b*c^3*d + 2*C*
```

```
b*c^3*d + 3*A*a*c^2*d^2 - C*a*c^2*d^2 - B*b*c^2*d^2 + A*a*d^4)/((c^4*d + 2*
c^2*d^3 + d^5)*(d*tan(f*x + e) + c)))/f
```

maple [B] time = 0.27, size = 948, normalized size = 3.25

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x)
```

```
[Out] 1/f/(c^2+d^2)^2*C*arctan(tan(f*x+e))*a*d^2+3/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e
))*C*b*c^2+1/f/(c^2+d^2)/(c+d*tan(f*x+e))*B*a*c+1/f/(c^2+d^2)^2*ln(1+tan(f*
x+e)^2)*C*a*c*d+2/f/(c^2+d^2)^2*d*ln(c+d*tan(f*x+e))*A*a*c-2/f/(c^2+d^2)^2*
C*arctan(tan(f*x+e))*b*c*d+1/f/d^2/(c^2+d^2)/(c+d*tan(f*x+e))*C*c^3*b+2/f/(
c^2+d^2)^2*B*arctan(tan(f*x+e))*a*c*d-1/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*
a*c*d+1/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*b*c*d-2/f/(c^2+d^2)^2*d*ln(c+d*t
an(f*x+e))*B*b*c+2/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*b*c*d+1/f/(c^2+d^2)^2
/d^2*ln(c+d*tan(f*x+e))*C*b*c^4-2/f/(c^2+d^2)^2*d*ln(c+d*tan(f*x+e))*C*a*c-
1/f/d/(c^2+d^2)/(c+d*tan(f*x+e))*B*b*c^2-1/f/d/(c^2+d^2)/(c+d*tan(f*x+e))*C
*a*c^2-1/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a*d^2-1/f/(c^2+d^2)^2*ln(c+d*ta
n(f*x+e))*A*b*c^2-1/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*b*d^2-1/f*d/(c^2+d
^2)/(c+d*tan(f*x+e))*a*A+1/f/(c^2+d^2)^2*B*arctan(tan(f*x+e))*b*d^2+1/f/(c^
2+d^2)^2*d^2*ln(c+d*tan(f*x+e))*B*a+1/f/(c^2+d^2)/(c+d*tan(f*x+e))*A*b*c+1/
f/(c^2+d^2)^2*d^2*ln(c+d*tan(f*x+e))*A*b-1/f/(c^2+d^2)^2*C*arctan(tan(f*x+e
))*a*c^2-1/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*a*c^2+1/2/f/(c^2+d^2)^2*ln(1+
tan(f*x+e)^2)*A*b*c^2+1/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a*c^2-1/2/f/(c
^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a*d^2-1/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C
*b*c^2+1/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*b*d^2-1/f/(c^2+d^2)^2*B*arcta
n(tan(f*x+e))*b*c^2+1/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a*c^2
```

maxima [A] time = 0.50, size = 319, normalized size = 1.09

$$\frac{\frac{2\left(((A-C)a-Bb)c^2+2\,(Ba+(A-C)b)cd-((A-C)a-Bb)d^2\right)\left(fx+e\right)}{c^4+2\,c^2d^2+d^4}+\frac{2\left(Cbc^4-(Ba+(A-3\,C)b)c^2d^2+2\,((A-C)a-Bb)cd^3+(Ba+Ab)d^4\right)\log\left(d\tan\left(fx+e\right)+c\right)}{c^4d^2+2\,c^2d^4+d^6}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a - B*b)*c^2 + 2*(B*a + (A - C)*b)*c*d - ((A - C)*a - B*b)
*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d^4) + 2*(C*b*c^4 - (B*a + (A - 3*C)*b)*
c^2*d^2 + 2*((A - C)*a - B*b)*c*d^3 + (B*a + A*b)*d^4)*log(d*tan(f*x + e) +
 c)/(c^4*d^2 + 2*c^2*d^4 + d^6) + ((B*a + (A - C)*b)*c^2 - 2*((A - C)*a - B
```

```
*b)*c*d - (B*a + (A - C)*b)*d^2)*log(tan(f*x + e)^2 + 1)/(c^4 + 2*c^2*d^2 +
 d^4) + 2*(C*b*c^3 - A*a*d^3 - (C*a + B*b)*c^2*d + (B*a + A*b)*c*d^2)/(c^3*
d^2 + c*d^4 + (c^2*d^3 + d^5)*tan(f*x + e)))/f
```

mupad [B] time = 22.01, size = 1875, normalized size = 6.42

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x))^2,x)
```

```
[Out] (log(c + d*tan(e + f*x))*(d^4*(A*b + B*a) - d^3*(2*B*b*c - 2*A*a*c + 2*C*a*
c) - d^2*(A*b*c^2 + B*a*c^2 - 3*C*b*c^2) + C*b*c^4))/(f*(d^6 + 2*c^2*d^4 +
c^4*d^2)) - (log((A*B*b^2*d^4 - A*B*a^2*d^4 + B*C*a^2*d^4 + B*C*b^2*c^4 - A
^2*a*b*d^4 + B^2*a*b*d^4 + C^2*a*b*c^4 - A^2*a^2*c*d^3 + A^2*b^2*c*d^3 + B^
2*a^2*c*d^3 - B^2*b^2*c*d^3 - C^2*a^2*c*d^3 + C^2*b^2*c*d^3 + A*B*a^2*c^2*d
^2 - A*B*b^2*c^2*d^2 - B*C*a^2*c^2*d^2 + 3*B*C*b^2*c^2*d^2 + A^2*a*b*c^2*d^
2 - B^2*a*b*c^2*d^2 + 3*C^2*a*b*c^2*d^2 - A*C*a*b*c^4 + A*C*a*b*d^4 + 2*A*C
*a^2*c*d^3 - 2*A*C*b^2*c*d^3 - 4*A*C*a*b*c^2*d^2 + 4*A*B*a*b*c*d^3 - 4*B*C*
a*b*c*d^3)/(d*(c^2 + d^2)^2) + (tan(e + f*x)*(A^2*a^2*d^4 + B^2*b^2*d^4 + C
^2*a^2*d^4 + C^2*b^2*c^4 + C^2*b^2*d^4 + A^2*b^2*c^2*d^2 + B^2*a^2*c^2*d^2
+ 3*C^2*b^2*c^2*d^2 - 2*A*C*a^2*d^4 - A*C*b^2*c^4 - A*C*b^2*d^4 - 4*A*C*b^2
*c^2*d^2 - 2*A*B*a*b*d^4 - B*C*a*b*c^4 + B*C*a*b*d^4 - 2*A*B*a^2*c*d^3 + 2*
A*B*b^2*c*d^3 + 2*B*C*a^2*c*d^3 - 2*B*C*b^2*c*d^3 - 2*A^2*a*b*c*d^3 + 2*B^2
*a*b*c*d^3 - 2*C^2*a*b*c*d^3 + 2*A*B*a*b*c^2*d^2 - 4*B*C*a*b*c^2*d^2 + 4*A*
C*a*b*c*d^3))/(d*(c^2 + d^2)^2) + ((a*1i + b)*(B*1i - A + C)*(A*a*d - B*b*d
 - C*a*d - 4*C*b*c + (tan(e + f*x)*(3*A*b*d^4 + 3*B*a*d^4 + 2*C*b*c^4 - 5*C
*b*d^4 + 4*A*a*c*d^3 - 4*B*b*c*d^3 - 4*C*a*c*d^3 - A*b*c^2*d^2 - B*a*c^2*d^
2 + C*b*c^2*d^2))/(d*(c^2 + d^2)) + (d*(a*1i + b)*(4*c*d - c^2*tan(e + f*x)
 + 3*d^2*tan(e + f*x))*(B*1i - A + C))/(c*1i + d)^2))/(2*(c*1i + d)^2))*(A*
a*1i + A*b + B*a - B*b*1i - C*a*1i - C*b))/(2*f*(c*d*2i - c^2 + d^2)) - (lo
g((A*B*b^2*d^4 - A*B*a^2*d^4 + B*C*a^2*d^4 + B*C*b^2*c^4 - A^2*a*b*d^4 + B^
2*a*b*d^4 + C^2*a*b*c^4 - A^2*a^2*c*d^3 + A^2*b^2*c*d^3 + B^2*a^2*c*d^3 - B
^2*b^2*c*d^3 - C^2*a^2*c*d^3 + C^2*b^2*c*d^3 + A*B*a^2*c^2*d^2 - A*B*b^2*c^
2*d^2 - B*C*a^2*c^2*d^2 + 3*B*C*b^2*c^2*d^2 + A^2*a*b*c^2*d^2 - B^2*a*b*c^2
*d^2 + 3*C^2*a*b*c^2*d^2 - A*C*a*b*c^4 + A*C*a*b*d^4 + 2*A*C*a^2*c*d^3 - 2*
A*C*b^2*c*d^3 - 4*A*C*a*b*c^2*d^2 + 4*A*B*a*b*c*d^3 - 4*B*C*a*b*c*d^3)/(d*(
c^2 + d^2)^2) + (tan(e + f*x)*(A^2*a^2*d^4 + B^2*b^2*d^4 + C^2*a^2*d^4 + C^
2*b^2*c^4 + C^2*b^2*d^4 + A^2*b^2*c^2*d^2 + B^2*a^2*c^2*d^2 + 3*C^2*b^2*c^2
*d^2 - 2*A*C*a^2*d^4 - A*C*b^2*c^4 - A*C*b^2*d^4 - 4*A*C*b^2*c^2*d^2 - 2*A*
B*a*b*d^4 - B*C*a*b*c^4 + B*C*a*b*d^4 - 2*A*B*a^2*c*d^3 + 2*A*B*b^2*c*d^3 +
 2*B*C*a^2*c*d^3 - 2*B*C*b^2*c*d^3 - 2*A^2*a*b*c*d^3 + 2*B^2*a*b*c*d^3 - 2*
C^2*a*b*c*d^3 + 2*A*B*a*b*c^2*d^2 - 4*B*C*a*b*c^2*d^2 + 4*A*C*a*b*c*d^3))/(
d*(c^2 + d^2)^2) + ((a + b*1i)*(A + B*1i - C)*(A*a*d - B*b*d - C*a*d - 4*C*
```

```
b*c + (tan(e + f*x)*(3*A*b*d^4 + 3*B*a*d^4 + 2*C*b*c^4 - 5*C*b*d^4 + 4*A*a*
c*d^3 - 4*B*b*c*d^3 - 4*C*a*c*d^3 - A*b*c^2*d^2 - B*a*c^2*d^2 + C*b*c^2*d^2
))/(d*(c^2 + d^2)) + (d*(a + b*1i)*(4*c*d - c^2*tan(e + f*x) + 3*d^2*tan(e
+ f*x))*(A + B*1i - C)*1i)/(c*1i - d)^2)*1i)/(2*(c*1i - d)^2))*(A*a + A*b*1
i + B*a*1i - B*b - C*a - C*b*1i))/(2*f*(2*c*d - c^2*1i + d^2*1i)) - (A*a*d^
3 - C*b*c^3 - A*b*c*d^2 - B*a*c*d^2 + B*b*c^2*d + C*a*c^2*d)/(d^2*f*(c^2 +
d^2)*(c + d*tan(e + f*x)))
```

sympy [A] time = 4.10, size = 9721, normalized size = 33.29

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
)**2,x)
```

```
[Out] Piecewise((zoo*x*(a + b*tan(e))*(A + B*tan(e) + C*tan(e)**2)/tan(e)**2, Eq(
c, 0) & Eq(d, 0) & Eq(f, 0)), (A*a*f*x*tan(e + f*x)**2/(-4*d**2*f*tan(e + f
*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 2*I*A*a*f*x*tan(e + f*x)/(-4
*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - A*a*f*x/(-4
*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + A*a*tan(e +
 f*x)/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 2*
I*A*a/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - I*
A*b*f*x*tan(e + f*x)**2/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x
) + 4*d**2*f) - 2*A*b*f*x*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**
2*f*tan(e + f*x) + 4*d**2*f) + I*A*b*f*x/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d
**2*f*tan(e + f*x) + 4*d**2*f) - I*A*b*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)
**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - I*B*a*f*x*tan(e + f*x)**2/(-4*d
**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 2*B*a*f*x*tan
(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f)
+ I*B*a*f*x/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f
) - I*B*a*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x)
 + 4*d**2*f) - B*b*f*x*tan(e + f*x)**2/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**
2*f*tan(e + f*x) + 4*d**2*f) + 2*I*B*b*f*x*tan(e + f*x)/(-4*d**2*f*tan(e +
f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + B*b*f*x/(-4*d**2*f*tan(e +
f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + 3*B*b*tan(e + f*x)/(-4*d**2
*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 2*I*B*b/(-4*d**2
*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - C*a*f*x*tan(e +
f*x)**2/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) +
2*I*C*a*f*x*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*
x) + 4*d**2*f) + C*a*f*x/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*
x) + 4*d**2*f) + 3*C*a*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f
*tan(e + f*x) + 4*d**2*f) - 2*I*C*a/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f
*tan(e + f*x) + 4*d**2*f) - 3*I*C*b*f*x*tan(e + f*x)**2/(-4*d**2*f*tan(e +
f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 6*C*b*f*x*tan(e + f*x)/(-4*
```

```
d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + 3*I*C*b*f*x/
(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 2*C*b*lo
g(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**
2*f*tan(e + f*x) + 4*d**2*f) + 4*I*C*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x
)/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + 2*C*b*
log(tan(e + f*x)**2 + 1)/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*
x) + 4*d**2*f) + 5*I*C*b*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2
*f*tan(e + f*x) + 4*d**2*f) + 4*C*b/(-4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f
*tan(e + f*x) + 4*d**2*f), Eq(c, -I*d)), (A*a*f*x*tan(e + f*x)**2/(-4*d**2*
f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + 2*I*A*a*f*x*tan(e
 + f*x)/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) -
A*a*f*x/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) +
A*a*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d
**2*f) + 2*I*A*a/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d
**2*f) + I*A*b*f*x*tan(e + f*x)**2/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*
tan(e + f*x) + 4*d**2*f) - 2*A*b*f*x*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**
2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - I*A*b*f*x/(-4*d**2*f*tan(e + f*x)
**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + I*A*b*tan(e + f*x)/(-4*d**2*f*t
an(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + I*B*a*f*x*tan(e + f*
x)**2/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 2*
B*a*f*x*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) +
 4*d**2*f) - I*B*a*f*x/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x)
 + 4*d**2*f) + I*B*a*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*t
an(e + f*x) + 4*d**2*f) - B*b*f*x*tan(e + f*x)**2/(-4*d**2*f*tan(e + f*x)**
2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 2*I*B*b*f*x*tan(e + f*x)/(-4*d**2
*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + B*b*f*x/(-4*d**2
*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + 3*B*b*tan(e + f*
x)/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + 2*I*B
*b/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - C*a*f
*x*tan(e + f*x)**2/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4
*d**2*f) - 2*I*C*a*f*x*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f
*tan(e + f*x) + 4*d**2*f) + C*a*f*x/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f
*tan(e + f*x) + 4*d**2*f) + 3*C*a*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 -
 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + 2*I*C*a/(-4*d**2*f*tan(e + f*x)**2 -
 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + 3*I*C*b*f*x*tan(e + f*x)**2/(-4*d**2
*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 6*C*b*f*x*tan(e
+ f*x)/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 3
*I*C*b*f*x/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f)
 - 2*C*b*log(tan(e + f*x)**2 + 1)*tan(e + f*x)**2/(-4*d**2*f*tan(e + f*x)**
2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) - 4*I*C*b*log(tan(e + f*x)**2 + 1)*
tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*
f) + 2*C*b*log(tan(e + f*x)**2 + 1)/(-4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f
*tan(e + f*x) + 4*d**2*f) - 5*I*C*b*tan(e + f*x)/(-4*d**2*f*tan(e + f*x)**2
 - 8*I*d**2*f*tan(e + f*x) + 4*d**2*f) + 4*C*b/(-4*d**2*f*tan(e + f*x)**2 -
 8*I*d**2*f*tan(e + f*x) + 4*d**2*f), Eq(c, I*d)), ((A*a*x + A*b*log(tan(e
```

```
+ f*x)**2 + 1)/(2*f) + B*a*log(tan(e + f*x)**2 + 1)/(2*f) - B*b*x + B*b*tan
(e + f*x)/f - C*a*x + C*a*tan(e + f*x)/f - C*b*log(tan(e + f*x)**2 + 1)/(2*
f) + C*b*tan(e + f*x)**2/(2*f))/c**2, Eq(d, 0)), (x*(a + b*tan(e))*(A + B*t
an(e) + C*tan(e)**2)/(c + d*tan(e))**2, Eq(f, 0)), (2*A*a*c**3*d**2*f*x/(2*
c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*ta
n(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*A*a*c**2*d**3*f*x*tan(
e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c*
*2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 4*A*a*c**2*d
**3*log(c/d + tan(e + f*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4
*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f
*x)) - 2*A*a*c**2*d**3*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2*f + 2*c**4*d**
3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f
+ 2*d**7*f*tan(e + f*x)) - 2*A*a*c**2*d**3/(2*c**5*d**2*f + 2*c**4*d**3*f*t
an(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d
**7*f*tan(e + f*x)) - 2*A*a*c*d**4*f*x/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e
 + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*
f*tan(e + f*x)) + 4*A*a*c*d**4*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5
*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e
+ f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*A*a*c*d**4*log(tan(e + f*x
)**2 + 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3
*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x))
- 2*A*a*d**5*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) +
 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e +
 f*x)) - 2*A*a*d**5/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d*
*4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2
*A*b*c**3*d**2*log(c/d + tan(e + f*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e
 + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*
f*tan(e + f*x)) + A*b*c**3*d**2*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2*f + 2
*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*
c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*A*b*c**3*d**2/(2*c**5*d**2*f + 2*c**4
*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**
6*f + 2*d**7*f*tan(e + f*x)) + 4*A*b*c**2*d**3*f*x/(2*c**5*d**2*f + 2*c**4*
d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6
*f + 2*d**7*f*tan(e + f*x)) - 2*A*b*c**2*d**3*log(c/d + tan(e + f*x))*tan(e
 + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**
2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + A*b*c**2*d**3
*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e
 + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*
f*tan(e + f*x)) + 4*A*b*c*d**4*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**
3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f
+ 2*d**7*f*tan(e + f*x)) + 2*A*b*c*d**4*log(c/d + tan(e + f*x))/(2*c**5*d**
2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*
x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - A*b*c*d**4*log(tan(e + f*x)**2 +
 1)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d*
*5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*A*b*c*d**4/(2*c
```

```
**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan
(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*A*b*d**5*log(c/d + tan(
e + f*x))*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3
*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x))
- A*b*d**5*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d*
*3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f
 + 2*d**7*f*tan(e + f*x)) - 2*B*a*c**3*d**2*log(c/d + tan(e + f*x))/(2*c**5
*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e
+ f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + B*a*c**3*d**2*log(tan(e + f*
x)**2 + 1)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*
c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*B*a*c**3
*d**2/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*
d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 4*B*a*c**2*d**3
*f*x/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d
**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*B*a*c**2*d**3*
log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e +
 f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*
tan(e + f*x)) + B*a*c**2*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5
*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e
+ f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 4*B*a*c*d**4*f*x*tan(e + f*x
)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5
*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*B*a*c*d**4*log(c/
d + tan(e + f*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4
*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - B*a
*c*d**4*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x
) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(
e + f*x)) + 2*B*a*c*d**4/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c*
*3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)
) + 2*B*a*d**5*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4
*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**
6*f + 2*d**7*f*tan(e + f*x)) - B*a*d**5*log(tan(e + f*x)**2 + 1)*tan(e + f*
x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**
5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*B*b*c**4*d/(2*c*
*5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(
e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*B*b*c**3*d**2*f*x/(2*c**
5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e
 + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*B*b*c**2*d**3*f*x*tan(e +
 f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*
d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 4*B*b*c**2*d**3
*log(c/d + tan(e + f*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c*
*3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)
) + 2*B*b*c**2*d**3*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2*f + 2*c**4*d**3*f
*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2
*d**7*f*tan(e + f*x)) - 2*B*b*c**2*d**3/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(
e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7
```

```
*f*tan(e + f*x)) + 2*B*b*c*d**4*f*x/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e +
f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*t
an(e + f*x)) - 4*B*b*c*d**4*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d*
*2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f
*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*B*b*c*d**4*log(tan(e + f*x)**
2 + 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d*
*4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2
*B*b*d**5*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*
c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*
x)) - 2*C*a*c**4*d/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**
4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*
C*a*c**3*d**2*f*x/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4
*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*C
*a*c**2*d**3*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) +
 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e +
 f*x)) - 4*C*a*c**2*d**3*log(c/d + tan(e + f*x))/(2*c**5*d**2*f + 2*c**4*d*
*3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f
 + 2*d**7*f*tan(e + f*x)) + 2*C*a*c**2*d**3*log(tan(e + f*x)**2 + 1)/(2*c**
5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e
 + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 2*C*a*c**2*d**3/(2*c**5*d**
2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*
x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*C*a*c*d**4*f*x/(2*c**5*d**2*f
+ 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) +
 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - 4*C*a*c*d**4*log(c/d + tan(e + f*x))
*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f +
 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*C*a*c
*d**4*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*
tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*
d**7*f*tan(e + f*x)) + 2*C*a*d**5*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*
d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6
*f + 2*d**7*f*tan(e + f*x)) + 2*C*b*c**5*log(c/d + tan(e + f*x))/(2*c**5*d*
*2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f
*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 2*C*b*c**5/(2*c**5*d**2*f + 2*c
**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*
d**6*f + 2*d**7*f*tan(e + f*x)) + 2*C*b*c**4*d*log(c/d + tan(e + f*x))*tan(
e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c*
*2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) + 6*C*b*c**3*d
**2*log(c/d + tan(e + f*x))/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4
*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f
*x)) - C*b*c**3*d**2*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2*f + 2*c**4*d**3*
f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f +
2*d**7*f*tan(e + f*x)) + 2*C*b*c**3*d**2/(2*c**5*d**2*f + 2*c**4*d**3*f*tan
(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**
7*f*tan(e + f*x)) - 4*C*b*c**2*d**3*f*x/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(
e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7
```

```
*f*tan(e + f*x)) + 6*C*b*c**2*d**3*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*
c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*ta
n(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)) - C*b*c**2*d**3*log(tan(e
+ f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4
*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f
*x)) - 4*C*b*c*d**4*f*x*tan(e + f*x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e +
 f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*
tan(e + f*x)) + C*b*c*d**4*log(tan(e + f*x)**2 + 1)/(2*c**5*d**2*f + 2*c**4
*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**5*f*tan(e + f*x) + 2*c*d**
6*f + 2*d**7*f*tan(e + f*x)) + C*b*d**5*log(tan(e + f*x)**2 + 1)*tan(e + f*
x)/(2*c**5*d**2*f + 2*c**4*d**3*f*tan(e + f*x) + 4*c**3*d**4*f + 4*c**2*d**
5*f*tan(e + f*x) + 2*c*d**6*f + 2*d**7*f*tan(e + f*x)), True))
```

3.80 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx$

Optimal. Leaf size=140

$$-\frac{Ad^2-Bcd+c^2C}{df\left(c^2+d^2\right)(c+d\tan(e+fx))}+\frac{\left(2cd(A-C)-B\left(c^2-d^2\right)\right)\log(c\cos(e+fx)+d\sin(e+fx))}{f\left(c^2+d^2\right)^2}-\frac{x\left(-A\left(c^2-d^2\right.\right.}{(}$$

```
[Out] -(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))*x/(c^2+d^2)^2+(2*c*(A-C)*d-B*(c^2-d^2))*
ln(c*cos(f*x+e)+d*sin(f*x+e))/(c^2+d^2)^2/f+(-A*d^2+B*c*d-C*c^2)/d/(c^2+d^2
)/f/(c+d*tan(f*x+e))
```

Rubi [A] time = 0.21, antiderivative size = 140, normalized size of antiderivative = 1.00, number of steps used = 3, number of rules used = 3, integrand size = 33, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.091, Rules used = {3628, 3531, 3530}

$$-\frac{Ad^2-Bcd+c^2C}{df\left(c^2+d^2\right)(c+d\tan(e+fx))}+\frac{\left(2cd(A-C)-B\left(c^2-d^2\right)\right)\log(c\cos(e+fx)+d\sin(e+fx))}{f\left(c^2+d^2\right)^2}-\frac{x\left(-A\left(c^2-d^2\right.\right.}{(}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^2,x]
```

```
[Out] -(((c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2))*x)/(c^2 + d^2)^2) + ((2*c*(A -
 C)*d - B*(c^2 - d^2))*Log[c*Cos[e + f*x] + d*Sin[e + f*x]])/((c^2 + d^2)^2
*f) - (c^2*C - B*c*d + A*d^2)/(d*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^2}\,dx = -\frac{c^2C-Bcd+Ad^2}{d\left(c^2+d^2\right)f(c+d\tan(e+fx))}+\frac{\int \frac{Ac-cC+Bd+(Bc-(A-C)d)\tan(e+fx)}{c+d\tan(e+fx)}}{c^2+d^2}$$

$$= -\frac{\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)x}{\left(c^2+d^2\right)^2}-\frac{c^2C-Bcd+Ad^2}{d\left(c^2+d^2\right)f(c+d\tan(e+$$

$$= -\frac{\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)x}{\left(c^2+d^2\right)^2}+\frac{\left(2c(A-C)d-B\left(c^2-d^2\right)\right.}{\left(\right.}$$

Mathematica [C] time = 2.55, size = 207, normalized size = 1.48

$$\frac{(d(C-A)+Bc)\left(\frac{2d\left(\frac{c^2+d^2}{c+d\tan(e+fx)}-2c\log(c+d\tan(e+fx))\right)}{\left(c^2+d^2\right)^2}+\frac{i\log(-\tan(e+fx)+i)}{(c+id)^2}-\frac{i\log(\tan(e+fx)+i)}{(c-id)^2}\right)+\frac{B((-d-ic)\log(-\tan(e+fx)+i)+}{}}{2df}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^2,x]
```

```
[Out] ((B*(((-I)*c - d)*Log[I - Tan[e + f*x]] + I*(c + I*d)*Log[I + Tan[e + f*x]]
 + 2*d*Log[c + d*Tan[e + f*x]]))/(c^2 + d^2) - (2*C)/(c + d*Tan[e + f*x]) +
 (B*c + (-A + C)*d)*((I*Log[I - Tan[e + f*x]])/(c + I*d)^2 - (I*Log[I + Tan
[e + f*x]])/(c - I*d)^2 + (2*d*(-2*c*Log[c + d*Tan[e + f*x]] + (c^2 + d^2)/
(c + d*Tan[e + f*x])))/(c^2 + d^2)^2))/(2*d*f)
```

fricas [A] time = 0.58, size = 256, normalized size = 1.83

$$-\frac{2Cc^2d-2Bcd^2+2Ad^3-2\left((A-C)c^3+2Bc^2d-(A-C)cd^2\right)fx+\left(Bc^3-2(A-C)c^2d-Bcd^2+\left(Bc^2d-2\right.\right.}{2\left(\left(c^4d+2c^2\right.\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x, algorithm="
fricas")
```

```
[Out] -1/2*(2*C*c^2*d - 2*B*c*d^2 + 2*A*d^3 - 2*((A - C)*c^3 + 2*B*c^2*d - (A - C
)*c*d^2)*f*x + (B*c^3 - 2*(A - C)*c^2*d - B*c*d^2 + (B*c^2*d - 2*(A - C)*c*
d^2 - B*d^3)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c
^2)/(tan(f*x + e)^2 + 1)) - 2*(C*c^3 - B*c^2*d + A*c*d^2 + ((A - C)*c^2*d +
 2*B*c*d^2 - (A - C)*d^3)*f*x)*tan(f*x + e))/((c^4*d + 2*c^2*d^3 + d^5)*f*t
an(f*x + e) + (c^5 + 2*c^3*d^2 + c*d^4)*f)
```

giac [B] time = 3.99, size = 299, normalized size = 2.14

$$\frac{\frac{2\left(Ac^2-Cc^2+2\,Bcd-Ad^2+Cd^2\right)(fx+e)}{c^4+2\,c^2d^2+d^4}+\frac{\left(Bc^2-2\,Acd+2\,Ccd-Bd^2\right)\log\left(\tan\left(fx+e\right)^2+1\right)}{c^4+2\,c^2d^2+d^4}-\frac{2\left(Bc^2d-2\,Acd^2+2\,Ccd^2-Bd^3\right)\log\left(\left|d\tan\left(fx+e\right)+c\right|\right)}{c^4d+2\,c^2d^3+d^5}+\frac{2}{}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x, algorithm="
giac")
```

```
[Out] 1/2*(2*(A*c^2 - C*c^2 + 2*B*c*d - A*d^2 + C*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2
 + d^4) + (B*c^2 - 2*A*c*d + 2*C*c*d - B*d^2)*log(tan(f*x + e)^2 + 1)/(c^4
+ 2*c^2*d^2 + d^4) - 2*(B*c^2*d - 2*A*c*d^2 + 2*C*c*d^2 - B*d^3)*log(abs(d*
tan(f*x + e) + c))/(c^4*d + 2*c^2*d^3 + d^5) + 2*(B*c^2*d^2*tan(f*x + e) -
2*A*c*d^3*tan(f*x + e) + 2*C*c*d^3*tan(f*x + e) - B*d^4*tan(f*x + e) - C*c^
4 + 2*B*c^3*d - 3*A*c^2*d^2 + C*c^2*d^2 - A*d^4)/((c^4*d + 2*c^2*d^3 + d^5)
*(d*tan(f*x + e) + c)))/f
```

maple [B] time = 0.30, size = 438, normalized size = 3.13

$$-\frac{dA}{f\left(c^2+d^2\right)\left(c+d\tan\left(fx+e\right)\right)}+\frac{Bc}{f\left(c^2+d^2\right)\left(c+d\tan\left(fx+e\right)\right)}-\frac{c^2C}{f\left(c^2+d^2\right)d\left(c+d\tan\left(fx+e\right)\right)}+\frac{2\ln\left(c+d\right.}{f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x)
```

```
[Out] -1/f/(c^2+d^2)*d/(c+d*tan(f*x+e))*A+1/f/(c^2+d^2)/(c+d*tan(f*x+e))*B*c-1/f/
(c^2+d^2)/d/(c+d*tan(f*x+e))*c^2*C+2/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*c*d
-1/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*c^2+1/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e)
)*B*d^2-2/f/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*c*C*d-1/f/(c^2+d^2)^2*ln(1+tan(f
*x+e)^2)*A*c*d+1/2/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*c^2-1/2/f/(c^2+d^2)^2
```

```
*ln(1+tan(f*x+e)^2)*B*d^2+1/f/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*c*C*d+1/f/(c^2
+d^2)^2*A*arctan(tan(f*x+e))*c^2-1/f/(c^2+d^2)^2*A*arctan(tan(f*x+e))*d^2+2
/f/(c^2+d^2)^2*B*arctan(tan(f*x+e))*c*d-1/f/(c^2+d^2)^2*C*arctan(tan(f*x+e)
)*c^2+1/f/(c^2+d^2)^2*C*arctan(tan(f*x+e))*d^2
```

maxima [A] time = 0.67, size = 205, normalized size = 1.46

$$\frac{\frac{2\left((A-C)c^2+2\,Bcd-(A-C)d^2\right)(fx+e)}{c^4+2\,c^2d^2+d^4}-\frac{2\left(Bc^2-2\,(A-C)cd-Bd^2\right)\log\left(d\tan\left(fx+e\right)+c\right)}{c^4+2\,c^2d^2+d^4}+\frac{\left(Bc^2-2\,(A-C)cd-Bd^2\right)\log\left(\tan\left(fx+e\right)^2+1\right)}{c^4+2\,c^2d^2+d^4}-\frac{2\,(C}{c^3d+cd^3+(}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x, algorithm="
maxima")
```

```
[Out] 1/2*(2*((A - C)*c^2 + 2*B*c*d - (A - C)*d^2)*(f*x + e)/(c^4 + 2*c^2*d^2 + d
^4) - 2*(B*c^2 - 2*(A - C)*c*d - B*d^2)*log(d*tan(f*x + e) + c)/(c^4 + 2*c^
2*d^2 + d^4) + (B*c^2 - 2*(A - C)*c*d - B*d^2)*log(tan(f*x + e)^2 + 1)/(c^4
 + 2*c^2*d^2 + d^4) - 2*(C*c^2 - B*c*d + A*d^2)/(c^3*d + c*d^3 + (c^2*d^2 +
 d^4)*tan(f*x + e)))/f
```

mupad [B] time = 11.35, size = 184, normalized size = 1.31

$$\frac{\ln\left(c+d\tan\left(e+f\,x\right)\right)\,\left(-B\,c^2+\left(2\,A-2\,C\right)\,c\,d+B\,d^2\right)}{f\,\left(c^4+2\,c^2\,d^2+d^4\right)}-\frac{\ln\left(\tan\left(e+f\,x\right)-\mathrm{i}\right)\,\left(A-C+B\,1\mathrm{i}\right)}{2\,f\,\left(-c^2\,1\mathrm{i}+2\,c\,d+d^2\,1\mathrm{i}\right)}-\frac{\ln\left(\tan\left(e+f\,x\right.}{2\,f\,\left(-c\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x))^2,x)
```

```
[Out] (log(c + d*tan(e + f*x))*(B*d^2 - B*c^2 + c*d*(2*A - 2*C)))/(f*(c^4 + d^4 +
 2*c^2*d^2)) - (log(tan(e + f*x) - 1i)*(A + B*1i - C))/(2*f*(2*c*d - c^2*1i
 + d^2*1i)) - (log(tan(e + f*x) + 1i)*(A*1i + B - C*1i))/(2*f*(c*d*2i - c^2
 + d^2)) - (A*d^2 + C*c^2 - B*c*d)/(d*f*(c^2 + d^2)*(c + d*tan(e + f*x)))
```

sympy [A] time = 2.13, size = 4396, normalized size = 31.40

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**2,x)
```

```
[Out] Piecewise((zoo*x*(A + B*tan(e) + C*tan(e)**2)/tan(e)**2, Eq(c, 0) & Eq(d, 0
) & Eq(f, 0)), (-A*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2
```

```
*f*tan(e + f*x) - 4*d**2*f) + 2*I*A*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)
**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + A*f*x/(4*d**2*f*tan(e + f*x)**2
 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - A*tan(e + f*x)/(4*d**2*f*tan(e + f
*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*A/(4*d**2*f*tan(e + f*x)
**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*f*x*tan(e + f*x)**2/(4*d**2
*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*B*f*x*tan(e +
f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - I*B*
f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + I*B*t
an(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x) - 4*d**2*f)
 + C*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*tan(e + f*x
) - 4*d**2*f) - 2*I*C*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2
*f*tan(e + f*x) - 4*d**2*f) - C*f*x/(4*d**2*f*tan(e + f*x)**2 - 8*I*d**2*f*
tan(e + f*x) - 4*d**2*f) - 3*C*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 - 8*I
*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*C/(4*d**2*f*tan(e + f*x)**2 - 8*I*d*
*2*f*tan(e + f*x) - 4*d**2*f), Eq(c, -I*d)), (-A*f*x*tan(e + f*x)**2/(4*d**
2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*A*f*x*tan(e
 + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + A
*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - A*ta
n(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f)
- 2*I*A/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - I
*B*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x)
- 4*d**2*f) + 2*B*f*x*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*t
an(e + f*x) - 4*d**2*f) + I*B*f*x/(4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*ta
n(e + f*x) - 4*d**2*f) - I*B*tan(e + f*x)/(4*d**2*f*tan(e + f*x)**2 + 8*I*d
**2*f*tan(e + f*x) - 4*d**2*f) + C*f*x*tan(e + f*x)**2/(4*d**2*f*tan(e + f*
x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) + 2*I*C*f*x*tan(e + f*x)/(4*d**
2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - C*f*x/(4*d**2*f
*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 3*C*tan(e + f*x)/(
4*d**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f) - 2*I*C/(4*d
**2*f*tan(e + f*x)**2 + 8*I*d**2*f*tan(e + f*x) - 4*d**2*f), Eq(c, I*d)), (
(A*x + B*log(tan(e + f*x)**2 + 1)/(2*f) - C*x + C*tan(e + f*x)/f)/c**2, Eq(
d, 0)), (x*(A + B*tan(e) + C*tan(e)**2)/(c + d*tan(e))**2, Eq(f, 0)), (2*A*
c**3*d*f*x/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**
2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 2*A*c**2*d**2
*f*x*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f
+ 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 4*A*c*
*2*d**2*log(c/d + tan(e + f*x))/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) +
4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e +
f*x)) - 2*A*c**2*d**2*log(tan(e + f*x)**2 + 1)/(2*c**5*d*f + 2*c**4*d**2*f*
tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*
d**6*f*tan(e + f*x)) - 2*A*c**2*d**2/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*
x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan
(e + f*x)) - 2*A*c*d**3*f*x/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c*
*3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)
) + 4*A*c*d**3*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d*
```

```
*2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f
 + 2*d**6*f*tan(e + f*x)) - 2*A*c*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x
)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*
tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*A*d**4*f*x*tan(e + f
*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*
f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*A*d**4/(2*c**5*d*f
 + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x)
+ 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*B*c**3*d*log(c/d + tan(e + f*x))/
(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*ta
n(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + B*c**3*d*log(tan(e + f*x
)**2 + 1)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2
*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 2*B*c**3*d/(2*
c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e
 + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 4*B*c**2*d**2*f*x/(2*c**5*d
*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x
) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*B*c**2*d**2*log(c/d + tan(e + f
*x))*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f
+ 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + B*c**2
*d**2*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan
(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**
6*f*tan(e + f*x)) + 4*B*c*d**3*f*x*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f
*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2
*d**6*f*tan(e + f*x)) + 2*B*c*d**3*log(c/d + tan(e + f*x))/(2*c**5*d*f + 2*
c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c
*d**5*f + 2*d**6*f*tan(e + f*x)) - B*c*d**3*log(tan(e + f*x)**2 + 1)/(2*c**
5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e +
f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 2*B*c*d**3/(2*c**5*d*f + 2*c**
4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d*
*5*f + 2*d**6*f*tan(e + f*x)) + 2*B*d**4*log(c/d + tan(e + f*x))*tan(e + f*
x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f
*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - B*d**4*log(tan(e + f*
x)**2 + 1)*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d
**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) -
2*C*c**4/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*
d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*C*c**3*d*f*x/
(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*ta
n(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 2*C*c**2*d**2*f*x*tan(e
+ f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d*
*4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 4*C*c**2*d**2*log
(c/d + tan(e + f*x))/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3
*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 2*C
*c**2*d**2*log(tan(e + f*x)**2 + 1)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x
) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(
e + f*x)) - 2*C*c**2*d**2/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3
*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x))
```

```
+ 2*C*c*d**3*f*x/(2*c**5*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f +
 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) - 4*C*c*d
**3*log(c/d + tan(e + f*x))*tan(e + f*x)/(2*c**5*d*f + 2*c**4*d**2*f*tan(e
+ f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x) + 2*c*d**5*f + 2*d**6*f
*tan(e + f*x)) + 2*C*c*d**3*log(tan(e + f*x)**2 + 1)*tan(e + f*x)/(2*c**5*d
*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f*x
) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)) + 2*C*d**4*f*x*tan(e + f*x)/(2*c**5
*d*f + 2*c**4*d**2*f*tan(e + f*x) + 4*c**3*d**3*f + 4*c**2*d**4*f*tan(e + f
*x) + 2*c*d**5*f + 2*d**6*f*tan(e + f*x)), True))
```

3.81 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2}\,dx$

Optimal. Leaf size=293

$$-\frac{x\left(a\left(-A\left(c^2-d^2\right)-2Bcd+c^2C-Cd^2\right)+b\left(2cd(A-C)-B\left(c^2-d^2\right)\right)\right)}{\left(a^2+b^2\right)\left(c^2+d^2\right)^2}+\frac{b\left(Ab^2-a(bB-aC)\right)\log(a\cos(e+}{f\left(a^2+b^2\right)(bc-a}$$

```
[Out] -(a*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))+b*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(a^2+b
^2)/(c^2+d^2)^2+b*(A*b^2-a*(B*b-C*a))*ln(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^
2)/(-a*d+b*c)^2/f-(b*(c^4*C-2*B*c^3*d+c^2*(3*A-C)*d^2+A*d^4)-a*d^2*(2*c*(A-
C)*d-B*(c^2-d^2)))*ln(c*cos(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)^2/(c^2+d^2)^2/f
+(A*d^2-B*c*d+C*c^2)/(-a*d+b*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))
```

Rubi [**A**] time = 0.81, antiderivative size = 293, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 3, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3649, 3651, 3530}

$$-\frac{x\left(a\left(-A\left(c^2-d^2\right)-2Bcd+c^2C-Cd^2\right)+b\left(2cd(A-C)-B\left(c^2-d^2\right)\right)\right)}{\left(a^2+b^2\right)\left(c^2+d^2\right)^2}+\frac{b\left(Ab^2-a(bB-aC)\right)\log(a\cos(e+}{f\left(a^2+b^2\right)(bc-a}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c + d*Ta
n[e + f*x])^2),x]
```

```
[Out] -(((a*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + b*(2*c*(A - C)*d - B*(c^2
 - d^2)))*x)/((a^2 + b^2)*(c^2 + d^2)^2)) + (b*(A*b^2 - a*(b*B - a*C))*Log[
a*Cos[e + f*x] + b*Sin[e + f*x]])/((a^2 + b^2)*(b*c - a*d)^2*f) - ((b*(c^4*
C - 2*B*c^3*d + c^2*(3*A - C)*d^2 + A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2
- d^2)))*Log[c*Cos[e + f*x] + d*Sin[e + f*x]])/((b*c - a*d)^2*(c^2 + d^2)^2
*f) + (c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x
]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^2}\,dx = \frac{c^2C-Bcd+Ad^2}{(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))} + \frac{\int \frac{-aAcd+ad(cC-Bd)+Ab\left(c^2\right.}{}}{}$$

$$= -\frac{\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d-B\left(c^2-\right.\right.\right.}{\left(a^2+b^2\right)\left(c^2+d^2\right)^2}$$

$$= -\frac{\left(a\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)+b\left(2c(A-C)d-B\left(c^2-\right.\right.\right.}{\left(a^2+b^2\right)\left(c^2+d^2\right)^2}$$

Mathematica [B] time = 7.50, size = 592, normalized size = 2.02

$$-\frac{\frac{b^2\left(c^2+d^2\right)\left(Ab^2-a(bB-aC)\right)\log(a+b\tan(e+fx))}{\left(a^2+b^2\right)(bc-ad)}-\frac{b(bc-ad)\log\left(\sqrt{-b^2}-b\tan(e+fx)\right)\left(-\frac{\sqrt{-b^2}\left(a\left(-A\left(c^2-d^2\right)-2Bcd+c^2C-Cd^2\right)+b\left(2cd(A-C)-B\left(c^2-d^2\right)\right)\right)}{b}+2a\right.}{2\left(a^2+b^2\right)\left(c^2+d^2\right)}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c
+ d*Tan[e + f*x])^2),x]
```

```
[Out] -((-1/2*(b*(b*c - a*d)*(A*b*c^2 - a*B*c^2 - b*c^2*C + 2*a*A*c*d + 2*b*B*c*d
 - 2*a*c*C*d - A*b*d^2 + a*B*d^2 + b*C*d^2 - (Sqrt[-b^2]*(a*(c^2*C - 2*B*c*
d - C*d^2 - A*(c^2 - d^2)) + b*(2*c*(A - C)*d - B*(c^2 - d^2))))/b)*Log[Sqr
t[-b^2] - b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2)) + (b^2*(A*b^2 - a*(b*B
 - a*C))*(c^2 + d^2)*Log[a + b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)) - (
b*(b*c - a*d)*(A*b*c^2 - a*B*c^2 - b*c^2*C + 2*a*A*c*d + 2*b*B*c*d - 2*a*c*
C*d - A*b*d^2 + a*B*d^2 + b*C*d^2 + (Sqrt[-b^2]*(a*(c^2*C - 2*B*c*d - C*d^2
 - A*(c^2 - d^2)) + b*(2*c*(A - C)*d - B*(c^2 - d^2))))/b)*Log[Sqrt[-b^2] +
 b*Tan[e + f*x]])/(2*(a^2 + b^2)*(c^2 + d^2)) - (b*(b*(c^4*C - 2*B*c^3*d +
c^2*(3*A - C)*d^2 + A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Log[c +
 d*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/(b*(-(b*c) + a*d)*(c^2 + d^2)*
f)) - (A*d^2 - c*(-(c*C) + B*d))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e
 + f*x]))
```

fricas [B] time = 2.71, size = 1275, normalized size = 4.35

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^2,x, algorithm="fricas")
```

```
[Out] 1/2*(2*(C*a^2*b + C*b^3)*c^3*d^2 - 2*(C*a^3 + B*a^2*b + C*a*b^2 + B*b^3)*c^
2*d^3 + 2*(B*a^3 + A*a^2*b + B*a*b^2 + A*b^3)*c*d^4 - 2*(A*a^3 + A*a*b^2)*d
^5 + 2*(((A - C)*a*b^2 + B*b^3)*c^5 - 2*((A - C)*a^2*b + (A - C)*b^3)*c^4*d
 + ((A - C)*a^3 - 3*B*a^2*b + 3*(A - C)*a*b^2 - B*b^3)*c^3*d^2 + 2*(B*a^3 +
 B*a*b^2)*c^2*d^3 - ((A - C)*a^3 + B*a^2*b)*c*d^4)*f*x + ((C*a^2*b - B*a*b^
2 + A*b^3)*c^5 + 2*(C*a^2*b - B*a*b^2 + A*b^3)*c^3*d^2 + (C*a^2*b - B*a*b^2
 + A*b^3)*c*d^4 + ((C*a^2*b - B*a*b^2 + A*b^3)*c^4*d + 2*(C*a^2*b - B*a*b^2
 + A*b^3)*c^2*d^3 + (C*a^2*b - B*a*b^2 + A*b^3)*d^5)*tan(f*x + e))*log((b^2
*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - ((C*a^2
*b + C*b^3)*c^5 - 2*(B*a^2*b + B*b^3)*c^4*d + (B*a^3 + (3*A - C)*a^2*b + B*
a*b^2 + (3*A - C)*b^3)*c^3*d^2 - 2*((A - C)*a^3 + (A - C)*a*b^2)*c^2*d^3 -
(B*a^3 - A*a^2*b + B*a*b^2 - A*b^3)*c*d^4 + ((C*a^2*b + C*b^3)*c^4*d - 2*(B
*a^2*b + B*b^3)*c^3*d^2 + (B*a^3 + (3*A - C)*a^2*b + B*a*b^2 + (3*A - C)*b^
3)*c^2*d^3 - 2*((A - C)*a^3 + (A - C)*a*b^2)*c*d^4 - (B*a^3 - A*a^2*b + B*a
*b^2 - A*b^3)*d^5)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x +
e) + c^2)/(tan(f*x + e)^2 + 1)) - 2*((C*a^2*b + C*b^3)*c^4*d - (C*a^3 + B*a
^2*b + C*a*b^2 + B*b^3)*c^3*d^2 + (B*a^3 + A*a^2*b + B*a*b^2 + A*b^3)*c^2*d
^3 - (A*a^3 + A*a*b^2)*c*d^4 - (((A - C)*a*b^2 + B*b^3)*c^4*d - 2*((A - C)*
a^2*b + (A - C)*b^3)*c^3*d^2 + ((A - C)*a^3 - 3*B*a^2*b + 3*(A - C)*a*b^2 -
 B*b^3)*c^2*d^3 + 2*(B*a^3 + B*a*b^2)*c*d^4 - ((A - C)*a^3 + B*a^2*b)*d^5)*
```

```
f*x)*tan(f*x + e))/(((a^2*b^2 + b^4)*c^6*d - 2*(a^3*b + a*b^3)*c^5*d^2 + (a
^4 + 3*a^2*b^2 + 2*b^4)*c^4*d^3 - 4*(a^3*b + a*b^3)*c^3*d^4 + (2*a^4 + 3*a^
2*b^2 + b^4)*c^2*d^5 - 2*(a^3*b + a*b^3)*c*d^6 + (a^4 + a^2*b^2)*d^7)*f*tan
(f*x + e) + ((a^2*b^2 + b^4)*c^7 - 2*(a^3*b + a*b^3)*c^6*d + (a^4 + 3*a^2*b
^2 + 2*b^4)*c^5*d^2 - 4*(a^3*b + a*b^3)*c^4*d^3 + (2*a^4 + 3*a^2*b^2 + b^4)
*c^3*d^4 - 2*(a^3*b + a*b^3)*c^2*d^5 + (a^4 + a^2*b^2)*c*d^6)*f)
```

giac [B] time = 5.67, size = 846, normalized size = 2.89

$$\frac{2\left(Aac^2-Cac^2+Bbc^2+2\,Bacd-2\,Abcd+2\,Cbcd-Aad^2+Cad^2-Bbd^2\right)(fx+e)}{a^2c^4+b^2c^4+2\,a^2c^2d^2+2\,b^2c^2d^2+a^2d^4+b^2d^4}+\frac{\left(Bac^2-Abc^2+Cbc^2-2\,Aacd+2\,Cacd-2\,Bbcd-Bad^2+Abd^2-Cbd^2\right)\log\left(\tan\right.}{a^2c^4+b^2c^4+2\,a^2c^2d^2+2\,b^2c^2d^2+a^2d^4+b^2d^4}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c^2 - C*a*c^2 + B*b*c^2 + 2*B*a*c*d - 2*A*b*c*d + 2*C*b*c*d - A
*a*d^2 + C*a*d^2 - B*b*d^2)*(f*x + e)/(a^2*c^4 + b^2*c^4 + 2*a^2*c^2*d^2 +
2*b^2*c^2*d^2 + a^2*d^4 + b^2*d^4) + (B*a*c^2 - A*b*c^2 + C*b*c^2 - 2*A*a*c
*d + 2*C*a*c*d - 2*B*b*c*d - B*a*d^2 + A*b*d^2 - C*b*d^2)*log(tan(f*x + e)^
2 + 1)/(a^2*c^4 + b^2*c^4 + 2*a^2*c^2*d^2 + 2*b^2*c^2*d^2 + a^2*d^4 + b^2*d
^4) + 2*(C*a^2*b^2 - B*a*b^3 + A*b^4)*log(abs(b*tan(f*x + e) + a))/(a^2*b^3
*c^2 + b^5*c^2 - 2*a^3*b^2*c*d - 2*a*b^4*c*d + a^4*b*d^2 + a^2*b^3*d^2) - 2
*(C*b*c^4*d - 2*B*b*c^3*d^2 + B*a*c^2*d^3 + 3*A*b*c^2*d^3 - C*b*c^2*d^3 - 2
*A*a*c*d^4 + 2*C*a*c*d^4 - B*a*d^5 + A*b*d^5)*log(abs(d*tan(f*x + e) + c))/
(b^2*c^6*d - 2*a*b*c^5*d^2 + a^2*c^4*d^3 + 2*b^2*c^4*d^3 - 4*a*b*c^3*d^4 +
2*a^2*c^2*d^5 + b^2*c^2*d^5 - 2*a*b*c*d^6 + a^2*d^7) + 2*(C*b*c^4*d*tan(f*x
 + e) - 2*B*b*c^3*d^2*tan(f*x + e) + B*a*c^2*d^3*tan(f*x + e) + 3*A*b*c^2*d
^3*tan(f*x + e) - C*b*c^2*d^3*tan(f*x + e) - 2*A*a*c*d^4*tan(f*x + e) + 2*C
*a*c*d^4*tan(f*x + e) - B*a*d^5*tan(f*x + e) + A*b*d^5*tan(f*x + e) + 2*C*b
*c^5 - C*a*c^4*d - 3*B*b*c^4*d + 2*B*a*c^3*d^2 + 4*A*b*c^3*d^2 - 3*A*a*c^2*
d^3 + C*a*c^2*d^3 - B*b*c^2*d^3 + 2*A*b*c*d^4 - A*a*d^5)/((b^2*c^6 - 2*a*b*
c^5*d + a^2*c^4*d^2 + 2*b^2*c^4*d^2 - 4*a*b*c^3*d^3 + 2*a^2*c^2*d^4 + b^2*c
^2*d^4 - 2*a*b*c*d^5 + a^2*d^6)*(d*tan(f*x + e) + c)))/f
```

maple [B] time = 0.55, size = 1263, normalized size = 4.31

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))^2,x)
```

```
[Out] -1/f/(a^2+b^2)/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*b*c*d+2/f/(a^2+b^2)/(c^2+d^
2)^2*C*arctan(tan(f*x+e))*b*c*d+2/f/(a*d-b*c)^2/(c^2+d^2)^2*ln(c+d*tan(f*x+
```

```
e))*A*a*c*d^3-3/f/(a*d-b*c)^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*b*c^2*d^2-1/
f/(a*d-b*c)^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*a*c^2*d^2+2/f/(a*d-b*c)^2/(c
^2+d^2)^2*ln(c+d*tan(f*x+e))*B*b*c^3*d-2/f/(a*d-b*c)^2/(c^2+d^2)^2*ln(c+d*t
an(f*x+e))*C*a*c*d^3+1/f/(a*d-b*c)^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*b*c^2
*d^2-1/f/(a^2+b^2)/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*a*c*d+2/f/(a^2+b^2)/(c^
2+d^2)^2*B*arctan(tan(f*x+e))*a*c*d-2/f/(a^2+b^2)/(c^2+d^2)^2*A*arctan(tan(
f*x+e))*b*c*d+1/f/(a^2+b^2)/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*a*c*d-1/2/f/(a
^2+b^2)/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*b*c^2+1/f/(a*d-b*c)^2/(c^2+d^2)^2*
ln(c+d*tan(f*x+e))*B*a*d^4-1/f/(a*d-b*c)^2/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C
*b*c^4+1/f/(a*d-b*c)/(c^2+d^2)/(c+d*tan(f*x+e))*B*c*d+1/f/(a^2+b^2)/(c^2+d^
2)^2*B*arctan(tan(f*x+e))*b*c^2-1/f/(a^2+b^2)/(c^2+d^2)^2*B*arctan(tan(f*x+
e))*b*d^2-1/f/(a^2+b^2)/(c^2+d^2)^2*C*arctan(tan(f*x+e))*a*c^2+1/f/(a^2+b^2
)/(c^2+d^2)^2*C*arctan(tan(f*x+e))*a*d^2-1/2/f/(a^2+b^2)/(c^2+d^2)^2*ln(1+t
an(f*x+e)^2)*B*a*d^2+1/2/f/(a^2+b^2)/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*b*c^2
+1/2/f/(a^2+b^2)/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a*c^2-1/2/f/(a^2+b^2)/(c^
2+d^2)^2*ln(1+tan(f*x+e)^2)*C*b*d^2+1/f/(a^2+b^2)/(c^2+d^2)^2*A*arctan(tan(
f*x+e))*a*c^2-1/f/(a^2+b^2)/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a*d^2+1/f*b/(a
*d-b*c)^2/(a^2+b^2)*ln(a+b*tan(f*x+e))*a^2*C-1/f/(a*d-b*c)^2/(c^2+d^2)^2*ln
(c+d*tan(f*x+e))*A*b*d^4+1/2/f/(a^2+b^2)/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*b
*d^2-1/f*b^2/(a*d-b*c)^2/(a^2+b^2)*ln(a+b*tan(f*x+e))*B*a+1/f*b^3/(a*d-b*c)
^2/(a^2+b^2)*ln(a+b*tan(f*x+e))*A-1/f/(a*d-b*c)/(c^2+d^2)/(c+d*tan(f*x+e))*
A*d^2-1/f/(a*d-b*c)/(c^2+d^2)/(c+d*tan(f*x+e))*c^2*C
```

maxima [A] time = 0.50, size = 513, normalized size = 1.75

$$\frac{\frac{2\left(\left(\left(A-C\right)a+Bb\right)c^{2}+2\left(Ba-\left(A-C\right)b\right)cd-\left(\left(A-C\right)a+Bb\right)d^{2}\right)\left(fx+e\right)}{\left(a^{2}+b^{2}\right)c^{4}+2\left(a^{2}+b^{2}\right)c^{2}d^{2}+\left(a^{2}+b^{2}\right)d^{4}}+\frac{2\left(Ca^{2}b-Bab^{2}+Ab^{3}\right)\log\left(b\tan\left(fx+e\right)+a\right)}{\left(a^{2}b^{2}+b^{4}\right)c^{2}-2\left(a^{3}b+ab^{3}\right)cd+\left(a^{4}+a^{2}b^{2}\right)d^{2}}-\frac{2\left(Cbc^{4}-2\,Bbc^{3}d-2\left(A-C\right)acd^{3}+(B\cdots\right.}{b^{2}c^{6}-2\,abc^{5}d-4\,abc^{3}d^{3}-2\,\cdots}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a + B*b)*c^2 + 2*(B*a - (A - C)*b)*c*d - ((A - C)*a + B*b)
*d^2)*(f*x + e)/((a^2 + b^2)*c^4 + 2*(a^2 + b^2)*c^2*d^2 + (a^2 + b^2)*d^4)
 + 2*(C*a^2*b - B*a*b^2 + A*b^3)*log(b*tan(f*x + e) + a)/((a^2*b^2 + b^4)*c
^2 - 2*(a^3*b + a*b^3)*c*d + (a^4 + a^2*b^2)*d^2) - 2*(C*b*c^4 - 2*B*b*c^3*
d - 2*(A - C)*a*c*d^3 + (B*a + (3*A - C)*b)*c^2*d^2 - (B*a - A*b)*d^4)*log(
d*tan(f*x + e) + c)/(b^2*c^6 - 2*a*b*c^5*d - 4*a*b*c^3*d^3 - 2*a*b*c*d^5 +
a^2*d^6 + (a^2 + 2*b^2)*c^4*d^2 + (2*a^2 + b^2)*c^2*d^4) + ((B*a - (A - C)*
b)*c^2 - 2*((A - C)*a + B*b)*c*d - (B*a - (A - C)*b)*d^2)*log(tan(f*x + e)^
2 + 1)/((a^2 + b^2)*c^4 + 2*(a^2 + b^2)*c^2*d^2 + (a^2 + b^2)*d^4) + 2*(C*c
^2 - B*c*d + A*d^2)/(b*c^4 - a*c^3*d + b*c^2*d^2 - a*c*d^3 + (b*c^3*d - a*c
^2*d^2 + b*c*d^3 - a*d^4)*tan(f*x + e)))/f
```

mupad [B] time = 85.86, size = 430, normalized size = 1.47

$$\frac{\ln\left(\tan\left(e+f\,x\right)-\mathrm{i}\right)\,\left(B-A\,1\mathrm{i}+C\,1\mathrm{i}\right)}{2\,f\,\left(a\,c^2-a\,d^2-2\,b\,c\,d+b\,c^2\,1\mathrm{i}-b\,d^2\,1\mathrm{i}+a\,c\,d\,2\mathrm{i}\right)}-\frac{\ln\left(\tan\left(e+f\,x\right)+1\mathrm{i}\right)\,\left(A\,1\mathrm{i}+B-C\,1\mathrm{i}\right)}{2\,f\,\left(a\,d^2-a\,c^2+2\,b\,c\,d+b\,c^2\,1\mathrm{i}-b\,d^2\,1\mathrm{i}+a\,c\,d\,2\mathrm{i}\right)}+\frac{}{f\,\left(a^4\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))^2),x)
```

```
[Out] (log(tan(e + f*x) - 1i)*(B - A*1i + C*1i))/(2*f*(a*c^2 - a*d^2 + b*c^2*1i -
 b*d^2*1i + a*c*d*2i - 2*b*c*d)) - (log(tan(e + f*x) + 1i)*(A*1i + B - C*1i
))/(2*f*(a*d^2 - a*c^2 + b*c^2*1i - b*d^2*1i + a*c*d*2i + 2*b*c*d)) + (log(
a + b*tan(e + f*x))*(A*b^3 - B*a*b^2 + C*a^2*b))/(f*(a^4*d^2 + b^4*c^2 + a^
2*b^2*c^2 + a^2*b^2*d^2 - 2*a*b^3*c*d - 2*a^3*b*c*d)) - (log(c + d*tan(e +
f*x))*(d^4*(A*b - B*a) + c^2*d^2*(3*A*b + B*a - C*b) + C*b*c^4 - c*d^3*(2*A
*a - 2*C*a) - 2*B*b*c^3*d))/(f*(a^2*d^6 + b^2*c^6 + 2*a^2*c^2*d^4 + a^2*c^4
*d^2 + b^2*c^2*d^4 + 2*b^2*c^4*d^2 - 2*a*b*c*d^5 - 2*a*b*c^5*d - 4*a*b*c^3*
d^3)) - (A*d^2 + C*c^2 - B*c*d)/(f*(a*d - b*c)*(c^2 + d^2)*(c + d*tan(e + f
*x)))
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: NotImplementedError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)
)**2,x)
```

```
[Out] Exception raised: NotImplementedError
```

3.82 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx$

Optimal. Leaf size=509

$$-\frac{d\left(A\left(a^2d^2+b^2\left(c^2+2d^2\right)\right)+a^2\left(-Bcd+2c^2C+Cd^2\right)-abB\left(c^2+d^2\right)+b^2c(cC-Bd)\right)}{f\left(a^2+b^2\right)\left(c^2+d^2\right)(bc-ad)^2(c+d\tan(e+fx))}-\frac{x\left(a^2\left(-A\left(c^2-d^2\right)-\right.\right.}{}$$

```
[Out] -(a^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-b^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2
))+2*a*b*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(a^2+b^2)^2/(c^2+d^2)^2+b*(3*a^3*b*B*
d-2*a^4*C*d+b^4*(-2*A*d+B*c)-a^2*b^2*(4*A*d+B*c)+a*b^3*(2*A*c+B*d-2*C*c))*l
n(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^2)^2/(-a*d+b*c)^3/f+d*(b*(4*A*c^2*d^2+2
*A*d^4-3*B*c^3*d-B*c*d^3+2*C*c^4)-a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*ln(c*cos
(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)^3/(c^2+d^2)^2/f-d*(b^2*c*(-B*d+C*c)-a*b*B*
(c^2+d^2)+a^2*(-B*c*d+2*C*c^2+C*d^2)+A*(a^2*d^2+b^2*(c^2+2*d^2)))/(a^2+b^2)
/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))+(-A*b^2+a*(B*b-C*a))/(a^2+b^2)/(
-a*d+b*c)/f/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
```

Rubi [A] time = 2.15, antiderivative size = 508, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 3, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3649, 3651, 3530}

$$-\frac{d\left(a^2Ad^2+a^2\left(-Bcd+2c^2C+Cd^2\right)-abB\left(c^2+d^2\right)+Ab^2\left(c^2+2d^2\right)+b^2c(cC-Bd)\right)}{f\left(a^2+b^2\right)\left(c^2+d^2\right)(bc-ad)^2(c+d\tan(e+fx))}-\frac{x\left(a^2\left(-A\left(c^2-d^2\right)-\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(c + d*
Tan[e + f*x])^2),x]
```

```
[Out] -(((a^2*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C - 2*B*c*d -
C*d^2 - A*(c^2 - d^2)) + 2*a*b*(2*c*(A - C)*d - B*(c^2 - d^2)))*x)/((a^2 +
b^2)^2*(c^2 + d^2)^2)) + (b*(3*a^3*b*B*d - 2*a^4*C*d + b^4*(B*c - 2*A*d) -
a^2*b^2*(B*c + 4*A*d) + a*b^3*(2*A*c - 2*c*C + B*d))*Log[a*Cos[e + f*x] + b
*Sin[e + f*x]])/((a^2 + b^2)^2*(b*c - a*d)^3*f) + (d*(b*(2*c^4*C - 3*B*c^3*
d + 4*A*c^2*d^2 - B*c*d^3 + 2*A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)
))*Log[c*Cos[e + f*x] + d*Sin[e + f*x]])/((b*c - a*d)^3*(c^2 + d^2)^2*f) -
(d*(a^2*A*d^2 + b^2*c*(c*C - B*d) - a*b*B*(c^2 + d^2) + A*b^2*(c^2 + 2*d^2)
 + a^2*(2*c^2*C - B*c*d + C*d^2)))/((a^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f
*(c + d*Tan[e + f*x])) - (A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f
*(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^2}\,dx = -\frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))} - $$

$$= -\frac{d\left(a^2Ad^2+b^2c(cC-Bd)-abB\left(c^2+d^2\right)+Ab^2\left(c^2+2d^2\right)+\right.}{\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f(c+d\tan(e}$$

$$= -\frac{\left(a^2\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(c^2C-2Bcd-Cd\right.\right.}{\left(a^2+b^2\right)^2\left(c^2\right.}$$

$$= -\frac{\left(a^2\left(c^2C-2Bcd-Cd^2-A\left(c^2-d^2\right)\right)-b^2\left(c^2C-2Bcd-Cd\right.\right.}{\left(a^2+b^2\right)^2\left(c^2\right.}$$

Mathematica [A] time = 8.91, size = 984, normalized size = 1.93

$$-\frac{Ab^2-a(bB-aC)}{(a^2+b^2)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))} - \frac{-\frac{d^2\left(2Adb^2-aA(bc-ad)-(bB-aC)(bc+ad)\right)-c\left((Ab-Cb-aB)d(bc-ad\right.}{(ad-bc)\left(c^2+d^2\right)f(c+d\tan(e+fx))}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^2),x]
```

```
[Out] -((A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])*(
c + d*Tan[e + f*x]))) - (-(((b*(b*c - a*d)^2*(2*a*A*b*c^2 - a^2*B*c^2 + b^2
*B*c^2 - 2*a*b*c^2*C + 2*a^2*A*c*d - 2*A*b^2*c*d + 4*a*b*B*c*d - 2*a^2*c*C*
d + 2*b^2*c*C*d - 2*a*A*b*d^2 + a^2*B*d^2 - b^2*B*d^2 + 2*a*b*C*d^2 - (Sqrt
[-b^2]*(a^2*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C - 2*B*c*
d - C*d^2 - A*(c^2 - d^2)) + 2*a*b*(2*c*(A - C)*d - B*(c^2 - d^2))))/b)*Log
[Sqrt[-b^2] - b*Tan[e + f*x]])/(2*(a^2 + b^2)*(c^2 + d^2)) - (b^2*(c^2 + d^
2)*(3*a^3*b*B*d - 2*a^4*C*d + b^4*(B*c - 2*A*d) - a^2*b^2*(B*c + 4*A*d) + a
*b^3*(2*A*c - 2*c*C + B*d))*Log[a + b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*
d)) + (b*(b*c - a*d)^2*(2*a*A*b*c^2 - a^2*B*c^2 + b^2*B*c^2 - 2*a*b*c^2*C +
 2*a^2*A*c*d - 2*A*b^2*c*d + 4*a*b*B*c*d - 2*a^2*c*C*d + 2*b^2*c*C*d - 2*a*
A*b*d^2 + a^2*B*d^2 - b^2*B*d^2 + 2*a*b*C*d^2 + (Sqrt[-b^2]*(a^2*(c^2*C - 2
*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d
^2)) + 2*a*b*(2*c*(A - C)*d - B*(c^2 - d^2))))/b)*Log[Sqrt[-b^2] + b*Tan[e
```

```
+ f*x]])/(2*(a^2 + b^2)*(c^2 + d^2)) - (b*(a^2 + b^2)*d*(b*(2*c^4*C - 3*B*c
^3*d + 4*A*c^2*d^2 - B*c*d^3 + 2*A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2 - d
^2)))*Log[c + d*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/(b*(-(b*c) + a*d)
*(c^2 + d^2)*f)) - (-(c*(-2*c*(A*b^2 - a*(b*B - a*C))*d + (A*b - a*B - b*C)
*d*(b*c - a*d))) + d^2*(2*A*b^2*d - a*A*(b*c - a*d) - (b*B - a*C)*(b*c + a*
d)))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])))/((a^2 + b^2)*(b*c
 - a*d))
```

fricas [B] time = 6.55, size = 4174, normalized size = 8.20

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] -1/2*(2*(C*a^2*b^4 - B*a*b^5 + A*b^6)*c^6 - 2*(C*a^3*b^3 - B*a^2*b^4 + A*a*
b^5)*c^5*d + 4*(C*a^2*b^4 - B*a*b^5 + A*b^6)*c^4*d^2 + 2*(C*a^5*b + 2*B*a^2
*b^4 - (2*A - C)*a*b^5)*c^3*d^3 - 2*(C*a^6 + B*a^5*b + 2*C*a^4*b^2 + 2*B*a^
3*b^3 + 2*B*a*b^5 - A*b^6)*c^2*d^4 + 2*(B*a^6 + A*a^5*b + 2*B*a^4*b^2 + (2*
A - C)*a^3*b^3 + 2*B*a^2*b^4)*c*d^5 - 2*(A*a^6 + 2*A*a^4*b^2 + A*a^2*b^4)*d
^6 - 2*(((A - C)*a^3*b^3 + 2*B*a^2*b^4 - (A - C)*a*b^5)*c^6 - (3*(A - C)*a^
4*b^2 + 4*B*a^3*b^3 + (A - C)*a^2*b^4 + 2*B*a*b^5)*c^5*d + (3*(A - C)*a^5*b
 + 8*(A - C)*a^3*b^3 + 4*B*a^2*b^4 + (A - C)*a*b^5)*c^4*d^2 - ((A - C)*a^6
- 4*B*a^5*b + 8*(A - C)*a^4*b^2 + 3*(A - C)*a^2*b^4)*c^3*d^3 - (2*B*a^6 - (
A - C)*a^5*b + 4*B*a^4*b^2 - 3*(A - C)*a^3*b^3)*c^2*d^4 + ((A - C)*a^6 + 2*
B*a^5*b - (A - C)*a^4*b^2)*c*d^5)*f*x - 2*((C*a^3*b^3 - B*a^2*b^4 + A*a*b^5
)*c^5*d + (B*a^3*b^3 - (A - 2*C)*a^2*b^4 + C*b^6)*c^4*d^2 - (C*a^5*b + B*a^
4*b^2 + 4*B*a^2*b^4 - (2*A - C)*a*b^5 + B*b^6)*c^3*d^3 + (B*a^5*b + (A - 2*
C)*a^4*b^2 + 4*B*a^3*b^3 + B*a*b^5 + A*b^6)*c^2*d^4 - (A*a^5*b + (2*A - C)*
a^3*b^3 + B*a^2*b^4)*c*d^5 - (C*a^4*b^2 - B*a^3*b^3 + A*a^2*b^4)*d^6 + (((A
 - C)*a^2*b^4 + 2*B*a*b^5 - (A - C)*b^6)*c^5*d - (3*(A - C)*a^3*b^3 + 4*B*a
^2*b^4 + (A - C)*a*b^5 + 2*B*b^6)*c^4*d^2 + (3*(A - C)*a^4*b^2 + 8*(A - C)*
a^2*b^4 + 4*B*a*b^5 + (A - C)*b^6)*c^3*d^3 - ((A - C)*a^5*b - 4*B*a^4*b^2 +
 8*(A - C)*a^3*b^3 + 3*(A - C)*a*b^5)*c^2*d^4 - (2*B*a^5*b - (A - C)*a^4*b^
2 + 4*B*a^3*b^3 - 3*(A - C)*a^2*b^4)*c*d^5 + ((A - C)*a^5*b + 2*B*a^4*b^2 -
 (A - C)*a^3*b^3)*d^6)*f*x)*tan(f*x + e)^2 + ((B*a^3*b^3 - 2*(A - C)*a^2*b^
4 - B*a*b^5)*c^6 + (2*C*a^5*b - 3*B*a^4*b^2 + 4*A*a^3*b^3 - B*a^2*b^4 + 2*A
*a*b^5)*c^5*d + 2*(B*a^3*b^3 - 2*(A - C)*a^2*b^4 - B*a*b^5)*c^4*d^2 + 2*(2*
C*a^5*b - 3*B*a^4*b^2 + 4*A*a^3*b^3 - B*a^2*b^4 + 2*A*a*b^5)*c^3*d^3 + (B*a
^3*b^3 - 2*(A - C)*a^2*b^4 - B*a*b^5)*c^2*d^4 + (2*C*a^5*b - 3*B*a^4*b^2 +
4*A*a^3*b^3 - B*a^2*b^4 + 2*A*a*b^5)*c*d^5 + ((B*a^2*b^4 - 2*(A - C)*a*b^5
- B*b^6)*c^5*d + (2*C*a^4*b^2 - 3*B*a^3*b^3 + 4*A*a^2*b^4 - B*a*b^5 + 2*A*b
^6)*c^4*d^2 + 2*(B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^3*d^3 + 2*(2*C*a^4*
b^2 - 3*B*a^3*b^3 + 4*A*a^2*b^4 - B*a*b^5 + 2*A*b^6)*c^2*d^4 + (B*a^2*b^4 -
```

```
2*(A - C)*a*b^5 - B*b^6)*c*d^5 + (2*C*a^4*b^2 - 3*B*a^3*b^3 + 4*A*a^2*b^4
- B*a*b^5 + 2*A*b^6)*d^6)*tan(f*x + e)^2 + ((B*a^2*b^4 - 2*(A - C)*a*b^5 -
B*b^6)*c^6 + 2*(C*a^4*b^2 - B*a^3*b^3 + (A + C)*a^2*b^4 - B*a*b^5 + A*b^6)*
c^5*d + (2*C*a^5*b - 3*B*a^4*b^2 + 4*A*a^3*b^3 + B*a^2*b^4 - 2*(A - 2*C)*a*
b^5 - 2*B*b^6)*c^4*d^2 + 4*(C*a^4*b^2 - B*a^3*b^3 + (A + C)*a^2*b^4 - B*a*b
^5 + A*b^6)*c^3*d^3 + (4*C*a^5*b - 6*B*a^4*b^2 + 8*A*a^3*b^3 - B*a^2*b^4 +
2*(A + C)*a*b^5 - B*b^6)*c^2*d^4 + 2*(C*a^4*b^2 - B*a^3*b^3 + (A + C)*a^2*b
^4 - B*a*b^5 + A*b^6)*c*d^5 + (2*C*a^5*b - 3*B*a^4*b^2 + 4*A*a^3*b^3 - B*a^
2*b^4 + 2*A*a*b^5)*d^6)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f
*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - (2*(C*a^5*b + 2*C*a^3*b^3 + C*a*b^5)
*c^5*d - 3*(B*a^5*b + 2*B*a^3*b^3 + B*a*b^5)*c^4*d^2 + (B*a^6 + 4*A*a^5*b +
 2*B*a^4*b^2 + 8*A*a^3*b^3 + B*a^2*b^4 + 4*A*a*b^5)*c^3*d^3 - (2*(A - C)*a^
6 + B*a^5*b + 4*(A - C)*a^4*b^2 + 2*B*a^3*b^3 + 2*(A - C)*a^2*b^4 + B*a*b^5
)*c^2*d^4 - (B*a^6 - 2*A*a^5*b + 2*B*a^4*b^2 - 4*A*a^3*b^3 + B*a^2*b^4 - 2*
A*a*b^5)*c*d^5 + (2*(C*a^4*b^2 + 2*C*a^2*b^4 + C*b^6)*c^4*d^2 - 3*(B*a^4*b^
2 + 2*B*a^2*b^4 + B*b^6)*c^3*d^3 + (B*a^5*b + 4*A*a^4*b^2 + 2*B*a^3*b^3 + 8
*A*a^2*b^4 + B*a*b^5 + 4*A*b^6)*c^2*d^4 - (2*(A - C)*a^5*b + B*a^4*b^2 + 4*
(A - C)*a^3*b^3 + 2*B*a^2*b^4 + 2*(A - C)*a*b^5 + B*b^6)*c*d^5 - (B*a^5*b -
 2*A*a^4*b^2 + 2*B*a^3*b^3 - 4*A*a^2*b^4 + B*a*b^5 - 2*A*b^6)*d^6)*tan(f*x
+ e)^2 + (2*(C*a^4*b^2 + 2*C*a^2*b^4 + C*b^6)*c^5*d + (2*C*a^5*b - 3*B*a^4*
b^2 + 4*C*a^3*b^3 - 6*B*a^2*b^4 + 2*C*a*b^5 - 3*B*b^6)*c^4*d^2 - 2*(B*a^5*b
 - 2*A*a^4*b^2 + 2*B*a^3*b^3 - 4*A*a^2*b^4 + B*a*b^5 - 2*A*b^6)*c^3*d^3 + (
B*a^6 + 2*(A + C)*a^5*b + B*a^4*b^2 + 4*(A + C)*a^3*b^3 - B*a^2*b^4 + 2*(A
+ C)*a*b^5 - B*b^6)*c^2*d^4 - 2*((A - C)*a^6 + B*a^5*b + (A - 2*C)*a^4*b^2
+ 2*B*a^3*b^3 - (A + C)*a^2*b^4 + B*a*b^5 - A*b^6)*c*d^5 - (B*a^6 - 2*A*a^5
*b + 2*B*a^4*b^2 - 4*A*a^3*b^3 + B*a^2*b^4 - 2*A*a*b^5)*d^6)*tan(f*x + e))*
log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) -
 2*((C*a^3*b^3 - B*a^2*b^4 + A*a*b^5)*c^6 - (C*a^4*b^2 - B*a^3*b^3 + (A + C
)*a^2*b^4 - B*a*b^5 + A*b^6)*c^5*d + (C*a^5*b + 5*C*a^3*b^3 - 3*B*a^2*b^4 +
 (3*A + C)*a*b^5)*c^4*d^2 - (C*a^6 + B*a^5*b + 5*C*a^4*b^2 + (2*A + 5*C)*a^
2*b^4 - B*a*b^5 + (2*A + C)*b^6)*c^3*d^3 + (B*a^6 + (A + C)*a^5*b + 3*B*a^4
*b^2 + (2*A + 5*C)*a^3*b^3 + (4*A + C)*a*b^5 + B*b^6)*c^2*d^4 - (A*a^6 + B*
a^5*b + (3*A + C)*a^4*b^2 + B*a^3*b^3 + (4*A + C)*a^2*b^4 + 2*A*b^6)*c*d^5
+ (A*a^5*b + (2*A + C)*a^3*b^3 - B*a^2*b^4 + 2*A*a*b^5)*d^6 + (((A - C)*a^2
*b^4 + 2*B*a*b^5 - (A - C)*b^6)*c^6 - 2*((A - C)*a^3*b^3 + B*a^2*b^4 + (A -
 C)*a*b^5 + B*b^6)*c^5*d - (4*B*a^3*b^3 - 7*(A - C)*a^2*b^4 - 2*B*a*b^5 - (
A - C)*b^6)*c^4*d^2 + 2*((A - C)*a^5*b + 2*B*a^4*b^2 + 2*B*a^2*b^4 - (A - C
)*a*b^5)*c^3*d^3 - ((A - C)*a^6 - 2*B*a^5*b + 7*(A - C)*a^4*b^2 + 4*B*a^3*b
^3)*c^2*d^4 - 2*(B*a^6 - (A - C)*a^5*b + B*a^4*b^2 - (A - C)*a^3*b^3)*c*d^5
 + ((A - C)*a^6 + 2*B*a^5*b - (A - C)*a^4*b^2)*d^6)*f*x)*tan(f*x + e))/(((a
^4*b^4 + 2*a^2*b^6 + b^8)*c^7*d - 3*(a^5*b^3 + 2*a^3*b^5 + a*b^7)*c^6*d^2 +
 (3*a^6*b^2 + 8*a^4*b^4 + 7*a^2*b^6 + 2*b^8)*c^5*d^3 - (a^7*b + 8*a^5*b^3 +
 13*a^3*b^5 + 6*a*b^7)*c^4*d^4 + (6*a^6*b^2 + 13*a^4*b^4 + 8*a^2*b^6 + b^8)
*c^3*d^5 - (2*a^7*b + 7*a^5*b^3 + 8*a^3*b^5 + 3*a*b^7)*c^2*d^6 + 3*(a^6*b^2
 + 2*a^4*b^4 + a^2*b^6)*c*d^7 - (a^7*b + 2*a^5*b^3 + a^3*b^5)*d^8)*f*tan(f*
```

```
x + e)^2 + ((a^4*b^4 + 2*a^2*b^6 + b^8)*c^8 - 2*(a^5*b^3 + 2*a^3*b^5 + a*b^
7)*c^7*d + 2*(a^4*b^4 + 2*a^2*b^6 + b^8)*c^6*d^2 + 2*(a^7*b - 3*a^3*b^5 - 2
*a*b^7)*c^5*d^3 - (a^8 + 2*a^6*b^2 - 2*a^2*b^6 - b^8)*c^4*d^4 + 2*(2*a^7*b
+ 3*a^5*b^3 - a*b^7)*c^3*d^5 - 2*(a^8 + 2*a^6*b^2 + a^4*b^4)*c^2*d^6 + 2*(a
^7*b + 2*a^5*b^3 + a^3*b^5)*c*d^7 - (a^8 + 2*a^6*b^2 + a^4*b^4)*d^8)*f*tan(
f*x + e) + ((a^5*b^3 + 2*a^3*b^5 + a*b^7)*c^8 - 3*(a^6*b^2 + 2*a^4*b^4 + a^
2*b^6)*c^7*d + (3*a^7*b + 8*a^5*b^3 + 7*a^3*b^5 + 2*a*b^7)*c^6*d^2 - (a^8 +
 8*a^6*b^2 + 13*a^4*b^4 + 6*a^2*b^6)*c^5*d^3 + (6*a^7*b + 13*a^5*b^3 + 8*a^
3*b^5 + a*b^7)*c^4*d^4 - (2*a^8 + 7*a^6*b^2 + 8*a^4*b^4 + 3*a^2*b^6)*c^3*d^
5 + 3*(a^7*b + 2*a^5*b^3 + a^3*b^5)*c^2*d^6 - (a^8 + 2*a^6*b^2 + a^4*b^4)*c
*d^7)*f)
```

giac [B] time = 4.53, size = 2893, normalized size = 5.68

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c^2 - C*a^2*c^2 + 2*B*a*b*c^2 - A*b^2*c^2 + C*b^2*c^2 + 2*B*a
^2*c*d - 4*A*a*b*c*d + 4*C*a*b*c*d - 2*B*b^2*c*d - A*a^2*d^2 + C*a^2*d^2 -
2*B*a*b*d^2 + A*b^2*d^2 - C*b^2*d^2)*(f*x + e)/(a^4*c^4 + 2*a^2*b^2*c^4 + b
^4*c^4 + 2*a^4*c^2*d^2 + 4*a^2*b^2*c^2*d^2 + 2*b^4*c^2*d^2 + a^4*d^4 + 2*a^
2*b^2*d^4 + b^4*d^4) + (B*a^2*c^2 - 2*A*a*b*c^2 + 2*C*a*b*c^2 - B*b^2*c^2 -
 2*A*a^2*c*d + 2*C*a^2*c*d - 4*B*a*b*c*d + 2*A*b^2*c*d - 2*C*b^2*c*d - B*a^
2*d^2 + 2*A*a*b*d^2 - 2*C*a*b*d^2 + B*b^2*d^2)*log(tan(f*x + e)^2 + 1)/(a^4
*c^4 + 2*a^2*b^2*c^4 + b^4*c^4 + 2*a^4*c^2*d^2 + 4*a^2*b^2*c^2*d^2 + 2*b^4*
c^2*d^2 + a^4*d^4 + 2*a^2*b^2*d^4 + b^4*d^4) - 2*(B*a^2*b^4*c - 2*A*a*b^5*c
 + 2*C*a*b^5*c - B*b^6*c + 2*C*a^4*b^2*d - 3*B*a^3*b^3*d + 4*A*a^2*b^4*d -
B*a*b^5*d + 2*A*b^6*d)*log(abs(b*tan(f*x + e) + a))/(a^4*b^4*c^3 + 2*a^2*b^
6*c^3 + b^8*c^3 - 3*a^5*b^3*c^2*d - 6*a^3*b^5*c^2*d - 3*a*b^7*c^2*d + 3*a^6
*b^2*c*d^2 + 6*a^4*b^4*c*d^2 + 3*a^2*b^6*c*d^2 - a^7*b*d^3 - 2*a^5*b^3*d^3
- a^3*b^5*d^3) + 2*(2*C*b*c^4*d^2 - 3*B*b*c^3*d^3 + B*a*c^2*d^4 + 4*A*b*c^2
*d^4 - 2*A*a*c*d^5 + 2*C*a*c*d^5 - B*b*c*d^5 - B*a*d^6 + 2*A*b*d^6)*log(abs
(d*tan(f*x + e) + c))/(b^3*c^7*d - 3*a*b^2*c^6*d^2 + 3*a^2*b*c^5*d^3 + 2*b^
3*c^5*d^3 - a^3*c^4*d^4 - 6*a*b^2*c^4*d^4 + 6*a^2*b*c^3*d^5 + b^3*c^3*d^5 -
 2*a^3*c^2*d^6 - 3*a*b^2*c^2*d^6 + 3*a^2*b*c*d^7 - a^3*d^8) + (B*a^2*b^3*c^
4*d*tan(f*x + e)^2 - 2*A*a*b^4*c^4*d*tan(f*x + e)^2 + 2*C*a*b^4*c^4*d*tan(f
*x + e)^2 - B*b^5*c^4*d*tan(f*x + e)^2 - 2*B*a^3*b^2*c^3*d^2*tan(f*x + e)^2
 + 2*A*a^2*b^3*c^3*d^2*tan(f*x + e)^2 - 2*C*a^2*b^3*c^3*d^2*tan(f*x + e)^2
- 2*B*a*b^4*c^3*d^2*tan(f*x + e)^2 + 2*A*b^5*c^3*d^2*tan(f*x + e)^2 - 2*C*b
^5*c^3*d^2*tan(f*x + e)^2 + B*a^4*b*c^2*d^3*tan(f*x + e)^2 + 2*A*a^3*b^2*c^
2*d^3*tan(f*x + e)^2 - 2*C*a^3*b^2*c^2*d^3*tan(f*x + e)^2 + 6*B*a^2*b^3*c^2
*d^3*tan(f*x + e)^2 - 2*A*a*b^4*c^2*d^3*tan(f*x + e)^2 + 2*C*a*b^4*c^2*d^3*
```

```
tan(f*x + e)^2 + B*b^5*c^2*d^3*tan(f*x + e)^2 - 2*A*a^4*b*c*d^4*tan(f*x + e
)^2 + 2*C*a^4*b*c*d^4*tan(f*x + e)^2 - 2*B*a^3*b^2*c*d^4*tan(f*x + e)^2 - 2
*A*a^2*b^3*c*d^4*tan(f*x + e)^2 + 2*C*a^2*b^3*c*d^4*tan(f*x + e)^2 - 2*B*a*
b^4*c*d^4*tan(f*x + e)^2 - B*a^4*b*d^5*tan(f*x + e)^2 + 2*A*a^3*b^2*d^5*tan
(f*x + e)^2 - 2*C*a^3*b^2*d^5*tan(f*x + e)^2 + B*a^2*b^3*d^5*tan(f*x + e)^2
 + B*a^2*b^3*c^5*tan(f*x + e) - 2*A*a*b^4*c^5*tan(f*x + e) + 2*C*a*b^4*c^5*
tan(f*x + e) - B*b^5*c^5*tan(f*x + e) - 4*C*a^4*b*c^4*d*tan(f*x + e) + B*a^
3*b^2*c^4*d*tan(f*x + e) - 2*A*a^2*b^3*c^4*d*tan(f*x + e) - 6*C*a^2*b^3*c^4
*d*tan(f*x + e) - B*a*b^4*c^4*d*tan(f*x + e) - 4*C*b^5*c^4*d*tan(f*x + e) +
 B*a^4*b*c^3*d^2*tan(f*x + e) + 4*A*a^3*b^2*c^3*d^2*tan(f*x + e) - 4*C*a^3*
b^2*c^3*d^2*tan(f*x + e) + 8*B*a^2*b^3*c^3*d^2*tan(f*x + e) + 3*B*b^5*c^3*d
^2*tan(f*x + e) + B*a^5*c^2*d^3*tan(f*x + e) - 2*A*a^4*b*c^2*d^3*tan(f*x +
e) - 6*C*a^4*b*c^2*d^3*tan(f*x + e) + 8*B*a^3*b^2*c^2*d^3*tan(f*x + e) - 12
*A*a^2*b^3*c^2*d^3*tan(f*x + e) - 4*C*a^2*b^3*c^2*d^3*tan(f*x + e) + 3*B*a*
b^4*c^2*d^3*tan(f*x + e) - 6*A*b^5*c^2*d^3*tan(f*x + e) - 2*C*b^5*c^2*d^3*t
an(f*x + e) - 2*A*a^5*c*d^4*tan(f*x + e) + 2*C*a^5*c*d^4*tan(f*x + e) - B*a
^4*b*c*d^4*tan(f*x + e) + 3*B*a^2*b^3*c*d^4*tan(f*x + e) + 2*B*b^5*c*d^4*ta
n(f*x + e) - B*a^5*d^5*tan(f*x + e) - 4*C*a^4*b*d^5*tan(f*x + e) + 3*B*a^3*
b^2*d^5*tan(f*x + e) - 6*A*a^2*b^3*d^5*tan(f*x + e) - 2*C*a^2*b^3*d^5*tan(f
*x + e) + 2*B*a*b^4*d^5*tan(f*x + e) - 4*A*b^5*d^5*tan(f*x + e) - 2*C*a^4*b
*c^5 + 3*B*a^3*b^2*c^5 - 4*A*a^2*b^3*c^5 + B*a*b^4*c^5 - 2*A*b^5*c^5 - 2*C*
a^5*c^4*d - 2*B*a^4*b*c^4*d + 2*A*a^3*b^2*c^4*d - 6*C*a^3*b^2*c^4*d - 2*B*a
^2*b^3*c^4*d + 2*A*a*b^4*c^4*d - 4*C*a*b^4*c^4*d + 3*B*a^5*c^3*d^2 + 2*A*a^
4*b*c^3*d^2 - 6*C*a^4*b*c^3*d^2 + 14*B*a^3*b^2*c^3*d^2 - 6*A*a^2*b^3*c^3*d^
2 - 2*C*a^2*b^3*c^3*d^2 + 7*B*a*b^4*c^3*d^2 - 4*A*b^5*c^3*d^2 - 4*A*a^5*c^2
*d^3 - 2*B*a^4*b*c^2*d^3 - 6*A*a^3*b^2*c^2*d^3 - 2*C*a^3*b^2*c^2*d^3 - 2*B*
a^2*b^3*c^2*d^3 - 2*A*a*b^4*c^2*d^3 - 2*C*a*b^4*c^2*d^3 + B*a^5*c*d^4 + 2*A
*a^4*b*c*d^4 - 4*C*a^4*b*c*d^4 + 7*B*a^3*b^2*c*d^4 - 2*A*a^2*b^3*c*d^4 - 2*
C*a^2*b^3*c*d^4 + 4*B*a*b^4*c*d^4 - 2*A*b^5*c*d^4 - 2*A*a^5*d^5 - 4*A*a^3*b
^2*d^5 - 2*A*a*b^4*d^5)/((a^4*b^2*c^6 + 2*a^2*b^4*c^6 + b^6*c^6 - 2*a^5*b*c
^5*d - 4*a^3*b^3*c^5*d - 2*a*b^5*c^5*d + a^6*c^4*d^2 + 4*a^4*b^2*c^4*d^2 +
5*a^2*b^4*c^4*d^2 + 2*b^6*c^4*d^2 - 4*a^5*b*c^3*d^3 - 8*a^3*b^3*c^3*d^3 - 4
*a*b^5*c^3*d^3 + 2*a^6*c^2*d^4 + 5*a^4*b^2*c^2*d^4 + 4*a^2*b^4*c^2*d^4 + b^
6*c^2*d^4 - 2*a^5*b*c*d^5 - 4*a^3*b^3*c*d^5 - 2*a*b^5*c*d^5 + a^6*d^6 + 2*a
^4*b^2*d^6 + a^2*b^4*d^6)*(b*d*tan(f*x + e)^2 + b*c*tan(f*x + e) + a*d*tan(
f*x + e) + a*c)))/f
```

maple [B] time = 0.48, size = 2012, normalized size = 3.95

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e))^2,x
)
```

```
[Out] -2/f/(a^2+b^2)^2/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a*b*c*d-4/f/(a^2+b^2)^2/(
c^2+d^2)^2*A*arctan(tan(f*x+e))*a*b*c*d-2/f*b^4/(a*d-b*c)^3/(a^2+b^2)^2*ln(
a+b*tan(f*x+e))*A*a*c-3/f*b^2/(a*d-b*c)^3/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*a^
3*B*d+1/f/(a^2+b^2)^2/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*b^2*c*d-1/f/(a^2+b^2
)^2/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*a*b*c^2+1/f/(a^2+b^2)^2/(c^2+d^2)^2*ln
(1+tan(f*x+e)^2)*A*a*b*d^2+1/f*b^3/(a*d-b*c)^3/(a^2+b^2)^2*ln(a+b*tan(f*x+e
))*B*a^2*c+4/f*b^3/(a*d-b*c)^3/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*a^2*d-4/f*d
^3/(a*d-b*c)^3/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*b*c^2-1/f*b^4/(a*d-b*c)^3/(
a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*a*d+2/f*b/(a*d-b*c)^3/(a^2+b^2)^2*ln(a+b*ta
n(f*x+e))*a^4*C*d+2/f*b^4/(a*d-b*c)^3/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*C*a*c-
1/f*d^3/(a*d-b*c)^3/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*a*c^2+3/f*d^2/(a*d-b*c
)^3/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*b*c^3+1/f/(a^2+b^2)^2/(c^2+d^2)^2*ln(1
+tan(f*x+e)^2)*C*a*b*c^2-1/f/(a^2+b^2)^2/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*a
*b*d^2-1/f/(a^2+b^2)^2/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*b^2*c*d-2/f/(a^2+b^
2)^2/(c^2+d^2)^2*B*arctan(tan(f*x+e))*b^2*c*d+1/f/(a^2+b^2)^2/(c^2+d^2)^2*l
n(1+tan(f*x+e)^2)*C*a^2*c*d+1/f*d^4/(a*d-b*c)^3/(c^2+d^2)^2*ln(c+d*tan(f*x+
e))*B*b*c-2/f*d^4/(a*d-b*c)^3/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*a*c-2/f*d/(a
*d-b*c)^3/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*b*c^4-1/f/(a^2+b^2)^2/(c^2+d^2)^
2*ln(1+tan(f*x+e)^2)*A*a^2*c*d+2/f*d^4/(a*d-b*c)^3/(c^2+d^2)^2*ln(c+d*tan(f
*x+e))*A*a*c+2/f/(a^2+b^2)^2/(c^2+d^2)^2*B*arctan(tan(f*x+e))*a^2*c*d+2/f/(
a^2+b^2)^2/(c^2+d^2)^2*B*arctan(tan(f*x+e))*a*b*c^2-2/f/(a^2+b^2)^2/(c^2+d^
2)^2*B*arctan(tan(f*x+e))*a*b*d^2+4/f/(a^2+b^2)^2/(c^2+d^2)^2*C*arctan(tan(
f*x+e))*a*b*c*d+2/f*b^5/(a*d-b*c)^3/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*d-2/f*
d^5/(a*d-b*c)^3/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*b+1/f*d^5/(a*d-b*c)^3/(c^2
+d^2)^2*ln(c+d*tan(f*x+e))*B*a+1/2/f/(a^2+b^2)^2/(c^2+d^2)^2*ln(1+tan(f*x+e
)^2)*B*a^2*c^2-1/2/f/(a^2+b^2)^2/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a^2*d^2-1
/f/(a^2+b^2)^2/(c^2+d^2)^2*C*arctan(tan(f*x+e))*b^2*d^2-1/2/f/(a^2+b^2)^2/(
c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*b^2*c^2+1/2/f/(a^2+b^2)^2/(c^2+d^2)^2*ln(1+
tan(f*x+e)^2)*B*b^2*d^2+1/f/(a^2+b^2)^2/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a^
2*c^2-1/f/(a^2+b^2)^2/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a^2*d^2-1/f/(a^2+b^2
)^2/(c^2+d^2)^2*A*arctan(tan(f*x+e))*b^2*c^2+1/f/(a^2+b^2)^2/(c^2+d^2)^2*C*
arctan(tan(f*x+e))*b^2*c^2+1/f/(a^2+b^2)^2/(c^2+d^2)^2*A*arctan(tan(f*x+e))
*b^2*d^2-1/f/(a^2+b^2)^2/(c^2+d^2)^2*C*arctan(tan(f*x+e))*a^2*c^2+1/f/(a^2+
b^2)^2/(c^2+d^2)^2*C*arctan(tan(f*x+e))*a^2*d^2+1/f*d^2/(a*d-b*c)^2/(c^2+d^
2)/(c+d*tan(f*x+e))*B*c-1/f*d/(a*d-b*c)^2/(c^2+d^2)/(c+d*tan(f*x+e))*c^2*C-
1/f*b^5/(a*d-b*c)^3/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*c+1/f*b^2/(a*d-b*c)^2/
(a^2+b^2)/(a+b*tan(f*x+e))*B*a-1/f*b/(a*d-b*c)^2/(a^2+b^2)/(a+b*tan(f*x+e))
*a^2*C-1/f*d^3/(a*d-b*c)^2/(c^2+d^2)/(c+d*tan(f*x+e))*A-1/f*b^3/(a*d-b*c)^2
/(a^2+b^2)/(a+b*tan(f*x+e))*A
```

maxima [B] time = 0.76, size = 1185, normalized size = 2.33

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c^2 + 2*(B*a^2 - 2*(A - C)*a*
b - B*b^2)*c*d - ((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*d^2)*(f*x + e)/((a^4
 + 2*a^2*b^2 + b^4)*c^4 + 2*(a^4 + 2*a^2*b^2 + b^4)*c^2*d^2 + (a^4 + 2*a^2*
b^2 + b^4)*d^4) - 2*((B*a^2*b^3 - 2*(A - C)*a*b^4 - B*b^5)*c + (2*C*a^4*b -
 3*B*a^3*b^2 + 4*A*a^2*b^3 - B*a*b^4 + 2*A*b^5)*d)*log(b*tan(f*x + e) + a)/
((a^4*b^3 + 2*a^2*b^5 + b^7)*c^3 - 3*(a^5*b^2 + 2*a^3*b^4 + a*b^6)*c^2*d +
3*(a^6*b + 2*a^4*b^3 + a^2*b^5)*c*d^2 - (a^7 + 2*a^5*b^2 + a^3*b^4)*d^3) +
2*(2*C*b*c^4*d - 3*B*b*c^3*d^2 + (B*a + 4*A*b)*c^2*d^3 - (2*(A - C)*a + B*b
)*c*d^4 - (B*a - 2*A*b)*d^5)*log(d*tan(f*x + e) + c)/(b^3*c^7 - 3*a*b^2*c^6
*d + 3*a^2*b*c*d^6 - a^3*d^7 + (3*a^2*b + 2*b^3)*c^5*d^2 - (a^3 + 6*a*b^2)*
c^4*d^3 + (6*a^2*b + b^3)*c^3*d^4 - (2*a^3 + 3*a*b^2)*c^2*d^5) + ((B*a^2 -
2*(A - C)*a*b - B*b^2)*c^2 - 2*((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c*d -
(B*a^2 - 2*(A - C)*a*b - B*b^2)*d^2)*log(tan(f*x + e)^2 + 1)/((a^4 + 2*a^2*
b^2 + b^4)*c^4 + 2*(a^4 + 2*a^2*b^2 + b^4)*c^2*d^2 + (a^4 + 2*a^2*b^2 + b^4
)*d^4) - 2*((C*a^2*b - B*a*b^2 + A*b^3)*c^3 + (C*a^3 + C*a*b^2)*c^2*d - (B*
a^3 - C*a^2*b + 2*B*a*b^2 - A*b^3)*c*d^2 + (A*a^3 + A*a*b^2)*d^3 + ((2*C*a^
2*b - B*a*b^2 + (A + C)*b^3)*c^2*d - (B*a^2*b + B*b^3)*c*d^2 + ((A + C)*a^2
*b - B*a*b^2 + 2*A*b^3)*d^3)*tan(f*x + e))/((a^3*b^2 + a*b^4)*c^5 - 2*(a^4*
b + a^2*b^3)*c^4*d + (a^5 + 2*a^3*b^2 + a*b^4)*c^3*d^2 - 2*(a^4*b + a^2*b^3
)*c^2*d^3 + (a^5 + a^3*b^2)*c*d^4 + ((a^2*b^3 + b^5)*c^4*d - 2*(a^3*b^2 + a
*b^4)*c^3*d^2 + (a^4*b + 2*a^2*b^3 + b^5)*c^2*d^3 - 2*(a^3*b^2 + a*b^4)*c*d
^4 + (a^4*b + a^2*b^3)*d^5)*tan(f*x + e)^2 + ((a^2*b^3 + b^5)*c^5 - (a^3*b^
2 + a*b^4)*c^4*d - (a^4*b - b^5)*c^3*d^2 + (a^5 - a*b^4)*c^2*d^3 - (a^4*b +
 a^2*b^3)*c*d^4 + (a^5 + a^3*b^2)*d^5)*tan(f*x + e)))/f
```

mupad [B] time = 31.51, size = 73684, normalized size = 144.76

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))^2),x)
```

```
[Out] (symsum(log((tan(e + f*x)*(4*A^3*a^3*b^4*d^7 + B^3*a^2*b^5*d^7 + 4*A^3*b^7*
c^3*d^4 + 2*C^3*a^5*b^2*d^7 + B^3*b^7*c^2*d^5 + 2*C^3*b^7*c^5*d^2 + 4*A^2*B
*b^7*d^7 - 4*B^3*a^2*b^5*c^2*d^5 - 3*B^3*a^2*b^5*c^4*d^3 + 10*B^3*a^3*b^4*c
^3*d^4 - 3*B^3*a^4*b^3*c^2*d^5 - 4*A*B^2*a*b^6*d^7 - 4*A*B^2*b^7*c*d^6 + 2*
B^3*a*b^6*c*d^6 - 6*A*B^2*a^3*b^4*d^7 + 8*A^2*B*a^2*b^5*d^7 - 3*A^2*B*a^4*b
^3*d^7 + 4*A*C^2*a^3*b^4*d^7 - 4*A*C^2*a^5*b^2*d^7 - 8*A^2*C*a^3*b^4*d^7 +
2*A^2*C*a^5*b^2*d^7 - 6*A*B^2*b^7*c^3*d^4 - 3*B*C^2*a^4*b^3*d^7 + 8*A^2*B*b
^7*c^2*d^5 - 3*A^2*B*b^7*c^4*d^3 + 4*A*C^2*b^7*c^3*d^4 - 4*A*C^2*b^7*c^5*d^
2 - 8*A^2*C*b^7*c^3*d^4 + 2*A^2*C*b^7*c^5*d^2 - 3*B*C^2*b^7*c^4*d^3 - 4*A^3
```

```
*a*b^6*c^2*d^5 - 4*A^3*a^2*b^5*c*d^6 + 6*B^3*a*b^6*c^3*d^4 + 6*B^3*a^3*b^4*
c*d^6 - 2*C^3*a*b^6*c^4*d^3 - 2*C^3*a^4*b^3*c*d^6 - 10*A*B^2*a^2*b^5*c^3*d^
4 - 10*A*B^2*a^3*b^4*c^2*d^5 + 18*A^2*B*a^2*b^5*c^2*d^5 + 2*B*C^2*a^2*b^5*c
^2*d^5 + 4*B*C^2*a^4*b^3*c^4*d^3 + 2*B^2*C*a^2*b^5*c^3*d^4 + 2*B^2*C*a^2*b^
5*c^5*d^2 + 2*B^2*C*a^3*b^4*c^2*d^5 - 6*B^2*C*a^3*b^4*c^4*d^3 - 6*B^2*C*a^4
*b^3*c^3*d^4 + 2*B^2*C*a^5*b^2*c^2*d^5 + 10*A*B*C*a^4*b^3*d^7 + 10*A*B*C*b^
7*c^4*d^3 - 8*A^2*B*a*b^6*c*d^6 - 2*A*B^2*a*b^6*c^2*d^5 + 6*A*B^2*a*b^6*c^4
*d^3 - 2*A*B^2*a^2*b^5*c*d^6 + 6*A*B^2*a^4*b^3*c*d^6 - 4*A^2*B*a*b^6*c^3*d^
4 - 4*A^2*B*a^3*b^4*c*d^6 - 4*A*C^2*a*b^6*c^2*d^5 + 4*A*C^2*a*b^6*c^4*d^3 -
 4*A*C^2*a^2*b^5*c*d^6 + 4*A*C^2*a^4*b^3*c*d^6 + 8*A^2*C*a*b^6*c^2*d^5 - 2*
A^2*C*a*b^6*c^4*d^3 + 8*A^2*C*a^2*b^5*c*d^6 - 2*A^2*C*a^4*b^3*c*d^6 + 4*B*C
^2*a*b^6*c^3*d^4 + 4*B*C^2*a*b^6*c^5*d^2 + 4*B*C^2*a^3*b^4*c*d^6 + 4*B*C^2*
a^5*b^2*c*d^6 - 4*B^2*C*a*b^6*c^2*d^5 - 10*B^2*C*a*b^6*c^4*d^3 - 4*B^2*C*a^
2*b^5*c*d^6 - 10*B^2*C*a^4*b^3*c*d^6 - 4*A*B*C*a^2*b^5*c^2*d^5 + 8*A*B*C*a^
2*b^5*c^4*d^3 + 8*A*B*C*a^4*b^3*c^2*d^5 + 8*A*B*C*a*b^6*c*d^6 - 4*A*B*C*a*b
^6*c^5*d^2 - 4*A*B*C*a^5*b^2*c*d^6))/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a
^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^
8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c
*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^
5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c
^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*
a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*
d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7
*c^7*d - 4*a^7*b*c*d^7) - (4*A^2*C*b^7*d^7 - 6*A^3*a^2*b^5*d^7 - B^3*a^3*b^
4*d^7 - 6*A^3*b^7*c^2*d^5 - B^3*b^7*c^3*d^4 - 4*A^3*b^7*d^7 - 8*A^3*a^2*b^5
*c^2*d^5 - 3*B^3*a^2*b^5*c^3*d^4 - 3*B^3*a^3*b^4*c^2*d^5 + 2*C^3*a^2*b^5*c^
4*d^3 + 2*C^3*a^4*b^3*c^2*d^5 + 4*C^3*a^4*b^3*c^4*d^3 + 4*A^2*B*a*b^6*d^7 +
 4*A^2*B*b^7*c*d^6 + 4*A^3*a*b^6*c*d^6 + A*B^2*a^2*b^5*d^7 - 3*A*B^2*a^4*b^
3*d^7 + 9*A^2*B*a^3*b^4*d^7 + 2*A*C^2*a^2*b^5*d^7 + 4*A*C^2*a^4*b^3*d^7 + 4
*A^2*C*a^2*b^5*d^7 - 4*A^2*C*a^4*b^3*d^7 + A*B^2*b^7*c^2*d^5 - 3*A*B^2*b^7*
c^4*d^3 - B*C^2*a^3*b^4*d^7 - 2*B*C^2*a^5*b^2*d^7 + 9*A^2*B*b^7*c^3*d^4 + B
^2*C*a^2*b^5*d^7 + 3*B^2*C*a^4*b^3*d^7 + 2*A*C^2*b^7*c^2*d^5 + 4*A*C^2*b^7*
c^4*d^3 + 4*A^2*C*b^7*c^2*d^5 - 4*A^2*C*b^7*c^4*d^3 - B*C^2*b^7*c^3*d^4 - 2
*B*C^2*b^7*c^5*d^2 + B^2*C*b^7*c^2*d^5 + 3*B^2*C*b^7*c^4*d^3 + 2*A^3*a*b^6*
c^3*d^4 + 2*A^3*a^3*b^4*c*d^6 + B^3*a*b^6*c^2*d^5 + 3*B^3*a*b^6*c^4*d^3 + B
^3*a^2*b^5*c*d^6 + 3*B^3*a^4*b^3*c*d^6 + 2*C^3*a*b^6*c^3*d^4 + 2*C^3*a*b^6*
c^5*d^2 + 2*C^3*a^3*b^4*c*d^6 + 2*C^3*a^5*b^2*c*d^6 - 4*A*B*C*a*b^6*d^7 - 4
*A*B*C*b^7*c*d^6 + 14*A*B^2*a^2*b^5*c^2*d^5 + 3*A*B^2*a^2*b^5*c^4*d^3 - 10*
A*B^2*a^3*b^4*c^3*d^4 + 3*A*B^2*a^4*b^3*c^2*d^5 + 7*A^2*B*a^2*b^5*c^3*d^4 +
 7*A^2*B*a^3*b^4*c^2*d^5 + 8*A*C^2*a^2*b^5*c^2*d^5 + 4*A*C^2*a^2*b^5*c^4*d^
3 + 4*A*C^2*a^4*b^3*c^2*d^5 - 4*A*C^2*a^4*b^3*c^4*d^3 - 6*A^2*C*a^2*b^5*c^4
*d^3 - 6*A^2*C*a^4*b^3*c^2*d^5 - B*C^2*a^2*b^5*c^3*d^4 + 2*B*C^2*a^2*b^5*c^
5*d^2 - B*C^2*a^3*b^4*c^2*d^5 - 6*B*C^2*a^3*b^4*c^4*d^3 - 6*B*C^2*a^4*b^3*c
^3*d^4 + 2*B*C^2*a^5*b^2*c^2*d^5 - 6*B^2*C*a^2*b^5*c^2*d^5 - B^2*C*a^2*b^5*
c^4*d^3 + 10*B^2*C*a^3*b^4*c^3*d^4 - B^2*C*a^4*b^3*c^2*d^5 - 8*A*B*C*a^3*b^
```

4*d^7 + 2*A*B*C*a^5*b^2*d^7 - 8*A*B*C*b^7*c^3*d^4 + 2*A*B*C*b^7*c^5*d^2 - 6
*A*B^2*a*b^6*c*d^6 + 4*A*C^2*a*b^6*c*d^6 - 8*A^2*C*a*b^6*c*d^6 + 2*B^2*C*a*
b^6*c*d^6 - 8*A*B^2*a*b^6*c^3*d^4 - 8*A*B^2*a^3*b^4*c*d^6 - A^2*B*a*b^6*c^2
*d^5 - 3*A^2*B*a*b^6*c^4*d^3 - A^2*B*a^2*b^5*c*d^6 - 3*A^2*B*a^4*b^3*c*d^6
- 2*A*C^2*a*b^6*c^3*d^4 - 4*A*C^2*a*b^6*c^5*d^2 - 2*A*C^2*a^3*b^4*c*d^6 - 4
*A*C^2*a^5*b^2*c*d^6 - 2*A^2*C*a*b^6*c^3*d^4 + 2*A^2*C*a*b^6*c^5*d^2 - 2*A^
2*C*a^3*b^4*c*d^6 + 2*A^2*C*a^5*b^2*c*d^6 - 3*B*C^2*a*b^6*c^2*d^5 - 5*B*C^2
*a*b^6*c^4*d^3 - 3*B*C^2*a^2*b^5*c*d^6 - 5*B*C^2*a^4*b^3*c*d^6 + 4*B^2*C*a*
b^6*c^3*d^4 - 2*B^2*C*a*b^6*c^5*d^2 + 4*B^2*C*a^3*b^4*c*d^6 - 2*B^2*C*a^5*b
^2*c*d^6 - 6*A*B*C*a^2*b^5*c^3*d^4 - 2*A*B*C*a^2*b^5*c^5*d^2 - 6*A*B*C*a^3*
b^4*c^2*d^5 + 6*A*B*C*a^3*b^4*c^4*d^3 + 6*A*B*C*a^4*b^3*c^3*d^4 - 2*A*B*C*a
^5*b^2*c^2*d^5 + 4*A*B*C*a*b^6*c^2*d^5 + 8*A*B*C*a*b^6*c^4*d^3 + 4*A*B*C*a^
2*b^5*c*d^6 + 8*A*B*C*a^4*b^3*c*d^6)/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a
^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^
8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c
*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^
5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c
^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*
a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*
d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7
*c^7*d - 4*a^7*b*c*d^7) - root(144*a^13*b*c^5*d^9*f^4 + 144*a^9*b^5*c*d^13*
f^4 + 144*a^5*b^9*c^13*d*f^4 + 144*a*b^13*c^9*d^5*f^4 + 96*a^13*b*c^7*d^7*f
^4 + 96*a^13*b*c^3*d^11*f^4 + 96*a^11*b^3*c*d^13*f^4 + 96*a^7*b^7*c^13*d*f^
4 + 96*a^7*b^7*c*d^13*f^4 + 96*a^3*b^11*c^13*d*f^4 + 96*a*b^13*c^11*d^3*f^4
 + 96*a*b^13*c^7*d^7*f^4 + 24*a^13*b*c^9*d^5*f^4 + 24*a^9*b^5*c^13*d*f^4 +
24*a^5*b^9*c*d^13*f^4 + 24*a*b^13*c^5*d^9*f^4 + 24*a^13*b*c*d^13*f^4 + 24*a
*b^13*c^13*d*f^4 + 3648*a^7*b^7*c^7*d^7*f^4 - 3188*a^8*b^6*c^6*d^8*f^4 - 31
88*a^6*b^8*c^8*d^6*f^4 - 2912*a^8*b^6*c^8*d^6*f^4 - 2912*a^6*b^8*c^6*d^8*f^
4 + 2592*a^9*b^5*c^7*d^7*f^4 + 2592*a^7*b^7*c^9*d^5*f^4 + 2592*a^7*b^7*c^5*
d^9*f^4 + 2592*a^5*b^9*c^7*d^7*f^4 + 2168*a^9*b^5*c^5*d^9*f^4 + 2168*a^5*b^
9*c^9*d^5*f^4 - 1776*a^10*b^4*c^6*d^8*f^4 - 1776*a^8*b^6*c^4*d^10*f^4 - 177
6*a^6*b^8*c^10*d^4*f^4 - 1776*a^4*b^10*c^8*d^6*f^4 + 1568*a^9*b^5*c^9*d^5*f
^4 + 1568*a^5*b^9*c^5*d^9*f^4 - 1344*a^10*b^4*c^8*d^6*f^4 - 1344*a^8*b^6*c^
10*d^4*f^4 - 1344*a^6*b^8*c^4*d^10*f^4 - 1344*a^4*b^10*c^6*d^8*f^4 - 1164*a
^10*b^4*c^4*d^10*f^4 - 1164*a^4*b^10*c^10*d^4*f^4 + 896*a^11*b^3*c^5*d^9*f^
4 + 896*a^9*b^5*c^3*d^11*f^4 + 896*a^5*b^9*c^11*d^3*f^4 + 896*a^3*b^11*c^9*
d^5*f^4 + 864*a^11*b^3*c^7*d^7*f^4 + 864*a^7*b^7*c^11*d^3*f^4 + 864*a^7*b^7
*c^3*d^11*f^4 + 864*a^3*b^11*c^7*d^7*f^4 - 480*a^10*b^4*c^10*d^4*f^4 - 480*
a^4*b^10*c^4*d^10*f^4 + 464*a^11*b^3*c^3*d^11*f^4 + 464*a^3*b^11*c^11*d^3*f
^4 - 424*a^12*b^2*c^6*d^8*f^4 - 424*a^8*b^6*c^2*d^12*f^4 - 424*a^6*b^8*c^12
*d^2*f^4 - 424*a^2*b^12*c^8*d^6*f^4 + 416*a^11*b^3*c^9*d^5*f^4 + 416*a^9*b^
5*c^11*d^3*f^4 + 416*a^5*b^9*c^3*d^11*f^4 + 416*a^3*b^11*c^5*d^9*f^4 - 336*
a^12*b^2*c^4*d^10*f^4 - 336*a^10*b^4*c^2*d^12*f^4 - 336*a^4*b^10*c^12*d^2*f
^4 - 336*a^2*b^12*c^10*d^4*f^4 - 256*a^12*b^2*c^8*d^6*f^4 - 256*a^8*b^6*c^1
2*d^2*f^4 - 256*a^6*b^8*c^2*d^12*f^4 - 256*a^2*b^12*c^6*d^8*f^4 - 124*a^12*

b^2*c^2*d^12*f^4 - 124*a^2*b^12*c^12*d^2*f^4 + 80*a^11*b^3*c^11*d^3*f^4 + 8
0*a^3*b^11*c^3*d^11*f^4 - 60*a^12*b^2*c^10*d^4*f^4 - 60*a^10*b^4*c^12*d^2*f
^4 - 60*a^4*b^10*c^2*d^12*f^4 - 60*a^2*b^12*c^4*d^10*f^4 - 24*b^14*c^10*d^4
*f^4 - 16*b^14*c^12*d^2*f^4 - 16*b^14*c^8*d^6*f^4 - 4*b^14*c^6*d^8*f^4 - 24
*a^14*c^4*d^10*f^4 - 16*a^14*c^6*d^8*f^4 - 16*a^14*c^2*d^12*f^4 - 4*a^14*c^
8*d^6*f^4 - 24*a^10*b^4*d^14*f^4 - 16*a^12*b^2*d^14*f^4 - 16*a^8*b^6*d^14*f
^4 - 4*a^6*b^8*d^14*f^4 - 24*a^4*b^10*c^14*f^4 - 16*a^6*b^8*c^14*f^4 - 16*a
^2*b^12*c^14*f^4 - 4*a^8*b^6*c^14*f^4 - 4*b^14*c^14*f^4 - 4*a^14*d^14*f^4 +
 36*A*C*a^9*b*c*d^9*f^2 + 36*A*C*a*b^9*c^9*d*f^2 + 32*A*C*a*b^9*c*d^9*f^2 -
 552*B*C*a^7*b^3*c^4*d^6*f^2 - 552*B*C*a^4*b^6*c^7*d^3*f^2 - 408*B*C*a^5*b^
5*c^4*d^6*f^2 - 408*B*C*a^4*b^6*c^5*d^5*f^2 + 360*B*C*a^6*b^4*c^3*d^7*f^2 +
 360*B*C*a^3*b^7*c^6*d^4*f^2 - 248*B*C*a^7*b^3*c^2*d^8*f^2 - 248*B*C*a^2*b^
8*c^7*d^3*f^2 + 184*B*C*a^6*b^4*c^5*d^5*f^2 + 184*B*C*a^5*b^5*c^6*d^4*f^2 +
 152*B*C*a^8*b^2*c^3*d^7*f^2 - 152*B*C*a^5*b^5*c^2*d^8*f^2 + 152*B*C*a^3*b^
7*c^8*d^2*f^2 - 152*B*C*a^2*b^8*c^5*d^5*f^2 - 104*B*C*a^7*b^3*c^6*d^4*f^2 -
 104*B*C*a^6*b^4*c^7*d^3*f^2 + 64*B*C*a^8*b^2*c^5*d^5*f^2 + 64*B*C*a^5*b^5*
c^8*d^2*f^2 - 56*B*C*a^4*b^6*c^3*d^7*f^2 - 56*B*C*a^3*b^7*c^4*d^6*f^2 - 24*
B*C*a^8*b^2*c^7*d^3*f^2 - 24*B*C*a^7*b^3*c^8*d^2*f^2 - 24*B*C*a^3*b^7*c^2*d
^8*f^2 - 24*B*C*a^2*b^8*c^3*d^7*f^2 - 696*A*C*a^5*b^5*c^5*d^5*f^2 + 536*A*C
*a^6*b^4*c^6*d^4*f^2 + 536*A*C*a^6*b^4*c^4*d^6*f^2 + 536*A*C*a^4*b^6*c^6*d^
4*f^2 + 472*A*C*a^4*b^6*c^4*d^6*f^2 - 232*A*C*a^7*b^3*c^5*d^5*f^2 - 232*A*C
*a^5*b^5*c^7*d^3*f^2 + 216*A*C*a^3*b^7*c^3*d^7*f^2 + 168*A*C*a^7*b^3*c^3*d^
7*f^2 + 168*A*C*a^3*b^7*c^7*d^3*f^2 - 154*A*C*a^8*b^2*c^2*d^8*f^2 - 154*A*C
*a^2*b^8*c^8*d^2*f^2 + 62*A*C*a^8*b^2*c^6*d^4*f^2 + 62*A*C*a^6*b^4*c^8*d^2*
f^2 - 40*A*C*a^7*b^3*c^7*d^3*f^2 - 40*A*C*a^5*b^5*c^3*d^7*f^2 - 40*A*C*a^3*
b^7*c^5*d^5*f^2 + 32*A*C*a^6*b^4*c^2*d^8*f^2 + 32*A*C*a^2*b^8*c^6*d^4*f^2 -
 32*A*C*a^2*b^8*c^2*d^8*f^2 + 30*A*C*a^4*b^6*c^2*d^8*f^2 + 30*A*C*a^2*b^8*c
^4*d^6*f^2 + 16*A*C*a^8*b^2*c^4*d^6*f^2 + 16*A*C*a^4*b^6*c^8*d^2*f^2 - 488*
A*B*a^6*b^4*c^3*d^7*f^2 - 488*A*B*a^3*b^7*c^6*d^4*f^2 + 440*A*B*a^7*b^3*c^4
*d^6*f^2 + 440*A*B*a^4*b^6*c^7*d^3*f^2 - 360*A*B*a^6*b^4*c^5*d^5*f^2 - 360*
A*B*a^5*b^5*c^6*d^4*f^2 - 192*A*B*a^8*b^2*c^3*d^7*f^2 - 192*A*B*a^3*b^7*c^8
*d^2*f^2 - 168*A*B*a^3*b^7*c^2*d^8*f^2 - 168*A*B*a^2*b^8*c^3*d^7*f^2 - 152*
A*B*a^4*b^6*c^3*d^7*f^2 - 152*A*B*a^3*b^7*c^4*d^6*f^2 - 120*A*B*a^8*b^2*c^5
*d^5*f^2 + 120*A*B*a^7*b^3*c^2*d^8*f^2 - 120*A*B*a^5*b^5*c^8*d^2*f^2 + 120*
A*B*a^5*b^5*c^4*d^6*f^2 - 120*A*B*a^5*b^5*c^2*d^8*f^2 + 120*A*B*a^4*b^6*c^5
*d^5*f^2 + 120*A*B*a^2*b^8*c^7*d^3*f^2 - 120*A*B*a^2*b^8*c^5*d^5*f^2 + 40*A
*B*a^7*b^3*c^6*d^4*f^2 + 40*A*B*a^6*b^4*c^7*d^3*f^2 - 72*B*C*a^9*b*c^4*d^6*
f^2 - 72*B*C*a^4*b^6*c^9*d*f^2 - 64*B*C*a^4*b^6*c*d^9*f^2 - 64*B*C*a*b^9*c^
4*d^6*f^2 - 32*B*C*a^8*b^2*c*d^9*f^2 - 32*B*C*a*b^9*c^8*d^2*f^2 - 16*B*C*a^
2*b^8*c*d^9*f^2 - 16*B*C*a*b^9*c^2*d^8*f^2 + 8*B*C*a^9*b*c^6*d^4*f^2 - 8*B*
C*a^9*b*c^2*d^8*f^2 + 8*B*C*a^6*b^4*c^9*d*f^2 - 8*B*C*a^2*b^8*c^9*d*f^2 + 1
04*A*C*a^7*b^3*c*d^9*f^2 + 104*A*C*a*b^9*c^7*d^3*f^2 + 96*A*C*a^3*b^7*c*d^9
*f^2 + 96*A*C*a*b^9*c^3*d^7*f^2 + 72*A*C*a^9*b*c^3*d^7*f^2 + 72*A*C*a^3*b^7
*c^9*d*f^2 + 68*A*C*a^5*b^5*c*d^9*f^2 + 68*A*C*a*b^9*c^5*d^5*f^2 - 28*A*C*a
^9*b*c^5*d^5*f^2 - 28*A*C*a^5*b^5*c^9*d*f^2 + 80*A*B*a^9*b*c^4*d^6*f^2 + 80

*A*B*a^4*b^6*c^9*d*f^2 + 24*A*B*a^8*b^2*c*d^9*f^2 - 24*A*B*a^6*b^4*c*d^9*f^
2 + 24*A*B*a^4*b^6*c*d^9*f^2 - 24*A*B*a^2*b^8*c*d^9*f^2 + 24*A*B*a*b^9*c^8*
d^2*f^2 - 24*A*B*a*b^9*c^6*d^4*f^2 + 24*A*B*a*b^9*c^4*d^6*f^2 - 24*A*B*a*b^
9*c^2*d^8*f^2 - 32*B*C*b^10*c^7*d^3*f^2 - 8*B*C*b^10*c^5*d^5*f^2 + 34*A*C*b
^10*c^6*d^4*f^2 + 16*B*C*a^10*c^3*d^7*f^2 + 16*A*C*b^10*c^4*d^6*f^2 - 12*A*
C*b^10*c^8*d^2*f^2 - 96*A*B*b^10*c^5*d^5*f^2 - 72*A*B*b^10*c^3*d^7*f^2 - 32
*B*C*a^7*b^3*d^10*f^2 - 28*A*C*a^10*c^2*d^8*f^2 - 24*A*B*b^10*c^7*d^3*f^2 -
 8*B*C*a^5*b^5*d^10*f^2 + 2*A*C*a^10*c^4*d^6*f^2 + 34*A*C*a^6*b^4*d^10*f^2
+ 16*B*C*a^3*b^7*c^10*f^2 + 16*A*C*a^4*b^6*d^10*f^2 - 16*A*B*a^10*c^3*d^7*f
^2 - 12*A*C*a^8*b^2*d^10*f^2 - 96*A*B*a^5*b^5*d^10*f^2 - 72*A*B*a^3*b^7*d^1
0*f^2 - 28*A*C*a^2*b^8*c^10*f^2 - 24*A*B*a^7*b^3*d^10*f^2 + 2*A*C*a^4*b^6*c
^10*f^2 - 16*A*B*a^3*b^7*c^10*f^2 + 444*C^2*a^5*b^5*c^5*d^5*f^2 + 148*C^2*a
^7*b^3*c^5*d^5*f^2 + 148*C^2*a^5*b^5*c^7*d^3*f^2 + 148*C^2*a^5*b^5*c^3*d^7*
f^2 + 148*C^2*a^3*b^7*c^5*d^5*f^2 - 140*C^2*a^6*b^4*c^6*d^4*f^2 - 140*C^2*a
^6*b^4*c^4*d^6*f^2 - 140*C^2*a^4*b^6*c^6*d^4*f^2 - 140*C^2*a^4*b^6*c^4*d^6*
f^2 + 109*C^2*a^8*b^2*c^2*d^8*f^2 + 109*C^2*a^2*b^8*c^8*d^2*f^2 + 48*C^2*a^
8*b^2*c^4*d^6*f^2 + 48*C^2*a^6*b^4*c^2*d^8*f^2 + 48*C^2*a^4*b^6*c^8*d^2*f^2
 + 48*C^2*a^2*b^8*c^6*d^4*f^2 + 20*C^2*a^7*b^3*c^7*d^3*f^2 - 20*C^2*a^7*b^3
*c^3*d^7*f^2 - 20*C^2*a^3*b^7*c^7*d^3*f^2 + 20*C^2*a^3*b^7*c^3*d^7*f^2 + 17
*C^2*a^8*b^2*c^6*d^4*f^2 + 17*C^2*a^6*b^4*c^8*d^2*f^2 + 17*C^2*a^4*b^6*c^2*
d^8*f^2 + 17*C^2*a^2*b^8*c^4*d^6*f^2 + 16*C^2*a^8*b^2*c^8*d^2*f^2 + 16*C^2*
a^2*b^8*c^2*d^8*f^2 - 396*B^2*a^5*b^5*c^5*d^5*f^2 + 308*B^2*a^6*b^4*c^4*d^6
*f^2 + 308*B^2*a^4*b^6*c^6*d^4*f^2 + 300*B^2*a^4*b^6*c^4*d^6*f^2 + 284*B^2*
a^6*b^4*c^6*d^4*f^2 - 132*B^2*a^7*b^3*c^5*d^5*f^2 - 132*B^2*a^5*b^5*c^7*d^3
*f^2 - 84*B^2*a^5*b^5*c^3*d^7*f^2 - 84*B^2*a^3*b^7*c^5*d^5*f^2 + 61*B^2*a^4
*b^6*c^2*d^8*f^2 + 61*B^2*a^2*b^8*c^4*d^6*f^2 - 59*B^2*a^8*b^2*c^2*d^8*f^2
- 59*B^2*a^2*b^8*c^8*d^2*f^2 + 56*B^2*a^6*b^4*c^2*d^8*f^2 + 56*B^2*a^2*b^8*
c^6*d^4*f^2 + 52*B^2*a^7*b^3*c^3*d^7*f^2 + 52*B^2*a^3*b^7*c^7*d^3*f^2 + 44*
B^2*a^3*b^7*c^3*d^7*f^2 + 33*B^2*a^8*b^2*c^6*d^4*f^2 + 33*B^2*a^6*b^4*c^8*d
^2*f^2 + 20*B^2*a^8*b^2*c^4*d^6*f^2 - 20*B^2*a^7*b^3*c^7*d^3*f^2 + 20*B^2*a
^4*b^6*c^8*d^2*f^2 + 8*B^2*a^2*b^8*c^2*d^8*f^2 + 337*A^2*a^4*b^6*c^2*d^8*f^
2 + 337*A^2*a^2*b^8*c^4*d^6*f^2 + 272*A^2*a^2*b^8*c^2*d^8*f^2 + 252*A^2*a^5
*b^5*c^5*d^5*f^2 + 244*A^2*a^4*b^6*c^4*d^6*f^2 - 236*A^2*a^3*b^7*c^3*d^7*f^
2 + 176*A^2*a^6*b^4*c^2*d^8*f^2 + 176*A^2*a^2*b^8*c^6*d^4*f^2 - 148*A^2*a^7
*b^3*c^3*d^7*f^2 - 148*A^2*a^3*b^7*c^7*d^3*f^2 - 140*A^2*a^6*b^4*c^6*d^4*f^
2 + 109*A^2*a^8*b^2*c^2*d^8*f^2 + 109*A^2*a^2*b^8*c^8*d^2*f^2 - 108*A^2*a^5
*b^5*c^3*d^7*f^2 - 108*A^2*a^3*b^7*c^5*d^5*f^2 + 84*A^2*a^7*b^3*c^5*d^5*f^2
 + 84*A^2*a^5*b^5*c^7*d^3*f^2 + 32*A^2*a^8*b^2*c^4*d^6*f^2 + 32*A^2*a^4*b^6
*c^8*d^2*f^2 + 20*A^2*a^7*b^3*c^7*d^3*f^2 - 15*A^2*a^8*b^2*c^6*d^4*f^2 - 15
*A^2*a^6*b^4*c^8*d^2*f^2 - 12*A^2*a^6*b^4*c^4*d^6*f^2 - 12*A^2*a^4*b^6*c^6*
d^4*f^2 + 8*B*C*b^10*c^9*d*f^2 - 16*B*C*a^10*c*d^9*f^2 - 16*A*B*b^10*c^9*d*
f^2 - 16*A*B*b^10*c*d^9*f^2 + 8*B*C*a^9*b*d^10*f^2 - 16*B*C*a*b^9*c^10*f^2
+ 16*A*B*a^10*c*d^9*f^2 - 16*A*B*a^9*b*d^10*f^2 - 16*A*B*a*b^9*d^10*f^2 + 1
6*A*B*a*b^9*c^10*f^2 + 22*C^2*a^9*b*c^5*d^5*f^2 + 22*C^2*a^5*b^5*c^9*d*f^2
+ 22*C^2*a^5*b^5*c*d^9*f^2 + 22*C^2*a*b^9*c^5*d^5*f^2 - 20*C^2*a^9*b*c^3*d^

7*f^2 - 20*C^2*a^7*b^3*c*d^9*f^2 - 20*C^2*a^3*b^7*c^9*d*f^2 - 20*C^2*a*b^9*
c^7*d^3*f^2 + 36*B^2*a^7*b^3*c*d^9*f^2 + 36*B^2*a*b^9*c^7*d^3*f^2 + 28*B^2*
a^9*b*c^3*d^7*f^2 + 28*B^2*a^3*b^7*c^9*d*f^2 + 24*B^2*a^3*b^7*c*d^9*f^2 + 2
4*B^2*a*b^9*c^3*d^7*f^2 - 18*B^2*a^9*b*c^5*d^5*f^2 - 18*B^2*a^5*b^5*c^9*d*f
^2 + 6*B^2*a^5*b^5*c*d^9*f^2 + 6*B^2*a*b^9*c^5*d^5*f^2 - 96*A^2*a^3*b^7*c*d
^9*f^2 - 96*A^2*a*b^9*c^3*d^7*f^2 - 90*A^2*a^5*b^5*c*d^9*f^2 - 90*A^2*a*b^9
*c^5*d^5*f^2 - 84*A^2*a^7*b^3*c*d^9*f^2 - 84*A^2*a*b^9*c^7*d^3*f^2 - 52*A^2
*a^9*b*c^3*d^7*f^2 - 52*A^2*a^3*b^7*c^9*d*f^2 + 6*A^2*a^9*b*c^5*d^5*f^2 + 6
*A^2*a^5*b^5*c^9*d*f^2 - 10*C^2*a^9*b*c*d^9*f^2 - 10*C^2*a*b^9*c^9*d*f^2 +
14*B^2*a^9*b*c*d^9*f^2 + 14*B^2*a*b^9*c^9*d*f^2 + 8*B^2*a*b^9*c*d^9*f^2 - 3
2*A^2*a*b^9*c*d^9*f^2 - 26*A^2*a^9*b*c*d^9*f^2 - 26*A^2*a*b^9*c^9*d*f^2 + 2
*A*C*b^10*c^10*f^2 + 2*A*C*a^10*d^10*f^2 + 14*C^2*b^10*c^8*d^2*f^2 - C^2*b^
10*c^6*d^4*f^2 + 31*B^2*b^10*c^6*d^4*f^2 + 20*B^2*b^10*c^4*d^6*f^2 + 14*C^2
*a^10*c^2*d^8*f^2 + 4*B^2*b^10*c^2*d^8*f^2 + 2*B^2*b^10*c^8*d^2*f^2 - C^2*a
^10*c^4*d^6*f^2 + 80*A^2*b^10*c^4*d^6*f^2 + 64*A^2*b^10*c^2*d^8*f^2 + 31*A^
2*b^10*c^6*d^4*f^2 + 14*C^2*a^8*b^2*d^10*f^2 + 14*A^2*b^10*c^8*d^2*f^2 - 10
*B^2*a^10*c^2*d^8*f^2 + 3*B^2*a^10*c^4*d^6*f^2 - C^2*a^6*b^4*d^10*f^2 + 31*
B^2*a^6*b^4*d^10*f^2 + 20*B^2*a^4*b^6*d^10*f^2 + 14*C^2*a^2*b^8*c^10*f^2 +
14*A^2*a^10*c^2*d^8*f^2 + 4*B^2*a^2*b^8*d^10*f^2 + 2*B^2*a^8*b^2*d^10*f^2 -
 C^2*a^4*b^6*c^10*f^2 - A^2*a^10*c^4*d^6*f^2 + 80*A^2*a^4*b^6*d^10*f^2 + 64
*A^2*a^2*b^8*d^10*f^2 + 31*A^2*a^6*b^4*d^10*f^2 + 14*A^2*a^8*b^2*d^10*f^2 -
 10*B^2*a^2*b^8*c^10*f^2 + 3*B^2*a^4*b^6*c^10*f^2 + 14*A^2*a^2*b^8*c^10*f^2
 - A^2*a^4*b^6*c^10*f^2 - C^2*b^10*c^10*f^2 - C^2*a^10*d^10*f^2 + 16*A^2*b^
10*d^10*f^2 + 3*B^2*b^10*c^10*f^2 + 3*B^2*a^10*d^10*f^2 - A^2*b^10*c^10*f^2
 - A^2*a^10*d^10*f^2 - 96*A*B*C*a*b^7*c*d^7*f - 28*A*B*C*a^7*b*c*d^7*f - 28
*A*B*C*a*b^7*c^7*d*f + 484*A*B*C*a^4*b^4*c^4*d^4*f - 424*A*B*C*a^3*b^5*c^3*
d^5*f + 320*A*B*C*a^2*b^6*c^2*d^6*f - 176*A*B*C*a^6*b^2*c^2*d^6*f - 176*A*B
*C*a^2*b^6*c^6*d^2*f + 158*A*B*C*a^4*b^4*c^2*d^6*f + 158*A*B*C*a^2*b^6*c^4*
d^4*f - 136*A*B*C*a^5*b^3*c^5*d^3*f - 34*A*B*C*a^6*b^2*c^4*d^4*f - 34*A*B*C
*a^4*b^4*c^6*d^2*f + 28*A*B*C*a^5*b^3*c^3*d^5*f + 28*A*B*C*a^3*b^5*c^5*d^3*
f + 308*A*B*C*a^5*b^3*c*d^7*f + 308*A*B*C*a*b^7*c^5*d^3*f + 20*A*B*C*a^7*b*
c^3*d^5*f + 20*A*B*C*a^3*b^5*c^7*d*f + 30*B*C^2*a^7*b*c*d^7*f + 30*B*C^2*a*
b^7*c^7*d*f + 160*A^2*B*a*b^7*c*d^7*f - 2*A^2*B*a^7*b*c*d^7*f - 2*A^2*B*a*b
^7*c^7*d*f - 96*A*B*C*b^8*c^4*d^4*f + 34*A*B*C*b^8*c^6*d^2*f - 32*A*B*C*b^8
*c^2*d^6*f + 2*A*B*C*a^8*c^2*d^6*f - 96*A*B*C*a^4*b^4*d^8*f + 34*A*B*C*a^6*
b^2*d^8*f - 32*A*B*C*a^2*b^6*d^8*f + 2*A*B*C*a^2*b^6*c^8*f - 210*B*C^2*a^4*
b^4*c^4*d^4*f - 182*B^2*C*a^5*b^3*c^2*d^6*f - 182*B^2*C*a^2*b^6*c^5*d^3*f +
 180*B*C^2*a^5*b^3*c^5*d^3*f + 180*B*C^2*a^3*b^5*c^3*d^5*f - 166*B^2*C*a^5*
b^3*c^4*d^4*f - 166*B^2*C*a^4*b^4*c^5*d^3*f + 152*B*C^2*a^6*b^2*c^2*d^6*f +
 152*B*C^2*a^2*b^6*c^6*d^2*f - 112*B^2*C*a^3*b^5*c^2*d^6*f - 112*B^2*C*a^2*
b^6*c^3*d^5*f + 94*B^2*C*a^4*b^4*c^3*d^5*f + 94*B^2*C*a^3*b^5*c^4*d^4*f - 8
0*B*C^2*a^2*b^6*c^2*d^6*f + 66*B*C^2*a^5*b^3*c^3*d^5*f + 66*B*C^2*a^3*b^5*c
^5*d^3*f + 46*B^2*C*a^6*b^2*c^3*d^5*f + 46*B^2*C*a^3*b^5*c^6*d^2*f + 33*B*C
^2*a^6*b^2*c^4*d^4*f + 33*B*C^2*a^4*b^4*c^6*d^2*f + 24*B^2*C*a^6*b^2*c^5*d^
3*f + 24*B^2*C*a^5*b^3*c^6*d^2*f - 16*B*C^2*a^6*b^2*c^6*d^2*f - 15*B*C^2*a^

4*b^4*c^2*d^6*f - 15*B*C^2*a^2*b^6*c^4*d^4*f - 190*A^2*C*a^4*b^4*c^3*d^5*f - 190*A^2*C*a^3*b^5*c^4*d^4*f + 182*A^2*C*a^5*b^3*c^2*d^6*f + 182*A^2*C*a^2*b^6*c^5*d^3*f + 160*A^2*C*a^3*b^5*c^2*d^6*f + 160*A^2*C*a^2*b^6*c^3*d^5*f - 150*A*C^2*a^5*b^3*c^2*d^6*f - 150*A*C^2*a^2*b^6*c^5*d^3*f - 126*A*C^2*a^5*b^3*c^4*d^4*f - 126*A*C^2*a^4*b^4*c^5*d^3*f + 126*A*C^2*a^4*b^4*c^3*d^5*f + 126*A*C^2*a^3*b^5*c^4*d^4*f - 96*A*C^2*a^3*b^5*c^2*d^6*f - 96*A*C^2*a^2*b^6*c^3*d^5*f + 94*A^2*C*a^5*b^3*c^4*d^4*f + 94*A^2*C*a^4*b^4*c^5*d^3*f + 54*A*C^2*a^6*b^2*c^3*d^5*f + 54*A*C^2*a^3*b^5*c^6*d^2*f + 32*A*C^2*a^6*b^2*c^5*d^3*f + 32*A*C^2*a^5*b^3*c^6*d^2*f - 22*A^2*C*a^6*b^2*c^3*d^5*f - 22*A^2*C*a^3*b^5*c^6*d^2*f + 500*A^2*B*a^3*b^5*c^3*d^5*f - 290*A^2*B*a^4*b^4*c^4*d^4*f - 256*A^2*B*a^2*b^6*c^2*d^6*f - 230*A*B^2*a^4*b^4*c^3*d^5*f - 230*A*B^2*a^3*b^5*c^4*d^4*f + 142*A*B^2*a^5*b^3*c^2*d^6*f + 142*A*B^2*a^2*b^6*c^5*d^3*f - 127*A^2*B*a^4*b^4*c^2*d^6*f - 127*A^2*B*a^2*b^6*c^4*d^4*f + 86*A*B^2*a^5*b^3*c^4*d^4*f + 86*A*B^2*a^4*b^4*c^5*d^3*f + 80*A*B^2*a^3*b^5*c^2*d^6*f + 80*A*B^2*a^2*b^6*c^3*d^5*f + 40*A^2*B*a^6*b^2*c^2*d^6*f + 40*A^2*B*a^2*b^6*c^6*d^2*f + 34*A^2*B*a^5*b^3*c^3*d^5*f + 34*A^2*B*a^3*b^5*c^5*d^3*f - 30*A*B^2*a^6*b^2*c^3*d^5*f - 30*A*B^2*a^3*b^5*c^6*d^2*f + 20*A^2*B*a^5*b^3*c^5*d^3*f - 15*A^2*B*a^6*b^2*c^4*d^4*f - 15*A^2*B*a^4*b^4*c^6*d^2*f - 98*B^2*C*a^6*b^2*c*d^7*f - 98*B^2*C*a*b^7*c^6*d^2*f - 90*B*C^2*a^5*b^3*c*d^7*f - 90*B*C^2*a*b^7*c^5*d^3*f + 48*B^2*C*a^4*b^4*c*d^7*f + 48*B^2*C*a*b^7*c^4*d^4*f + 40*B^2*C*a^2*b^6*c*d^7*f + 40*B^2*C*a*b^7*c^2*d^6*f - 32*B*C^2*a^3*b^5*c*d^7*f - 32*B*C^2*a*b^7*c^3*d^5*f + 26*B^2*C*a^7*b*c^2*d^6*f + 26*B^2*C*a^2*b^6*c^7*d*f - 26*B*C^2*a^7*b*c^3*d^5*f - 26*B*C^2*a^3*b^5*c^7*d*f - 8*B^2*C*a^7*b*c^4*d^4*f - 8*B^2*C*a^4*b^4*c^7*d*f - 224*A^2*C*a^4*b^4*c*d^7*f - 224*A^2*C*a*b^7*c^4*d^4*f - 96*A^2*C*a^2*b^6*c*d^7*f - 96*A^2*C*a*b^7*c^2*d^6*f + 96*A*C^2*a^4*b^4*c*d^7*f + 96*A*C^2*a*b^7*c^4*d^4*f - 66*A*C^2*a^6*b^2*c*d^7*f - 66*A*C^2*a*b^7*c^6*d^2*f + 64*A*C^2*a^2*b^6*c*d^7*f + 64*A*C^2*a*b^7*c^2*d^6*f + 34*A^2*C*a^6*b^2*c*d^7*f + 34*A^2*C*a*b^7*c^6*d^2*f + 34*A*C^2*a^7*b*c^2*d^6*f + 34*A*C^2*a^2*b^6*c^7*d*f - 2*A^2*C*a^7*b*c^2*d^6*f - 2*A^2*C*a^2*b^6*c^7*d*f - 208*A*B^2*a^4*b^4*c*d^7*f - 208*A*B^2*a*b^7*c^4*d^4*f + 160*A^2*B*a^3*b^5*c*d^7*f + 160*A^2*B*a*b^7*c^3*d^5*f - 154*A^2*B*a^5*b^3*c*d^7*f - 154*A^2*B*a*b^7*c^5*d^3*f - 112*A*B^2*a^2*b^6*c*d^7*f - 112*A*B^2*a*b^7*c^2*d^6*f + 58*A*B^2*a^6*b^2*c*d^7*f + 58*A*B^2*a*b^7*c^6*d^2*f - 10*A*B^2*a^7*b*c^2*d^6*f - 10*A*B^2*a^2*b^6*c^7*d*f + 6*A^2*B*a^7*b*c^3*d^5*f + 6*A^2*B*a^3*b^5*c^7*d*f + 32*B^2*C*b^8*c^5*d^3*f - 17*B*C^2*b^8*c^6*d^2*f + 8*B^2*C*b^8*c^3*d^5*f + 64*A^2*C*b^8*c^3*d^5*f - 32*A^2*C*b^8*c^5*d^3*f + 32*A*C^2*b^8*c^5*d^3*f - B*C^2*a^8*c^2*d^6*f + 112*A^2*B*b^8*c^4*d^4*f - 64*A*B^2*b^8*c^5*d^3*f + 32*B^2*C*a^5*b^3*d^8*f - 17*B*C^2*a^6*b^2*d^8*f + 16*A^2*B*b^8*c^2*d^6*f + 16*A*B^2*b^8*c^3*d^5*f + 8*B^2*C*a^3*b^5*d^8*f - A^2*B*b^8*c^6*d^2*f + 64*A^2*C*a^3*b^5*d^8*f - 32*A^2*C*a^5*b^3*d^8*f + 32*A*C^2*a^5*b^3*d^8*f - A^2*B*a^8*c^2*d^6*f - B*C^2*a^2*b^6*c^8*f + 112*A^2*B*a^4*b^4*d^8*f - 64*A*B^2*a^5*b^3*d^8*f + 16*A^2*B*a^2*b^6*d^8*f + 16*A*B^2*a^3*b^5*d^8*f - A^2*B*a^6*b^2*d^8*f - A^2*B*a^2*b^6*c^8*f - 8*B^3*a*b^7*c*d^7*f - 2*B^3*a^7*b*c*d^7*f - 2*B^3*a*b^7*c^7*d*f - 6*B^2*C*b^8*c^7*d*f + 32*A^2*C*b^8*c*d^7*f + 6*A^2*C*b^8*c^7*d*f - 6*A*C^2*b^8*c^7*d*f

- 2*B^2*C*a^8*c*d^7*f + 16*A*B^2*b^8*c*d^7*f - 6*B^2*C*a^7*b*d^8*f - 6*A^2*
C*a^8*c*d^7*f + 6*A*C^2*a^8*c*d^7*f - 2*A*B^2*b^8*c^7*d*f + 32*A^2*C*a*b^7*
d^8*f + 6*A^2*C*a^7*b*d^8*f - 6*A*C^2*a^7*b*d^8*f - 2*B^2*C*a*b^7*c^8*f + 2
*A*B^2*a^8*c*d^7*f + 16*A*B^2*a*b^7*d^8*f - 6*A^2*C*a*b^7*c^8*f + 6*A*C^2*a
*b^7*c^8*f - 2*A*B^2*a^7*b*d^8*f + 2*A*B^2*a*b^7*c^8*f - 50*C^3*a^6*b^2*c^3
*d^5*f + 50*C^3*a^5*b^3*c^2*d^6*f - 50*C^3*a^3*b^5*c^6*d^2*f + 50*C^3*a^2*b
^6*c^5*d^3*f + 42*C^3*a^5*b^3*c^4*d^4*f + 42*C^3*a^4*b^4*c^5*d^3*f - 42*C^3
*a^4*b^4*c^3*d^5*f - 42*C^3*a^3*b^5*c^4*d^4*f - 32*C^3*a^6*b^2*c^5*d^3*f -
32*C^3*a^5*b^3*c^6*d^2*f + 32*C^3*a^3*b^5*c^2*d^6*f + 32*C^3*a^2*b^6*c^3*d^
5*f + 94*B^3*a^4*b^4*c^4*d^4*f + 48*B^3*a^2*b^6*c^2*d^6*f - 44*B^3*a^3*b^5*
c^3*d^5*f - 32*B^3*a^6*b^2*c^2*d^6*f - 32*B^3*a^2*b^6*c^6*d^2*f + 29*B^3*a^
4*b^4*c^2*d^6*f + 29*B^3*a^2*b^6*c^4*d^4*f - 20*B^3*a^5*b^3*c^5*d^3*f + 18*
B^3*a^5*b^3*c^3*d^5*f + 18*B^3*a^3*b^5*c^5*d^3*f - 3*B^3*a^6*b^2*c^4*d^4*f
- 3*B^3*a^4*b^4*c^6*d^2*f + 106*A^3*a^4*b^4*c^3*d^5*f + 106*A^3*a^3*b^5*c^4
*d^4*f - 96*A^3*a^3*b^5*c^2*d^6*f - 96*A^3*a^2*b^6*c^3*d^5*f - 82*A^3*a^5*b
^3*c^2*d^6*f - 82*A^3*a^2*b^6*c^5*d^3*f + 18*A^3*a^6*b^2*c^3*d^5*f + 18*A^3
*a^3*b^5*c^6*d^2*f - 10*A^3*a^5*b^3*c^4*d^4*f - 10*A^3*a^4*b^4*c^5*d^3*f -
22*C^3*a^7*b*c^2*d^6*f + 22*C^3*a^6*b^2*c*d^7*f - 22*C^3*a^2*b^6*c^7*d*f +
22*C^3*a*b^7*c^6*d^2*f - 2*A*B*C*b^8*c^8*f - 2*A*B*C*a^8*d^8*f + 62*B^3*a^5
*b^3*c*d^7*f + 62*B^3*a*b^7*c^5*d^3*f + 16*B^3*a^3*b^5*c*d^7*f + 16*B^3*a*b
^7*c^3*d^5*f + 6*B^3*a^7*b*c^3*d^5*f + 6*B^3*a^3*b^5*c^7*d*f + 128*A^3*a^4*
b^4*c*d^7*f + 128*A^3*a*b^7*c^4*d^4*f + 32*A^3*a^2*b^6*c*d^7*f + 32*A^3*a*b
^7*c^2*d^6*f - 10*A^3*a^7*b*c^2*d^6*f + 10*A^3*a^6*b^2*c*d^7*f - 10*A^3*a^2
*b^6*c^7*d*f + 10*A^3*a*b^7*c^6*d^2*f + 11*B^3*b^8*c^6*d^2*f - 8*B^3*b^8*c^
4*d^4*f - 4*B^3*b^8*c^2*d^6*f - 64*A^3*b^8*c^3*d^5*f - B^3*a^8*c^2*d^6*f +
11*B^3*a^6*b^2*d^8*f - 8*B^3*a^4*b^4*d^8*f - 4*B^3*a^2*b^6*d^8*f - 64*A^3*a
^3*b^5*d^8*f - B^3*a^2*b^6*c^8*f + 2*C^3*b^8*c^7*d*f - 2*C^3*a^8*c*d^7*f -
32*A^3*b^8*c*d^7*f + 2*C^3*a^7*b*d^8*f - 2*A^3*b^8*c^7*d*f - 2*C^3*a*b^7*c^
8*f + 2*A^3*a^8*c*d^7*f - 32*A^3*a*b^7*d^8*f - 2*A^3*a^7*b*d^8*f + 2*A^3*a*
b^7*c^8*f - 16*A^2*B*b^8*d^8*f + B*C^2*b^8*c^8*f + B*C^2*a^8*d^8*f + A^2*B*
b^8*c^8*f + A^2*B*a^8*d^8*f + B^3*b^8*c^8*f + B^3*a^8*d^8*f - 4*A*B^2*C*a^5
*b*c*d^5 - 4*A*B^2*C*a*b^5*c^5*d + 4*A*B^2*C*a*b^5*c*d^5 + 22*A^2*B*C*a^3*b
^3*c^2*d^4 + 22*A^2*B*C*a^2*b^4*c^3*d^3 - 20*A*B^2*C*a^3*b^3*c^3*d^3 + 14*A
*B^2*C*a^4*b^2*c^2*d^4 + 14*A*B^2*C*a^2*b^4*c^4*d^2 - 14*A*B*C^2*a^3*b^3*c^
2*d^4 - 14*A*B*C^2*a^2*b^4*c^3*d^3 + 12*A*B*C^2*a^4*b^2*c^3*d^3 + 12*A*B*C^
2*a^3*b^3*c^4*d^2 - 6*A^2*B*C*a^4*b^2*c^3*d^3 - 6*A^2*B*C*a^3*b^3*c^4*d^2 -
4*A*B^2*C*a^2*b^4*c^2*d^4 + 22*A*B*C^2*a^4*b^2*c*d^5 + 22*A*B*C^2*a*b^5*c^
4*d^2 - 20*A^2*B*C*a^4*b^2*c*d^5 - 20*A^2*B*C*a*b^5*c^4*d^2 + 10*A*B*C^2*a^
2*b^4*c*d^5 + 10*A*B*C^2*a*b^5*c^2*d^4 - 8*A^2*B*C*a^2*b^4*c*d^5 - 8*A^2*B*
C*a*b^5*c^2*d^4 + 4*A*B^2*C*a^3*b^3*c*d^5 + 4*A*B^2*C*a*b^5*c^3*d^3 - 4*A*B
*C^2*a^5*b*c^2*d^4 - 4*A*B*C^2*a^2*b^4*c^5*d + 2*A^2*B*C*a^5*b*c^2*d^4 + 2*
A^2*B*C*a^2*b^4*c^5*d - 8*B^3*C*a^4*b^2*c*d^5 - 8*B^3*C*a*b^5*c^4*d^2 - 8*B
*C^3*a^4*b^2*c*d^5 - 8*B*C^3*a*b^5*c^4*d^2 - 4*B^3*C*a^2*b^4*c*d^5 - 4*B^3*
C*a*b^5*c^2*d^4 + 4*B^2*C^2*a^5*b*c*d^5 + 4*B^2*C^2*a*b^5*c^5*d - 4*B*C^3*a
^2*b^4*c*d^5 - 4*B*C^3*a*b^5*c^2*d^4 + 2*B^3*C*a^5*b*c^2*d^4 + 2*B^3*C*a^2*

b^4*c^5*d + 2*B^2*C^2*a*b^5*c*d^5 + 2*B*C^3*a^5*b*c^2*d^4 + 2*B*C^3*a^2*b^4
*c^5*d + 24*A^3*C*a^3*b^3*c*d^5 + 24*A^3*C*a*b^5*c^3*d^3 - 24*A^2*C^2*a*b^5
*c*d^5 + 12*A^2*C^2*a^5*b*c*d^5 + 12*A^2*C^2*a*b^5*c^5*d + 8*A*C^3*a^3*b^3*
c*d^5 + 8*A*C^3*a*b^5*c^3*d^3 + 6*A^3*B*a^4*b^2*c*d^5 + 6*A^3*B*a*b^5*c^4*d
^2 - 6*A^2*B^2*a*b^5*c*d^5 + 6*A*B^3*a^4*b^2*c*d^5 + 6*A*B^3*a*b^5*c^4*d^2
+ 2*A^3*B*a^2*b^4*c*d^5 + 2*A^3*B*a*b^5*c^2*d^4 + 2*A*B^3*a^2*b^4*c*d^5 + 2
*A*B^3*a*b^5*c^2*d^4 + 20*A^2*B*C*b^6*c^3*d^3 - 10*A*B*C^2*b^6*c^3*d^3 - 2*
A*B^2*C*b^6*c^4*d^2 - 2*A*B^2*C*b^6*c^2*d^4 + 20*A^2*B*C*a^3*b^3*d^6 - 10*A
*B*C^2*a^3*b^3*d^6 - 2*A*B^2*C*a^4*b^2*d^6 - 2*A*B^2*C*a^2*b^4*d^6 + 10*B^2
*C^2*a^3*b^3*c^3*d^3 + 4*B^2*C^2*a^4*b^2*c^4*d^2 - 3*B^2*C^2*a^4*b^2*c^2*d^
4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 2*B^2*C^2*a^2*b^4*c^2*d^4 + 40*A^2*C^2*a^2*
b^4*c^2*d^4 - 16*A^2*C^2*a^4*b^2*c^2*d^4 - 16*A^2*C^2*a^2*b^4*c^4*d^2 + 4*A
^2*C^2*a^4*b^2*c^4*d^2 + 18*A^2*B^2*a^2*b^4*c^2*d^4 + 10*A^2*B^2*a^3*b^3*c^
3*d^3 - 3*A^2*B^2*a^4*b^2*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 24*A^3*C*a*
b^5*c*d^5 - 12*A*C^3*a^5*b*c*d^5 - 12*A*C^3*a*b^5*c^5*d + 8*A*C^3*a*b^5*c*d
^5 - 4*A^3*C*a^5*b*c*d^5 - 4*A^3*C*a*b^5*c^5*d + 8*A^2*B*C*b^6*c*d^5 + 4*A*
B*C^2*b^6*c^5*d - 4*A*B*C^2*b^6*c*d^5 - 2*A^2*B*C*b^6*c^5*d + 8*A^2*B*C*a*b
^5*d^6 + 4*A*B*C^2*a^5*b*d^6 - 4*A*B*C^2*a*b^5*d^6 - 2*A^2*B*C*a^5*b*d^6 -
6*B^3*C*a^4*b^2*c^3*d^3 - 6*B^3*C*a^3*b^3*c^4*d^2 - 6*B*C^3*a^4*b^2*c^3*d^3
- 6*B*C^3*a^3*b^3*c^4*d^2 + 2*B^3*C*a^3*b^3*c^2*d^4 + 2*B^3*C*a^2*b^4*c^3*
d^3 + 2*B^2*C^2*a^3*b^3*c*d^5 + 2*B^2*C^2*a*b^5*c^3*d^3 + 2*B*C^3*a^3*b^3*c
^2*d^4 + 2*B*C^3*a^2*b^4*c^3*d^3 - 48*A^3*C*a^2*b^4*c^2*d^4 - 24*A^2*C^2*a^
3*b^3*c*d^5 - 24*A^2*C^2*a*b^5*c^3*d^3 - 16*A*C^3*a^2*b^4*c^2*d^4 + 8*A^3*C
*a^4*b^2*c^2*d^4 + 8*A^3*C*a^2*b^4*c^4*d^2 - 8*A*C^3*a^4*b^2*c^4*d^2 + 8*A*
C^3*a^4*b^2*c^2*d^4 + 8*A*C^3*a^2*b^4*c^4*d^2 - 10*A^3*B*a^3*b^3*c^2*d^4 -
10*A^3*B*a^2*b^4*c^3*d^3 - 10*A*B^3*a^3*b^3*c^2*d^4 - 10*A*B^3*a^2*b^4*c^3*
d^3 - 6*A^2*B^2*a^3*b^3*c*d^5 - 6*A^2*B^2*a*b^5*c^3*d^3 + 3*B^2*C^2*b^6*c^4
*d^2 - 8*A^2*C^2*b^6*c^4*d^2 + 8*A^2*C^2*b^6*c^2*d^4 + 9*A^2*B^2*b^6*c^2*d^
4 + 3*B^2*C^2*a^4*b^2*d^6 + 3*A^2*B^2*b^6*c^4*d^2 - 8*A^2*C^2*a^4*b^2*d^6 +
8*A^2*C^2*a^2*b^4*d^6 + 9*A^2*B^2*a^2*b^4*d^6 + 3*A^2*B^2*a^4*b^2*d^6 + 2*
B^4*a^3*b^3*c*d^5 + 2*B^4*a*b^5*c^3*d^3 - 8*A^4*a^3*b^3*c*d^5 - 8*A^4*a*b^5
*c^3*d^3 - 16*A^3*C*b^6*c^2*d^4 + 4*A^3*C*b^6*c^4*d^2 + 4*A*C^3*b^6*c^4*d^2
- 10*A^3*B*b^6*c^3*d^3 - 10*A*B^3*b^6*c^3*d^3 - 16*A^3*C*a^2*b^4*d^6 + 4*A
^3*C*a^4*b^2*d^6 + 4*A*C^3*a^4*b^2*d^6 - 10*A^3*B*a^3*b^3*d^6 - 10*A*B^3*a^
3*b^3*d^6 + 4*C^4*a^5*b*c*d^5 + 4*C^4*a*b^5*c^5*d + 2*B^4*a*b^5*c*d^5 - 8*A
^4*a*b^5*c*d^5 - 2*B^3*C*b^6*c^5*d - 2*B*C^3*b^6*c^5*d - 4*A^3*B*b^6*c*d^5
- 4*A*B^3*b^6*c*d^5 - 2*B^3*C*a^5*b*d^6 - 2*B*C^3*a^5*b*d^6 - 4*A^3*B*a*b^5
*d^6 - 4*A*B^3*a*b^5*d^6 + 4*C^4*a^4*b^2*c^4*d^2 + 4*C^4*a^2*b^4*c^2*d^4 +
10*B^4*a^3*b^3*c^3*d^3 - 3*B^4*a^4*b^2*c^2*d^4 - 3*B^4*a^2*b^4*c^4*d^2 - 2*
B^4*a^2*b^4*c^2*d^4 + 20*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*b^6*c^2*d^4 + B^2*C^
2*a^2*b^4*d^6 - 8*A^3*C*b^6*d^6 + 3*B^4*b^6*c^4*d^2 + 8*A^4*b^6*c^2*d^4 + 3
*B^4*a^4*b^2*d^6 + 8*A^4*a^2*b^4*d^6 + 4*A^2*C^2*b^6*d^6 + 4*A^2*B^2*b^6*d^
6 + 4*A^4*b^6*d^6 + B^4*b^6*c^2*d^4 + B^4*a^2*b^4*d^6, f, k)*((16*A^2*a^3*b
^6*d^9 + A^2*a^5*b^4*d^9 + 2*A^2*a^7*b^2*d^9 + 8*B^2*a^3*b^6*d^9 + 9*B^2*a^
5*b^4*d^9 + 16*A^2*b^9*c^3*d^6 + A^2*b^9*c^5*d^4 + 2*A^2*b^9*c^7*d^2 + C^2*

a^5*b^4*d^9 + 2*C^2*a^7*b^2*d^9 + 8*B^2*b^9*c^3*d^6 + 9*B^2*b^9*c^5*d^4 + C
^2*b^9*c^5*d^4 + 2*C^2*b^9*c^7*d^2 + 16*A^2*a*b^8*d^9 + 16*A^2*b^9*c*d^8 -
14*A^2*a^2*b^7*c^3*d^6 - 4*A^2*a^2*b^7*c^5*d^4 + 6*A^2*a^2*b^7*c^7*d^2 - 14
*A^2*a^3*b^6*c^2*d^7 - 6*A^2*a^3*b^6*c^4*d^5 - 12*A^2*a^3*b^6*c^6*d^3 - 6*A
^2*a^4*b^5*c^3*d^6 + 7*A^2*a^4*b^5*c^5*d^4 - 4*A^2*a^5*b^4*c^2*d^7 + 7*A^2*
a^5*b^4*c^4*d^5 - 12*A^2*a^6*b^3*c^3*d^6 + 6*A^2*a^7*b^2*c^2*d^7 + 18*B^2*a
^2*b^7*c^3*d^6 + 2*B^2*a^2*b^7*c^5*d^4 - 4*B^2*a^2*b^7*c^7*d^2 + 18*B^2*a^3
*b^6*c^2*d^7 - 20*B^2*a^3*b^6*c^4*d^5 + 6*B^2*a^3*b^6*c^6*d^3 - 20*B^2*a^4*
b^5*c^3*d^6 - 19*B^2*a^4*b^5*c^5*d^4 + 2*B^2*a^5*b^4*c^2*d^7 - 19*B^2*a^5*b
^4*c^4*d^5 + 6*B^2*a^6*b^3*c^3*d^6 - 4*B^2*a^7*b^2*c^2*d^7 + 2*C^2*a^2*b^7*
c^3*d^6 + 12*C^2*a^2*b^7*c^5*d^4 + 6*C^2*a^2*b^7*c^7*d^2 + 2*C^2*a^3*b^6*c^
2*d^7 + 10*C^2*a^3*b^6*c^4*d^5 - 28*C^2*a^3*b^6*c^6*d^3 + 10*C^2*a^4*b^5*c^
3*d^6 + 7*C^2*a^4*b^5*c^5*d^4 + 12*C^2*a^5*b^4*c^2*d^7 + 7*C^2*a^5*b^4*c^4*
d^5 - 16*C^2*a^5*b^4*c^6*d^3 - 28*C^2*a^6*b^3*c^3*d^6 - 16*C^2*a^6*b^3*c^5*
d^4 + 6*C^2*a^7*b^2*c^2*d^7 - 24*A*B*a^2*b^7*d^9 - 24*A*B*a^4*b^5*d^9 + A*B
*a^6*b^3*d^9 + 16*A*C*a^3*b^6*d^9 + 14*A*C*a^5*b^4*d^9 - 4*A*C*a^7*b^2*d^9
- 24*A*B*b^9*c^2*d^7 - 24*A*B*b^9*c^4*d^5 + A*B*b^9*c^6*d^3 - 8*B*C*a^4*b^5
*d^9 - 9*B*C*a^6*b^3*d^9 + 16*A*C*b^9*c^3*d^6 + 14*A*C*b^9*c^5*d^4 - 4*A*C*
b^9*c^7*d^2 - 8*B*C*b^9*c^4*d^5 - 9*B*C*b^9*c^6*d^3 - A^2*a*b^8*c^8*d - A^2
*a^8*b*c*d^8 + B^2*a*b^8*c^8*d + B^2*a^8*b*c*d^8 - C^2*a*b^8*c^8*d - C^2*a^
8*b*c*d^8 - 3*A^2*a*b^8*c^4*d^5 - 8*A^2*a*b^8*c^6*d^3 - 3*A^2*a^4*b^5*c*d^8
- 8*A^2*a^6*b^3*c*d^8 + 8*B^2*a*b^8*c^2*d^7 - 11*B^2*a*b^8*c^4*d^5 + 2*B^2
*a*b^8*c^6*d^3 + 8*B^2*a^2*b^7*c*d^8 - 11*B^2*a^4*b^5*c*d^8 + 2*B^2*a^6*b^3
*c*d^8 + 13*C^2*a*b^8*c^4*d^5 - 8*C^2*a*b^8*c^6*d^3 + 13*C^2*a^4*b^5*c*d^8
- 8*C^2*a^6*b^3*c*d^8 - A*B*a^8*b*d^9 - A*B*b^9*c^8*d + B*C*a^8*b*d^9 + B*C
*b^9*c^8*d - 16*A*B*a*b^8*c*d^8 + 2*A*C*a*b^8*c^8*d + 2*A*C*a^8*b*c*d^8 + 2
4*A*B*a*b^8*c^3*d^6 + 2*A*B*a*b^8*c^5*d^4 + 2*A*B*a*b^8*c^7*d^2 + A*B*a^2*b
^7*c^8*d + 24*A*B*a^3*b^6*c*d^8 + 2*A*B*a^5*b^4*c*d^8 + 2*A*B*a^7*b^2*c*d^8
+ A*B*a^8*b*c^2*d^7 + 16*A*C*a*b^8*c^2*d^7 - 26*A*C*a*b^8*c^4*d^5 + 16*A*C
*a^2*b^7*c*d^8 - 26*A*C*a^4*b^5*c*d^8 - 24*B*C*a*b^8*c^3*d^6 + 14*B*C*a*b^8
*c^5*d^4 - 2*B*C*a*b^8*c^7*d^2 - B*C*a^2*b^7*c^8*d - 24*B*C*a^3*b^6*c*d^8 +
14*B*C*a^5*b^4*c*d^8 - 2*B*C*a^7*b^2*c*d^8 - B*C*a^8*b*c^2*d^7 - 64*A*B*a^
2*b^7*c^2*d^7 - 25*A*B*a^2*b^7*c^4*d^5 + 8*A*B*a^2*b^7*c^6*d^3 + 108*A*B*a^
3*b^6*c^3*d^6 + 6*A*B*a^3*b^6*c^5*d^4 - 6*A*B*a^3*b^6*c^7*d^2 - 25*A*B*a^4*
b^5*c^2*d^7 + 34*A*B*a^4*b^5*c^4*d^5 + 15*A*B*a^4*b^5*c^6*d^3 + 6*A*B*a^5*b
^4*c^3*d^6 - 20*A*B*a^5*b^4*c^5*d^4 + 8*A*B*a^6*b^3*c^2*d^7 + 15*A*B*a^6*b^
3*c^4*d^5 - 6*A*B*a^7*b^2*c^3*d^6 + 44*A*C*a^2*b^7*c^3*d^6 + 8*A*C*a^2*b^7*
c^5*d^4 - 12*A*C*a^2*b^7*c^7*d^2 + 44*A*C*a^3*b^6*c^2*d^7 - 36*A*C*a^3*b^6*
c^4*d^5 + 8*A*C*a^3*b^6*c^6*d^3 - 36*A*C*a^4*b^5*c^3*d^6 - 30*A*C*a^4*b^5*c
^5*d^4 + 8*A*C*a^5*b^4*c^2*d^7 - 30*A*C*a^5*b^4*c^4*d^5 + 8*A*C*a^6*b^3*c^3
*d^6 - 12*A*C*a^7*b^2*c^2*d^7 - 15*B*C*a^2*b^7*c^4*d^5 - 8*B*C*a^2*b^7*c^6*
d^3 - 44*B*C*a^3*b^6*c^3*d^6 + 58*B*C*a^3*b^6*c^5*d^4 + 6*B*C*a^3*b^6*c^7*d
^2 - 15*B*C*a^4*b^5*c^2*d^7 - 34*B*C*a^4*b^5*c^4*d^5 - 7*B*C*a^4*b^5*c^6*d^
3 + 58*B*C*a^5*b^4*c^3*d^6 + 68*B*C*a^5*b^4*c^5*d^4 - 8*B*C*a^6*b^3*c^2*d^7
- 7*B*C*a^6*b^3*c^4*d^5 + 6*B*C*a^7*b^2*c^3*d^6)/(a^8*d^8 + b^8*c^8 + 2*a^

2*b^6*c^8 + a^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8
*c^4*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3
- 4*a^3*b^5*c*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8
*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 +
10*a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*
c^2*d^6 + 26*a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16
*a^5*b^3*c^5*d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*
d^2 - 4*a*b^7*c^7*d - 4*a^7*b*c*d^7) + root(144*a^13*b*c^5*d^9*f^4 + 144*a^
9*b^5*c*d^13*f^4 + 144*a^5*b^9*c^13*d*f^4 + 144*a*b^13*c^9*d^5*f^4 + 96*a^1
3*b*c^7*d^7*f^4 + 96*a^13*b*c^3*d^11*f^4 + 96*a^11*b^3*c*d^13*f^4 + 96*a^7*
b^7*c^13*d*f^4 + 96*a^7*b^7*c*d^13*f^4 + 96*a^3*b^11*c^13*d*f^4 + 96*a*b^13
*c^11*d^3*f^4 + 96*a*b^13*c^7*d^7*f^4 + 24*a^13*b*c^9*d^5*f^4 + 24*a^9*b^5*
c^13*d*f^4 + 24*a^5*b^9*c*d^13*f^4 + 24*a*b^13*c^5*d^9*f^4 + 24*a^13*b*c*d^
13*f^4 + 24*a*b^13*c^13*d*f^4 + 3648*a^7*b^7*c^7*d^7*f^4 - 3188*a^8*b^6*c^6
*d^8*f^4 - 3188*a^6*b^8*c^8*d^6*f^4 - 2912*a^8*b^6*c^8*d^6*f^4 - 2912*a^6*b
^8*c^6*d^8*f^4 + 2592*a^9*b^5*c^7*d^7*f^4 + 2592*a^7*b^7*c^9*d^5*f^4 + 2592
*a^7*b^7*c^5*d^9*f^4 + 2592*a^5*b^9*c^7*d^7*f^4 + 2168*a^9*b^5*c^5*d^9*f^4
+ 2168*a^5*b^9*c^9*d^5*f^4 - 1776*a^10*b^4*c^6*d^8*f^4 - 1776*a^8*b^6*c^4*d
^10*f^4 - 1776*a^6*b^8*c^10*d^4*f^4 - 1776*a^4*b^10*c^8*d^6*f^4 + 1568*a^9*
b^5*c^9*d^5*f^4 + 1568*a^5*b^9*c^5*d^9*f^4 - 1344*a^10*b^4*c^8*d^6*f^4 - 13
44*a^8*b^6*c^10*d^4*f^4 - 1344*a^6*b^8*c^4*d^10*f^4 - 1344*a^4*b^10*c^6*d^8
*f^4 - 1164*a^10*b^4*c^4*d^10*f^4 - 1164*a^4*b^10*c^10*d^4*f^4 + 896*a^11*b
^3*c^5*d^9*f^4 + 896*a^9*b^5*c^3*d^11*f^4 + 896*a^5*b^9*c^11*d^3*f^4 + 896*
a^3*b^11*c^9*d^5*f^4 + 864*a^11*b^3*c^7*d^7*f^4 + 864*a^7*b^7*c^11*d^3*f^4
+ 864*a^7*b^7*c^3*d^11*f^4 + 864*a^3*b^11*c^7*d^7*f^4 - 480*a^10*b^4*c^10*d
^4*f^4 - 480*a^4*b^10*c^4*d^10*f^4 + 464*a^11*b^3*c^3*d^11*f^4 + 464*a^3*b^
11*c^11*d^3*f^4 - 424*a^12*b^2*c^6*d^8*f^4 - 424*a^8*b^6*c^2*d^12*f^4 - 424
*a^6*b^8*c^12*d^2*f^4 - 424*a^2*b^12*c^8*d^6*f^4 + 416*a^11*b^3*c^9*d^5*f^4
+ 416*a^9*b^5*c^11*d^3*f^4 + 416*a^5*b^9*c^3*d^11*f^4 + 416*a^3*b^11*c^5*d
^9*f^4 - 336*a^12*b^2*c^4*d^10*f^4 - 336*a^10*b^4*c^2*d^12*f^4 - 336*a^4*b^
10*c^12*d^2*f^4 - 336*a^2*b^12*c^10*d^4*f^4 - 256*a^12*b^2*c^8*d^6*f^4 - 25
6*a^8*b^6*c^12*d^2*f^4 - 256*a^6*b^8*c^2*d^12*f^4 - 256*a^2*b^12*c^6*d^8*f^
4 - 124*a^12*b^2*c^2*d^12*f^4 - 124*a^2*b^12*c^12*d^2*f^4 + 80*a^11*b^3*c^1
1*d^3*f^4 + 80*a^3*b^11*c^3*d^11*f^4 - 60*a^12*b^2*c^10*d^4*f^4 - 60*a^10*b
^4*c^12*d^2*f^4 - 60*a^4*b^10*c^2*d^12*f^4 - 60*a^2*b^12*c^4*d^10*f^4 - 24*
b^14*c^10*d^4*f^4 - 16*b^14*c^12*d^2*f^4 - 16*b^14*c^8*d^6*f^4 - 4*b^14*c^6
*d^8*f^4 - 24*a^14*c^4*d^10*f^4 - 16*a^14*c^6*d^8*f^4 - 16*a^14*c^2*d^12*f^
4 - 4*a^14*c^8*d^6*f^4 - 24*a^10*b^4*d^14*f^4 - 16*a^12*b^2*d^14*f^4 - 16*a
^8*b^6*d^14*f^4 - 4*a^6*b^8*d^14*f^4 - 24*a^4*b^10*c^14*f^4 - 16*a^6*b^8*c^
14*f^4 - 16*a^2*b^12*c^14*f^4 - 4*a^8*b^6*c^14*f^4 - 4*b^14*c^14*f^4 - 4*a^
14*d^14*f^4 + 36*A*C*a^9*b*c*d^9*f^2 + 36*A*C*a*b^9*c^9*d*f^2 + 32*A*C*a*b^
9*c*d^9*f^2 - 552*B*C*a^7*b^3*c^4*d^6*f^2 - 552*B*C*a^4*b^6*c^7*d^3*f^2 - 4
08*B*C*a^5*b^5*c^4*d^6*f^2 - 408*B*C*a^4*b^6*c^5*d^5*f^2 + 360*B*C*a^6*b^4*
c^3*d^7*f^2 + 360*B*C*a^3*b^7*c^6*d^4*f^2 - 248*B*C*a^7*b^3*c^2*d^8*f^2 - 2
48*B*C*a^2*b^8*c^7*d^3*f^2 + 184*B*C*a^6*b^4*c^5*d^5*f^2 + 184*B*C*a^5*b^5*

c^6*d^4*f^2 + 152*B*C*a^8*b^2*c^3*d^7*f^2 - 152*B*C*a^5*b^5*c^2*d^8*f^2 + 1
52*B*C*a^3*b^7*c^8*d^2*f^2 - 152*B*C*a^2*b^8*c^5*d^5*f^2 - 104*B*C*a^7*b^3*
c^6*d^4*f^2 - 104*B*C*a^6*b^4*c^7*d^3*f^2 + 64*B*C*a^8*b^2*c^5*d^5*f^2 + 64
*B*C*a^5*b^5*c^8*d^2*f^2 - 56*B*C*a^4*b^6*c^3*d^7*f^2 - 56*B*C*a^3*b^7*c^4*
d^6*f^2 - 24*B*C*a^8*b^2*c^7*d^3*f^2 - 24*B*C*a^7*b^3*c^8*d^2*f^2 - 24*B*C*
a^3*b^7*c^2*d^8*f^2 - 24*B*C*a^2*b^8*c^3*d^7*f^2 - 696*A*C*a^5*b^5*c^5*d^5*
f^2 + 536*A*C*a^6*b^4*c^6*d^4*f^2 + 536*A*C*a^6*b^4*c^4*d^6*f^2 + 536*A*C*a
^4*b^6*c^6*d^4*f^2 + 472*A*C*a^4*b^6*c^4*d^6*f^2 - 232*A*C*a^7*b^3*c^5*d^5*
f^2 - 232*A*C*a^5*b^5*c^7*d^3*f^2 + 216*A*C*a^3*b^7*c^3*d^7*f^2 + 168*A*C*a
^7*b^3*c^3*d^7*f^2 + 168*A*C*a^3*b^7*c^7*d^3*f^2 - 154*A*C*a^8*b^2*c^2*d^8*
f^2 - 154*A*C*a^2*b^8*c^8*d^2*f^2 + 62*A*C*a^8*b^2*c^6*d^4*f^2 + 62*A*C*a^6
*b^4*c^8*d^2*f^2 - 40*A*C*a^7*b^3*c^7*d^3*f^2 - 40*A*C*a^5*b^5*c^3*d^7*f^2
- 40*A*C*a^3*b^7*c^5*d^5*f^2 + 32*A*C*a^6*b^4*c^2*d^8*f^2 + 32*A*C*a^2*b^8*
c^6*d^4*f^2 - 32*A*C*a^2*b^8*c^2*d^8*f^2 + 30*A*C*a^4*b^6*c^2*d^8*f^2 + 30*
A*C*a^2*b^8*c^4*d^6*f^2 + 16*A*C*a^8*b^2*c^4*d^6*f^2 + 16*A*C*a^4*b^6*c^8*d
^2*f^2 - 488*A*B*a^6*b^4*c^3*d^7*f^2 - 488*A*B*a^3*b^7*c^6*d^4*f^2 + 440*A*
B*a^7*b^3*c^4*d^6*f^2 + 440*A*B*a^4*b^6*c^7*d^3*f^2 - 360*A*B*a^6*b^4*c^5*d
^5*f^2 - 360*A*B*a^5*b^5*c^6*d^4*f^2 - 192*A*B*a^8*b^2*c^3*d^7*f^2 - 192*A*
B*a^3*b^7*c^8*d^2*f^2 - 168*A*B*a^3*b^7*c^2*d^8*f^2 - 168*A*B*a^2*b^8*c^3*d
^7*f^2 - 152*A*B*a^4*b^6*c^3*d^7*f^2 - 152*A*B*a^3*b^7*c^4*d^6*f^2 - 120*A*
B*a^8*b^2*c^5*d^5*f^2 + 120*A*B*a^7*b^3*c^2*d^8*f^2 - 120*A*B*a^5*b^5*c^8*d
^2*f^2 + 120*A*B*a^5*b^5*c^4*d^6*f^2 - 120*A*B*a^5*b^5*c^2*d^8*f^2 + 120*A*
B*a^4*b^6*c^5*d^5*f^2 + 120*A*B*a^2*b^8*c^7*d^3*f^2 - 120*A*B*a^2*b^8*c^5*d
^5*f^2 + 40*A*B*a^7*b^3*c^6*d^4*f^2 + 40*A*B*a^6*b^4*c^7*d^3*f^2 - 72*B*C*a
^9*b*c^4*d^6*f^2 - 72*B*C*a^4*b^6*c^9*d*f^2 - 64*B*C*a^4*b^6*c*d^9*f^2 - 64
*B*C*a*b^9*c^4*d^6*f^2 - 32*B*C*a^8*b^2*c*d^9*f^2 - 32*B*C*a*b^9*c^8*d^2*f^
2 - 16*B*C*a^2*b^8*c*d^9*f^2 - 16*B*C*a*b^9*c^2*d^8*f^2 + 8*B*C*a^9*b*c^6*d
^4*f^2 - 8*B*C*a^9*b*c^2*d^8*f^2 + 8*B*C*a^6*b^4*c^9*d*f^2 - 8*B*C*a^2*b^8*
c^9*d*f^2 + 104*A*C*a^7*b^3*c*d^9*f^2 + 104*A*C*a*b^9*c^7*d^3*f^2 + 96*A*C*
a^3*b^7*c*d^9*f^2 + 96*A*C*a*b^9*c^3*d^7*f^2 + 72*A*C*a^9*b*c^3*d^7*f^2 + 7
2*A*C*a^3*b^7*c^9*d*f^2 + 68*A*C*a^5*b^5*c*d^9*f^2 + 68*A*C*a*b^9*c^5*d^5*f
^2 - 28*A*C*a^9*b*c^5*d^5*f^2 - 28*A*C*a^5*b^5*c^9*d*f^2 + 80*A*B*a^9*b*c^4
*d^6*f^2 + 80*A*B*a^4*b^6*c^9*d*f^2 + 24*A*B*a^8*b^2*c*d^9*f^2 - 24*A*B*a^6
*b^4*c*d^9*f^2 + 24*A*B*a^4*b^6*c*d^9*f^2 - 24*A*B*a^2*b^8*c*d^9*f^2 + 24*A
*B*a*b^9*c^8*d^2*f^2 - 24*A*B*a*b^9*c^6*d^4*f^2 + 24*A*B*a*b^9*c^4*d^6*f^2
- 24*A*B*a*b^9*c^2*d^8*f^2 - 32*B*C*b^10*c^7*d^3*f^2 - 8*B*C*b^10*c^5*d^5*f
^2 + 34*A*C*b^10*c^6*d^4*f^2 + 16*B*C*a^10*c^3*d^7*f^2 + 16*A*C*b^10*c^4*d^
6*f^2 - 12*A*C*b^10*c^8*d^2*f^2 - 96*A*B*b^10*c^5*d^5*f^2 - 72*A*B*b^10*c^3
*d^7*f^2 - 32*B*C*a^7*b^3*d^10*f^2 - 28*A*C*a^10*c^2*d^8*f^2 - 24*A*B*b^10*
c^7*d^3*f^2 - 8*B*C*a^5*b^5*d^10*f^2 + 2*A*C*a^10*c^4*d^6*f^2 + 34*A*C*a^6*
b^4*d^10*f^2 + 16*B*C*a^3*b^7*c^10*f^2 + 16*A*C*a^4*b^6*d^10*f^2 - 16*A*B*a
^10*c^3*d^7*f^2 - 12*A*C*a^8*b^2*d^10*f^2 - 96*A*B*a^5*b^5*d^10*f^2 - 72*A*
B*a^3*b^7*d^10*f^2 - 28*A*C*a^2*b^8*c^10*f^2 - 24*A*B*a^7*b^3*d^10*f^2 + 2*
A*C*a^4*b^6*c^10*f^2 - 16*A*B*a^3*b^7*c^10*f^2 + 444*C^2*a^5*b^5*c^5*d^5*f^
2 + 148*C^2*a^7*b^3*c^5*d^5*f^2 + 148*C^2*a^5*b^5*c^7*d^3*f^2 + 148*C^2*a^5

*b^5*c^3*d^7*f^2 + 148*C^2*a^3*b^7*c^5*d^5*f^2 - 140*C^2*a^6*b^4*c^6*d^4*f^
2 - 140*C^2*a^6*b^4*c^4*d^6*f^2 - 140*C^2*a^4*b^6*c^6*d^4*f^2 - 140*C^2*a^4
*b^6*c^4*d^6*f^2 + 109*C^2*a^8*b^2*c^2*d^8*f^2 + 109*C^2*a^2*b^8*c^8*d^2*f^
2 + 48*C^2*a^8*b^2*c^4*d^6*f^2 + 48*C^2*a^6*b^4*c^2*d^8*f^2 + 48*C^2*a^4*b^
6*c^8*d^2*f^2 + 48*C^2*a^2*b^8*c^6*d^4*f^2 + 20*C^2*a^7*b^3*c^7*d^3*f^2 - 2
0*C^2*a^7*b^3*c^3*d^7*f^2 - 20*C^2*a^3*b^7*c^7*d^3*f^2 + 20*C^2*a^3*b^7*c^3
*d^7*f^2 + 17*C^2*a^8*b^2*c^6*d^4*f^2 + 17*C^2*a^6*b^4*c^8*d^2*f^2 + 17*C^2
*a^4*b^6*c^2*d^8*f^2 + 17*C^2*a^2*b^8*c^4*d^6*f^2 + 16*C^2*a^8*b^2*c^8*d^2*
f^2 + 16*C^2*a^2*b^8*c^2*d^8*f^2 - 396*B^2*a^5*b^5*c^5*d^5*f^2 + 308*B^2*a^
6*b^4*c^4*d^6*f^2 + 308*B^2*a^4*b^6*c^6*d^4*f^2 + 300*B^2*a^4*b^6*c^4*d^6*f
^2 + 284*B^2*a^6*b^4*c^6*d^4*f^2 - 132*B^2*a^7*b^3*c^5*d^5*f^2 - 132*B^2*a^
5*b^5*c^7*d^3*f^2 - 84*B^2*a^5*b^5*c^3*d^7*f^2 - 84*B^2*a^3*b^7*c^5*d^5*f^2
 + 61*B^2*a^4*b^6*c^2*d^8*f^2 + 61*B^2*a^2*b^8*c^4*d^6*f^2 - 59*B^2*a^8*b^2
*c^2*d^8*f^2 - 59*B^2*a^2*b^8*c^8*d^2*f^2 + 56*B^2*a^6*b^4*c^2*d^8*f^2 + 56
*B^2*a^2*b^8*c^6*d^4*f^2 + 52*B^2*a^7*b^3*c^3*d^7*f^2 + 52*B^2*a^3*b^7*c^7*
d^3*f^2 + 44*B^2*a^3*b^7*c^3*d^7*f^2 + 33*B^2*a^8*b^2*c^6*d^4*f^2 + 33*B^2*
a^6*b^4*c^8*d^2*f^2 + 20*B^2*a^8*b^2*c^4*d^6*f^2 - 20*B^2*a^7*b^3*c^7*d^3*f
^2 + 20*B^2*a^4*b^6*c^8*d^2*f^2 + 8*B^2*a^2*b^8*c^2*d^8*f^2 + 337*A^2*a^4*b
^6*c^2*d^8*f^2 + 337*A^2*a^2*b^8*c^4*d^6*f^2 + 272*A^2*a^2*b^8*c^2*d^8*f^2
+ 252*A^2*a^5*b^5*c^5*d^5*f^2 + 244*A^2*a^4*b^6*c^4*d^6*f^2 - 236*A^2*a^3*b
^7*c^3*d^7*f^2 + 176*A^2*a^6*b^4*c^2*d^8*f^2 + 176*A^2*a^2*b^8*c^6*d^4*f^2
- 148*A^2*a^7*b^3*c^3*d^7*f^2 - 148*A^2*a^3*b^7*c^7*d^3*f^2 - 140*A^2*a^6*b
^4*c^6*d^4*f^2 + 109*A^2*a^8*b^2*c^2*d^8*f^2 + 109*A^2*a^2*b^8*c^8*d^2*f^2
- 108*A^2*a^5*b^5*c^3*d^7*f^2 - 108*A^2*a^3*b^7*c^5*d^5*f^2 + 84*A^2*a^7*b^
3*c^5*d^5*f^2 + 84*A^2*a^5*b^5*c^7*d^3*f^2 + 32*A^2*a^8*b^2*c^4*d^6*f^2 + 3
2*A^2*a^4*b^6*c^8*d^2*f^2 + 20*A^2*a^7*b^3*c^7*d^3*f^2 - 15*A^2*a^8*b^2*c^6
*d^4*f^2 - 15*A^2*a^6*b^4*c^8*d^2*f^2 - 12*A^2*a^6*b^4*c^4*d^6*f^2 - 12*A^2
*a^4*b^6*c^6*d^4*f^2 + 8*B*C*b^10*c^9*d*f^2 - 16*B*C*a^10*c*d^9*f^2 - 16*A*
B*b^10*c^9*d*f^2 - 16*A*B*b^10*c*d^9*f^2 + 8*B*C*a^9*b*d^10*f^2 - 16*B*C*a*
b^9*c^10*f^2 + 16*A*B*a^10*c*d^9*f^2 - 16*A*B*a^9*b*d^10*f^2 - 16*A*B*a*b^9
*d^10*f^2 + 16*A*B*a*b^9*c^10*f^2 + 22*C^2*a^9*b*c^5*d^5*f^2 + 22*C^2*a^5*b
^5*c^9*d*f^2 + 22*C^2*a^5*b^5*c*d^9*f^2 + 22*C^2*a*b^9*c^5*d^5*f^2 - 20*C^2
*a^9*b*c^3*d^7*f^2 - 20*C^2*a^7*b^3*c*d^9*f^2 - 20*C^2*a^3*b^7*c^9*d*f^2 -
20*C^2*a*b^9*c^7*d^3*f^2 + 36*B^2*a^7*b^3*c*d^9*f^2 + 36*B^2*a*b^9*c^7*d^3*
f^2 + 28*B^2*a^9*b*c^3*d^7*f^2 + 28*B^2*a^3*b^7*c^9*d*f^2 + 24*B^2*a^3*b^7*
c*d^9*f^2 + 24*B^2*a*b^9*c^3*d^7*f^2 - 18*B^2*a^9*b*c^5*d^5*f^2 - 18*B^2*a^
5*b^5*c^9*d*f^2 + 6*B^2*a^5*b^5*c*d^9*f^2 + 6*B^2*a*b^9*c^5*d^5*f^2 - 96*A^
2*a^3*b^7*c*d^9*f^2 - 96*A^2*a*b^9*c^3*d^7*f^2 - 90*A^2*a^5*b^5*c*d^9*f^2 -
 90*A^2*a*b^9*c^5*d^5*f^2 - 84*A^2*a^7*b^3*c*d^9*f^2 - 84*A^2*a*b^9*c^7*d^3
*f^2 - 52*A^2*a^9*b*c^3*d^7*f^2 - 52*A^2*a^3*b^7*c^9*d*f^2 + 6*A^2*a^9*b*c^
5*d^5*f^2 + 6*A^2*a^5*b^5*c^9*d*f^2 - 10*C^2*a^9*b*c*d^9*f^2 - 10*C^2*a*b^9
*c^9*d*f^2 + 14*B^2*a^9*b*c*d^9*f^2 + 14*B^2*a*b^9*c^9*d*f^2 + 8*B^2*a*b^9*
c*d^9*f^2 - 32*A^2*a*b^9*c*d^9*f^2 - 26*A^2*a^9*b*c*d^9*f^2 - 26*A^2*a*b^9*
c^9*d*f^2 + 2*A*C*b^10*c^10*f^2 + 2*A*C*a^10*d^10*f^2 + 14*C^2*b^10*c^8*d^2
*f^2 - C^2*b^10*c^6*d^4*f^2 + 31*B^2*b^10*c^6*d^4*f^2 + 20*B^2*b^10*c^4*d^6

*f^2 + 14*C^2*a^10*c^2*d^8*f^2 + 4*B^2*b^10*c^2*d^8*f^2 + 2*B^2*b^10*c^8*d^
2*f^2 - C^2*a^10*c^4*d^6*f^2 + 80*A^2*b^10*c^4*d^6*f^2 + 64*A^2*b^10*c^2*d^
8*f^2 + 31*A^2*b^10*c^6*d^4*f^2 + 14*C^2*a^8*b^2*d^10*f^2 + 14*A^2*b^10*c^8
*d^2*f^2 - 10*B^2*a^10*c^2*d^8*f^2 + 3*B^2*a^10*c^4*d^6*f^2 - C^2*a^6*b^4*d
^10*f^2 + 31*B^2*a^6*b^4*d^10*f^2 + 20*B^2*a^4*b^6*d^10*f^2 + 14*C^2*a^2*b^
8*c^10*f^2 + 14*A^2*a^10*c^2*d^8*f^2 + 4*B^2*a^2*b^8*d^10*f^2 + 2*B^2*a^8*b
^2*d^10*f^2 - C^2*a^4*b^6*c^10*f^2 - A^2*a^10*c^4*d^6*f^2 + 80*A^2*a^4*b^6*
d^10*f^2 + 64*A^2*a^2*b^8*d^10*f^2 + 31*A^2*a^6*b^4*d^10*f^2 + 14*A^2*a^8*b
^2*d^10*f^2 - 10*B^2*a^2*b^8*c^10*f^2 + 3*B^2*a^4*b^6*c^10*f^2 + 14*A^2*a^2
*b^8*c^10*f^2 - A^2*a^4*b^6*c^10*f^2 - C^2*b^10*c^10*f^2 - C^2*a^10*d^10*f^
2 + 16*A^2*b^10*d^10*f^2 + 3*B^2*b^10*c^10*f^2 + 3*B^2*a^10*d^10*f^2 - A^2*
b^10*c^10*f^2 - A^2*a^10*d^10*f^2 - 96*A*B*C*a*b^7*c*d^7*f - 28*A*B*C*a^7*b
*c*d^7*f - 28*A*B*C*a*b^7*c^7*d*f + 484*A*B*C*a^4*b^4*c^4*d^4*f - 424*A*B*C
*a^3*b^5*c^3*d^5*f + 320*A*B*C*a^2*b^6*c^2*d^6*f - 176*A*B*C*a^6*b^2*c^2*d^
6*f - 176*A*B*C*a^2*b^6*c^6*d^2*f + 158*A*B*C*a^4*b^4*c^2*d^6*f + 158*A*B*C
*a^2*b^6*c^4*d^4*f - 136*A*B*C*a^5*b^3*c^5*d^3*f - 34*A*B*C*a^6*b^2*c^4*d^4
*f - 34*A*B*C*a^4*b^4*c^6*d^2*f + 28*A*B*C*a^5*b^3*c^3*d^5*f + 28*A*B*C*a^3
*b^5*c^5*d^3*f + 308*A*B*C*a^5*b^3*c*d^7*f + 308*A*B*C*a*b^7*c^5*d^3*f + 20
*A*B*C*a^7*b*c^3*d^5*f + 20*A*B*C*a^3*b^5*c^7*d*f + 30*B*C^2*a^7*b*c*d^7*f
+ 30*B*C^2*a*b^7*c^7*d*f + 160*A^2*B*a*b^7*c*d^7*f - 2*A^2*B*a^7*b*c*d^7*f
- 2*A^2*B*a*b^7*c^7*d*f - 96*A*B*C*b^8*c^4*d^4*f + 34*A*B*C*b^8*c^6*d^2*f -
 32*A*B*C*b^8*c^2*d^6*f + 2*A*B*C*a^8*c^2*d^6*f - 96*A*B*C*a^4*b^4*d^8*f +
34*A*B*C*a^6*b^2*d^8*f - 32*A*B*C*a^2*b^6*d^8*f + 2*A*B*C*a^2*b^6*c^8*f - 2
10*B*C^2*a^4*b^4*c^4*d^4*f - 182*B^2*C*a^5*b^3*c^2*d^6*f - 182*B^2*C*a^2*b^
6*c^5*d^3*f + 180*B*C^2*a^5*b^3*c^5*d^3*f + 180*B*C^2*a^3*b^5*c^3*d^5*f - 1
66*B^2*C*a^5*b^3*c^4*d^4*f - 166*B^2*C*a^4*b^4*c^5*d^3*f + 152*B*C^2*a^6*b^
2*c^2*d^6*f + 152*B*C^2*a^2*b^6*c^6*d^2*f - 112*B^2*C*a^3*b^5*c^2*d^6*f - 1
12*B^2*C*a^2*b^6*c^3*d^5*f + 94*B^2*C*a^4*b^4*c^3*d^5*f + 94*B^2*C*a^3*b^5*
c^4*d^4*f - 80*B*C^2*a^2*b^6*c^2*d^6*f + 66*B*C^2*a^5*b^3*c^3*d^5*f + 66*B*
C^2*a^3*b^5*c^5*d^3*f + 46*B^2*C*a^6*b^2*c^3*d^5*f + 46*B^2*C*a^3*b^5*c^6*d
^2*f + 33*B*C^2*a^6*b^2*c^4*d^4*f + 33*B*C^2*a^4*b^4*c^6*d^2*f + 24*B^2*C*a
^6*b^2*c^5*d^3*f + 24*B^2*C*a^5*b^3*c^6*d^2*f - 16*B*C^2*a^6*b^2*c^6*d^2*f
- 15*B*C^2*a^4*b^4*c^2*d^6*f - 15*B*C^2*a^2*b^6*c^4*d^4*f - 190*A^2*C*a^4*b
^4*c^3*d^5*f - 190*A^2*C*a^3*b^5*c^4*d^4*f + 182*A^2*C*a^5*b^3*c^2*d^6*f +
182*A^2*C*a^2*b^6*c^5*d^3*f + 160*A^2*C*a^3*b^5*c^2*d^6*f + 160*A^2*C*a^2*b
^6*c^3*d^5*f - 150*A*C^2*a^5*b^3*c^2*d^6*f - 150*A*C^2*a^2*b^6*c^5*d^3*f -
126*A*C^2*a^5*b^3*c^4*d^4*f - 126*A*C^2*a^4*b^4*c^5*d^3*f + 126*A*C^2*a^4*b
^4*c^3*d^5*f + 126*A*C^2*a^3*b^5*c^4*d^4*f - 96*A*C^2*a^3*b^5*c^2*d^6*f - 9
6*A*C^2*a^2*b^6*c^3*d^5*f + 94*A^2*C*a^5*b^3*c^4*d^4*f + 94*A^2*C*a^4*b^4*c
^5*d^3*f + 54*A*C^2*a^6*b^2*c^3*d^5*f + 54*A*C^2*a^3*b^5*c^6*d^2*f + 32*A*C
^2*a^6*b^2*c^5*d^3*f + 32*A*C^2*a^5*b^3*c^6*d^2*f - 22*A^2*C*a^6*b^2*c^3*d^
5*f - 22*A^2*C*a^3*b^5*c^6*d^2*f + 500*A^2*B*a^3*b^5*c^3*d^5*f - 290*A^2*B*
a^4*b^4*c^4*d^4*f - 256*A^2*B*a^2*b^6*c^2*d^6*f - 230*A*B^2*a^4*b^4*c^3*d^5
*f - 230*A*B^2*a^3*b^5*c^4*d^4*f + 142*A*B^2*a^5*b^3*c^2*d^6*f + 142*A*B^2*
a^2*b^6*c^5*d^3*f - 127*A^2*B*a^4*b^4*c^2*d^6*f - 127*A^2*B*a^2*b^6*c^4*d^4

*f + 86*A*B^2*a^5*b^3*c^4*d^4*f + 86*A*B^2*a^4*b^4*c^5*d^3*f + 80*A*B^2*a^3
*b^5*c^2*d^6*f + 80*A*B^2*a^2*b^6*c^3*d^5*f + 40*A^2*B*a^6*b^2*c^2*d^6*f +
40*A^2*B*a^2*b^6*c^6*d^2*f + 34*A^2*B*a^5*b^3*c^3*d^5*f + 34*A^2*B*a^3*b^5*
c^5*d^3*f - 30*A*B^2*a^6*b^2*c^3*d^5*f - 30*A*B^2*a^3*b^5*c^6*d^2*f + 20*A^
2*B*a^5*b^3*c^5*d^3*f - 15*A^2*B*a^6*b^2*c^4*d^4*f - 15*A^2*B*a^4*b^4*c^6*d
^2*f - 98*B^2*C*a^6*b^2*c*d^7*f - 98*B^2*C*a*b^7*c^6*d^2*f - 90*B*C^2*a^5*b
^3*c*d^7*f - 90*B*C^2*a*b^7*c^5*d^3*f + 48*B^2*C*a^4*b^4*c*d^7*f + 48*B^2*C
*a*b^7*c^4*d^4*f + 40*B^2*C*a^2*b^6*c*d^7*f + 40*B^2*C*a*b^7*c^2*d^6*f - 32
*B*C^2*a^3*b^5*c*d^7*f - 32*B*C^2*a*b^7*c^3*d^5*f + 26*B^2*C*a^7*b*c^2*d^6*
f + 26*B^2*C*a^2*b^6*c^7*d*f - 26*B*C^2*a^7*b*c^3*d^5*f - 26*B*C^2*a^3*b^5*
c^7*d*f - 8*B^2*C*a^7*b*c^4*d^4*f - 8*B^2*C*a^4*b^4*c^7*d*f - 224*A^2*C*a^4
*b^4*c*d^7*f - 224*A^2*C*a*b^7*c^4*d^4*f - 96*A^2*C*a^2*b^6*c*d^7*f - 96*A^
2*C*a*b^7*c^2*d^6*f + 96*A*C^2*a^4*b^4*c*d^7*f + 96*A*C^2*a*b^7*c^4*d^4*f -
66*A*C^2*a^6*b^2*c*d^7*f - 66*A*C^2*a*b^7*c^6*d^2*f + 64*A*C^2*a^2*b^6*c*d
^7*f + 64*A*C^2*a*b^7*c^2*d^6*f + 34*A^2*C*a^6*b^2*c*d^7*f + 34*A^2*C*a*b^7
*c^6*d^2*f + 34*A*C^2*a^7*b*c^2*d^6*f + 34*A*C^2*a^2*b^6*c^7*d*f - 2*A^2*C*
a^7*b*c^2*d^6*f - 2*A^2*C*a^2*b^6*c^7*d*f - 208*A*B^2*a^4*b^4*c*d^7*f - 208
*A*B^2*a*b^7*c^4*d^4*f + 160*A^2*B*a^3*b^5*c*d^7*f + 160*A^2*B*a*b^7*c^3*d^
5*f - 154*A^2*B*a^5*b^3*c*d^7*f - 154*A^2*B*a*b^7*c^5*d^3*f - 112*A*B^2*a^2
*b^6*c*d^7*f - 112*A*B^2*a*b^7*c^2*d^6*f + 58*A*B^2*a^6*b^2*c*d^7*f + 58*A*
B^2*a*b^7*c^6*d^2*f - 10*A*B^2*a^7*b*c^2*d^6*f - 10*A*B^2*a^2*b^6*c^7*d*f +
6*A^2*B*a^7*b*c^3*d^5*f + 6*A^2*B*a^3*b^5*c^7*d*f + 32*B^2*C*b^8*c^5*d^3*f
- 17*B*C^2*b^8*c^6*d^2*f + 8*B^2*C*b^8*c^3*d^5*f + 64*A^2*C*b^8*c^3*d^5*f
- 32*A^2*C*b^8*c^5*d^3*f + 32*A*C^2*b^8*c^5*d^3*f - B*C^2*a^8*c^2*d^6*f + 1
12*A^2*B*b^8*c^4*d^4*f - 64*A*B^2*b^8*c^5*d^3*f + 32*B^2*C*a^5*b^3*d^8*f -
17*B*C^2*a^6*b^2*d^8*f + 16*A^2*B*b^8*c^2*d^6*f + 16*A*B^2*b^8*c^3*d^5*f +
8*B^2*C*a^3*b^5*d^8*f - A^2*B*b^8*c^6*d^2*f + 64*A^2*C*a^3*b^5*d^8*f - 32*A
^2*C*a^5*b^3*d^8*f + 32*A*C^2*a^5*b^3*d^8*f - A^2*B*a^8*c^2*d^6*f - B*C^2*a
^2*b^6*c^8*f + 112*A^2*B*a^4*b^4*d^8*f - 64*A*B^2*a^5*b^3*d^8*f + 16*A^2*B*
a^2*b^6*d^8*f + 16*A*B^2*a^3*b^5*d^8*f - A^2*B*a^6*b^2*d^8*f - A^2*B*a^2*b^
6*c^8*f - 8*B^3*a*b^7*c*d^7*f - 2*B^3*a^7*b*c*d^7*f - 2*B^3*a*b^7*c^7*d*f -
6*B^2*C*b^8*c^7*d*f + 32*A^2*C*b^8*c*d^7*f + 6*A^2*C*b^8*c^7*d*f - 6*A*C^2
*b^8*c^7*d*f - 2*B^2*C*a^8*c*d^7*f + 16*A*B^2*b^8*c*d^7*f - 6*B^2*C*a^7*b*d
^8*f - 6*A^2*C*a^8*c*d^7*f + 6*A*C^2*a^8*c*d^7*f - 2*A*B^2*b^8*c^7*d*f + 32
*A^2*C*a*b^7*d^8*f + 6*A^2*C*a^7*b*d^8*f - 6*A*C^2*a^7*b*d^8*f - 2*B^2*C*a*
b^7*c^8*f + 2*A*B^2*a^8*c*d^7*f + 16*A*B^2*a*b^7*d^8*f - 6*A^2*C*a*b^7*c^8*
f + 6*A*C^2*a*b^7*c^8*f - 2*A*B^2*a^7*b*d^8*f + 2*A*B^2*a*b^7*c^8*f - 50*C^
3*a^6*b^2*c^3*d^5*f + 50*C^3*a^5*b^3*c^2*d^6*f - 50*C^3*a^3*b^5*c^6*d^2*f +
50*C^3*a^2*b^6*c^5*d^3*f + 42*C^3*a^5*b^3*c^4*d^4*f + 42*C^3*a^4*b^4*c^5*d
^3*f - 42*C^3*a^4*b^4*c^3*d^5*f - 42*C^3*a^3*b^5*c^4*d^4*f - 32*C^3*a^6*b^2
*c^5*d^3*f - 32*C^3*a^5*b^3*c^6*d^2*f + 32*C^3*a^3*b^5*c^2*d^6*f + 32*C^3*a
^2*b^6*c^3*d^5*f + 94*B^3*a^4*b^4*c^4*d^4*f + 48*B^3*a^2*b^6*c^2*d^6*f - 44
*B^3*a^3*b^5*c^3*d^5*f - 32*B^3*a^6*b^2*c^2*d^6*f - 32*B^3*a^2*b^6*c^6*d^2*
f + 29*B^3*a^4*b^4*c^2*d^6*f + 29*B^3*a^2*b^6*c^4*d^4*f - 20*B^3*a^5*b^3*c^
5*d^3*f + 18*B^3*a^5*b^3*c^3*d^5*f + 18*B^3*a^3*b^5*c^5*d^3*f - 3*B^3*a^6*b

^2*c^4*d^4*f - 3*B^3*a^4*b^4*c^6*d^2*f + 106*A^3*a^4*b^4*c^3*d^5*f + 106*A^
3*a^3*b^5*c^4*d^4*f - 96*A^3*a^3*b^5*c^2*d^6*f - 96*A^3*a^2*b^6*c^3*d^5*f -
 82*A^3*a^5*b^3*c^2*d^6*f - 82*A^3*a^2*b^6*c^5*d^3*f + 18*A^3*a^6*b^2*c^3*d
^5*f + 18*A^3*a^3*b^5*c^6*d^2*f - 10*A^3*a^5*b^3*c^4*d^4*f - 10*A^3*a^4*b^4
*c^5*d^3*f - 22*C^3*a^7*b*c^2*d^6*f + 22*C^3*a^6*b^2*c*d^7*f - 22*C^3*a^2*b
^6*c^7*d*f + 22*C^3*a*b^7*c^6*d^2*f - 2*A*B*C*b^8*c^8*f - 2*A*B*C*a^8*d^8*f
 + 62*B^3*a^5*b^3*c*d^7*f + 62*B^3*a*b^7*c^5*d^3*f + 16*B^3*a^3*b^5*c*d^7*f
 + 16*B^3*a*b^7*c^3*d^5*f + 6*B^3*a^7*b*c^3*d^5*f + 6*B^3*a^3*b^5*c^7*d*f +
 128*A^3*a^4*b^4*c*d^7*f + 128*A^3*a*b^7*c^4*d^4*f + 32*A^3*a^2*b^6*c*d^7*f
 + 32*A^3*a*b^7*c^2*d^6*f - 10*A^3*a^7*b*c^2*d^6*f + 10*A^3*a^6*b^2*c*d^7*f
 - 10*A^3*a^2*b^6*c^7*d*f + 10*A^3*a*b^7*c^6*d^2*f + 11*B^3*b^8*c^6*d^2*f -
 8*B^3*b^8*c^4*d^4*f - 4*B^3*b^8*c^2*d^6*f - 64*A^3*b^8*c^3*d^5*f - B^3*a^8
*c^2*d^6*f + 11*B^3*a^6*b^2*d^8*f - 8*B^3*a^4*b^4*d^8*f - 4*B^3*a^2*b^6*d^8
*f - 64*A^3*a^3*b^5*d^8*f - B^3*a^2*b^6*c^8*f + 2*C^3*b^8*c^7*d*f - 2*C^3*a
^8*c*d^7*f - 32*A^3*b^8*c*d^7*f + 2*C^3*a^7*b*d^8*f - 2*A^3*b^8*c^7*d*f - 2
*C^3*a*b^7*c^8*f + 2*A^3*a^8*c*d^7*f - 32*A^3*a*b^7*d^8*f - 2*A^3*a^7*b*d^8
*f + 2*A^3*a*b^7*c^8*f - 16*A^2*B*b^8*d^8*f + B*C^2*b^8*c^8*f + B*C^2*a^8*d
^8*f + A^2*B*b^8*c^8*f + A^2*B*a^8*d^8*f + B^3*b^8*c^8*f + B^3*a^8*d^8*f -
4*A*B^2*C*a^5*b*c*d^5 - 4*A*B^2*C*a*b^5*c^5*d + 4*A*B^2*C*a*b^5*c*d^5 + 22*
A^2*B*C*a^3*b^3*c^2*d^4 + 22*A^2*B*C*a^2*b^4*c^3*d^3 - 20*A*B^2*C*a^3*b^3*c
^3*d^3 + 14*A*B^2*C*a^4*b^2*c^2*d^4 + 14*A*B^2*C*a^2*b^4*c^4*d^2 - 14*A*B*C
^2*a^3*b^3*c^2*d^4 - 14*A*B*C^2*a^2*b^4*c^3*d^3 + 12*A*B*C^2*a^4*b^2*c^3*d^
3 + 12*A*B*C^2*a^3*b^3*c^4*d^2 - 6*A^2*B*C*a^4*b^2*c^3*d^3 - 6*A^2*B*C*a^3*
b^3*c^4*d^2 - 4*A*B^2*C*a^2*b^4*c^2*d^4 + 22*A*B*C^2*a^4*b^2*c*d^5 + 22*A*B
*C^2*a*b^5*c^4*d^2 - 20*A^2*B*C*a^4*b^2*c*d^5 - 20*A^2*B*C*a*b^5*c^4*d^2 +
10*A*B*C^2*a^2*b^4*c*d^5 + 10*A*B*C^2*a*b^5*c^2*d^4 - 8*A^2*B*C*a^2*b^4*c*d
^5 - 8*A^2*B*C*a*b^5*c^2*d^4 + 4*A*B^2*C*a^3*b^3*c*d^5 + 4*A*B^2*C*a*b^5*c^
3*d^3 - 4*A*B*C^2*a^5*b*c^2*d^4 - 4*A*B*C^2*a^2*b^4*c^5*d + 2*A^2*B*C*a^5*b
*c^2*d^4 + 2*A^2*B*C*a^2*b^4*c^5*d - 8*B^3*C*a^4*b^2*c*d^5 - 8*B^3*C*a*b^5*
c^4*d^2 - 8*B*C^3*a^4*b^2*c*d^5 - 8*B*C^3*a*b^5*c^4*d^2 - 4*B^3*C*a^2*b^4*c
*d^5 - 4*B^3*C*a*b^5*c^2*d^4 + 4*B^2*C^2*a^5*b*c*d^5 + 4*B^2*C^2*a*b^5*c^5*
d - 4*B*C^3*a^2*b^4*c*d^5 - 4*B*C^3*a*b^5*c^2*d^4 + 2*B^3*C*a^5*b*c^2*d^4 +
 2*B^3*C*a^2*b^4*c^5*d + 2*B^2*C^2*a*b^5*c*d^5 + 2*B*C^3*a^5*b*c^2*d^4 + 2*
B*C^3*a^2*b^4*c^5*d + 24*A^3*C*a^3*b^3*c*d^5 + 24*A^3*C*a*b^5*c^3*d^3 - 24*
A^2*C^2*a*b^5*c*d^5 + 12*A^2*C^2*a^5*b*c*d^5 + 12*A^2*C^2*a*b^5*c^5*d + 8*A
*C^3*a^3*b^3*c*d^5 + 8*A*C^3*a*b^5*c^3*d^3 + 6*A^3*B*a^4*b^2*c*d^5 + 6*A^3*
B*a*b^5*c^4*d^2 - 6*A^2*B^2*a*b^5*c*d^5 + 6*A*B^3*a^4*b^2*c*d^5 + 6*A*B^3*a
*b^5*c^4*d^2 + 2*A^3*B*a^2*b^4*c*d^5 + 2*A^3*B*a*b^5*c^2*d^4 + 2*A*B^3*a^2*
b^4*c*d^5 + 2*A*B^3*a*b^5*c^2*d^4 + 20*A^2*B*C*b^6*c^3*d^3 - 10*A*B*C^2*b^6
*c^3*d^3 - 2*A*B^2*C*b^6*c^4*d^2 - 2*A*B^2*C*b^6*c^2*d^4 + 20*A^2*B*C*a^3*b
^3*d^6 - 10*A*B*C^2*a^3*b^3*d^6 - 2*A*B^2*C*a^4*b^2*d^6 - 2*A*B^2*C*a^2*b^4
*d^6 + 10*B^2*C^2*a^3*b^3*c^3*d^3 + 4*B^2*C^2*a^4*b^2*c^4*d^2 - 3*B^2*C^2*a
^4*b^2*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 2*B^2*C^2*a^2*b^4*c^2*d^4 + 40
*A^2*C^2*a^2*b^4*c^2*d^4 - 16*A^2*C^2*a^4*b^2*c^2*d^4 - 16*A^2*C^2*a^2*b^4*
c^4*d^2 + 4*A^2*C^2*a^4*b^2*c^4*d^2 + 18*A^2*B^2*a^2*b^4*c^2*d^4 + 10*A^2*B

^2*a^3*b^3*c^3*d^3 - 3*A^2*B^2*a^4*b^2*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2
+ 24*A^3*C*a*b^5*c*d^5 - 12*A*C^3*a^5*b*c*d^5 - 12*A*C^3*a*b^5*c^5*d + 8*A*
C^3*a*b^5*c*d^5 - 4*A^3*C*a^5*b*c*d^5 - 4*A^3*C*a*b^5*c^5*d + 8*A^2*B*C*b^6
*c*d^5 + 4*A*B*C^2*b^6*c^5*d - 4*A*B*C^2*b^6*c*d^5 - 2*A^2*B*C*b^6*c^5*d +
8*A^2*B*C*a*b^5*d^6 + 4*A*B*C^2*a^5*b*d^6 - 4*A*B*C^2*a*b^5*d^6 - 2*A^2*B*C
*a^5*b*d^6 - 6*B^3*C*a^4*b^2*c^3*d^3 - 6*B^3*C*a^3*b^3*c^4*d^2 - 6*B*C^3*a^
4*b^2*c^3*d^3 - 6*B*C^3*a^3*b^3*c^4*d^2 + 2*B^3*C*a^3*b^3*c^2*d^4 + 2*B^3*C
*a^2*b^4*c^3*d^3 + 2*B^2*C^2*a^3*b^3*c*d^5 + 2*B^2*C^2*a*b^5*c^3*d^3 + 2*B*
C^3*a^3*b^3*c^2*d^4 + 2*B*C^3*a^2*b^4*c^3*d^3 - 48*A^3*C*a^2*b^4*c^2*d^4 -
24*A^2*C^2*a^3*b^3*c*d^5 - 24*A^2*C^2*a*b^5*c^3*d^3 - 16*A*C^3*a^2*b^4*c^2*
d^4 + 8*A^3*C*a^4*b^2*c^2*d^4 + 8*A^3*C*a^2*b^4*c^4*d^2 - 8*A*C^3*a^4*b^2*c
^4*d^2 + 8*A*C^3*a^4*b^2*c^2*d^4 + 8*A*C^3*a^2*b^4*c^4*d^2 - 10*A^3*B*a^3*b
^3*c^2*d^4 - 10*A^3*B*a^2*b^4*c^3*d^3 - 10*A*B^3*a^3*b^3*c^2*d^4 - 10*A*B^3
*a^2*b^4*c^3*d^3 - 6*A^2*B^2*a^3*b^3*c*d^5 - 6*A^2*B^2*a*b^5*c^3*d^3 + 3*B^
2*C^2*b^6*c^4*d^2 - 8*A^2*C^2*b^6*c^4*d^2 + 8*A^2*C^2*b^6*c^2*d^4 + 9*A^2*B
^2*b^6*c^2*d^4 + 3*B^2*C^2*a^4*b^2*d^6 + 3*A^2*B^2*b^6*c^4*d^2 - 8*A^2*C^2*
a^4*b^2*d^6 + 8*A^2*C^2*a^2*b^4*d^6 + 9*A^2*B^2*a^2*b^4*d^6 + 3*A^2*B^2*a^4
*b^2*d^6 + 2*B^4*a^3*b^3*c*d^5 + 2*B^4*a*b^5*c^3*d^3 - 8*A^4*a^3*b^3*c*d^5
- 8*A^4*a*b^5*c^3*d^3 - 16*A^3*C*b^6*c^2*d^4 + 4*A^3*C*b^6*c^4*d^2 + 4*A*C^
3*b^6*c^4*d^2 - 10*A^3*B*b^6*c^3*d^3 - 10*A*B^3*b^6*c^3*d^3 - 16*A^3*C*a^2*
b^4*d^6 + 4*A^3*C*a^4*b^2*d^6 + 4*A*C^3*a^4*b^2*d^6 - 10*A^3*B*a^3*b^3*d^6
- 10*A*B^3*a^3*b^3*d^6 + 4*C^4*a^5*b*c*d^5 + 4*C^4*a*b^5*c^5*d + 2*B^4*a*b^
5*c*d^5 - 8*A^4*a*b^5*c*d^5 - 2*B^3*C*b^6*c^5*d - 2*B*C^3*b^6*c^5*d - 4*A^3
*B*b^6*c*d^5 - 4*A*B^3*b^6*c*d^5 - 2*B^3*C*a^5*b*d^6 - 2*B*C^3*a^5*b*d^6 -
4*A^3*B*a*b^5*d^6 - 4*A*B^3*a*b^5*d^6 + 4*C^4*a^4*b^2*c^4*d^2 + 4*C^4*a^2*b
^4*c^2*d^4 + 10*B^4*a^3*b^3*c^3*d^3 - 3*B^4*a^4*b^2*c^2*d^4 - 3*B^4*a^2*b^4
*c^4*d^2 - 2*B^4*a^2*b^4*c^2*d^4 + 20*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*b^6*c^2
*d^4 + B^2*C^2*a^2*b^4*d^6 - 8*A^3*C*b^6*d^6 + 3*B^4*b^6*c^4*d^2 + 8*A^4*b^
6*c^2*d^4 + 3*B^4*a^4*b^2*d^6 + 8*A^4*a^2*b^4*d^6 + 4*A^2*C^2*b^6*d^6 + 4*A
^2*B^2*b^6*d^6 + 4*A^4*b^6*d^6 + B^4*b^6*c^2*d^4 + B^4*a^2*b^4*d^6, f, k)*(
root(144*a^13*b*c^5*d^9*f^4 + 144*a^9*b^5*c*d^13*f^4 + 144*a^5*b^9*c^13*d*f
^4 + 144*a*b^13*c^9*d^5*f^4 + 96*a^13*b*c^7*d^7*f^4 + 96*a^13*b*c^3*d^11*f^
4 + 96*a^11*b^3*c*d^13*f^4 + 96*a^7*b^7*c^13*d*f^4 + 96*a^7*b^7*c*d^13*f^4
+ 96*a^3*b^11*c^13*d*f^4 + 96*a*b^13*c^11*d^3*f^4 + 96*a*b^13*c^7*d^7*f^4 +
24*a^13*b*c^9*d^5*f^4 + 24*a^9*b^5*c^13*d*f^4 + 24*a^5*b^9*c*d^13*f^4 + 24
*a*b^13*c^5*d^9*f^4 + 24*a^13*b*c*d^13*f^4 + 24*a*b^13*c^13*d*f^4 + 3648*a^
7*b^7*c^7*d^7*f^4 - 3188*a^8*b^6*c^6*d^8*f^4 - 3188*a^6*b^8*c^8*d^6*f^4 - 2
912*a^8*b^6*c^8*d^6*f^4 - 2912*a^6*b^8*c^6*d^8*f^4 + 2592*a^9*b^5*c^7*d^7*f
^4 + 2592*a^7*b^7*c^9*d^5*f^4 + 2592*a^7*b^7*c^5*d^9*f^4 + 2592*a^5*b^9*c^7
*d^7*f^4 + 2168*a^9*b^5*c^5*d^9*f^4 + 2168*a^5*b^9*c^9*d^5*f^4 - 1776*a^10*
b^4*c^6*d^8*f^4 - 1776*a^8*b^6*c^4*d^10*f^4 - 1776*a^6*b^8*c^10*d^4*f^4 - 1
776*a^4*b^10*c^8*d^6*f^4 + 1568*a^9*b^5*c^9*d^5*f^4 + 1568*a^5*b^9*c^5*d^9*
f^4 - 1344*a^10*b^4*c^8*d^6*f^4 - 1344*a^8*b^6*c^10*d^4*f^4 - 1344*a^6*b^8*
c^4*d^10*f^4 - 1344*a^4*b^10*c^6*d^8*f^4 - 1164*a^10*b^4*c^4*d^10*f^4 - 116
4*a^4*b^10*c^10*d^4*f^4 + 896*a^11*b^3*c^5*d^9*f^4 + 896*a^9*b^5*c^3*d^11*f

^4 + 896*a^5*b^9*c^11*d^3*f^4 + 896*a^3*b^11*c^9*d^5*f^4 + 864*a^11*b^3*c^7
*d^7*f^4 + 864*a^7*b^7*c^11*d^3*f^4 + 864*a^7*b^7*c^3*d^11*f^4 + 864*a^3*b^
11*c^7*d^7*f^4 - 480*a^10*b^4*c^10*d^4*f^4 - 480*a^4*b^10*c^4*d^10*f^4 + 46
4*a^11*b^3*c^3*d^11*f^4 + 464*a^3*b^11*c^11*d^3*f^4 - 424*a^12*b^2*c^6*d^8*
f^4 - 424*a^8*b^6*c^2*d^12*f^4 - 424*a^6*b^8*c^12*d^2*f^4 - 424*a^2*b^12*c^
8*d^6*f^4 + 416*a^11*b^3*c^9*d^5*f^4 + 416*a^9*b^5*c^11*d^3*f^4 + 416*a^5*b
^9*c^3*d^11*f^4 + 416*a^3*b^11*c^5*d^9*f^4 - 336*a^12*b^2*c^4*d^10*f^4 - 33
6*a^10*b^4*c^2*d^12*f^4 - 336*a^4*b^10*c^12*d^2*f^4 - 336*a^2*b^12*c^10*d^4
*f^4 - 256*a^12*b^2*c^8*d^6*f^4 - 256*a^8*b^6*c^12*d^2*f^4 - 256*a^6*b^8*c^
2*d^12*f^4 - 256*a^2*b^12*c^6*d^8*f^4 - 124*a^12*b^2*c^2*d^12*f^4 - 124*a^2
*b^12*c^12*d^2*f^4 + 80*a^11*b^3*c^11*d^3*f^4 + 80*a^3*b^11*c^3*d^11*f^4 -
60*a^12*b^2*c^10*d^4*f^4 - 60*a^10*b^4*c^12*d^2*f^4 - 60*a^4*b^10*c^2*d^12*
f^4 - 60*a^2*b^12*c^4*d^10*f^4 - 24*b^14*c^10*d^4*f^4 - 16*b^14*c^12*d^2*f^
4 - 16*b^14*c^8*d^6*f^4 - 4*b^14*c^6*d^8*f^4 - 24*a^14*c^4*d^10*f^4 - 16*a^
14*c^6*d^8*f^4 - 16*a^14*c^2*d^12*f^4 - 4*a^14*c^8*d^6*f^4 - 24*a^10*b^4*d^
14*f^4 - 16*a^12*b^2*d^14*f^4 - 16*a^8*b^6*d^14*f^4 - 4*a^6*b^8*d^14*f^4 -
24*a^4*b^10*c^14*f^4 - 16*a^6*b^8*c^14*f^4 - 16*a^2*b^12*c^14*f^4 - 4*a^8*b
^6*c^14*f^4 - 4*b^14*c^14*f^4 - 4*a^14*d^14*f^4 + 36*A*C*a^9*b*c*d^9*f^2 +
36*A*C*a*b^9*c^9*d*f^2 + 32*A*C*a*b^9*c*d^9*f^2 - 552*B*C*a^7*b^3*c^4*d^6*f
^2 - 552*B*C*a^4*b^6*c^7*d^3*f^2 - 408*B*C*a^5*b^5*c^4*d^6*f^2 - 408*B*C*a^
4*b^6*c^5*d^5*f^2 + 360*B*C*a^6*b^4*c^3*d^7*f^2 + 360*B*C*a^3*b^7*c^6*d^4*f
^2 - 248*B*C*a^7*b^3*c^2*d^8*f^2 - 248*B*C*a^2*b^8*c^7*d^3*f^2 + 184*B*C*a^
6*b^4*c^5*d^5*f^2 + 184*B*C*a^5*b^5*c^6*d^4*f^2 + 152*B*C*a^8*b^2*c^3*d^7*f
^2 - 152*B*C*a^5*b^5*c^2*d^8*f^2 + 152*B*C*a^3*b^7*c^8*d^2*f^2 - 152*B*C*a^
2*b^8*c^5*d^5*f^2 - 104*B*C*a^7*b^3*c^6*d^4*f^2 - 104*B*C*a^6*b^4*c^7*d^3*f
^2 + 64*B*C*a^8*b^2*c^5*d^5*f^2 + 64*B*C*a^5*b^5*c^8*d^2*f^2 - 56*B*C*a^4*b
^6*c^3*d^7*f^2 - 56*B*C*a^3*b^7*c^4*d^6*f^2 - 24*B*C*a^8*b^2*c^7*d^3*f^2 -
24*B*C*a^7*b^3*c^8*d^2*f^2 - 24*B*C*a^3*b^7*c^2*d^8*f^2 - 24*B*C*a^2*b^8*c^
3*d^7*f^2 - 696*A*C*a^5*b^5*c^5*d^5*f^2 + 536*A*C*a^6*b^4*c^6*d^4*f^2 + 536
*A*C*a^6*b^4*c^4*d^6*f^2 + 536*A*C*a^4*b^6*c^6*d^4*f^2 + 472*A*C*a^4*b^6*c^
4*d^6*f^2 - 232*A*C*a^7*b^3*c^5*d^5*f^2 - 232*A*C*a^5*b^5*c^7*d^3*f^2 + 216
*A*C*a^3*b^7*c^3*d^7*f^2 + 168*A*C*a^7*b^3*c^3*d^7*f^2 + 168*A*C*a^3*b^7*c^
7*d^3*f^2 - 154*A*C*a^8*b^2*c^2*d^8*f^2 - 154*A*C*a^2*b^8*c^8*d^2*f^2 + 62*
A*C*a^8*b^2*c^6*d^4*f^2 + 62*A*C*a^6*b^4*c^8*d^2*f^2 - 40*A*C*a^7*b^3*c^7*d
^3*f^2 - 40*A*C*a^5*b^5*c^3*d^7*f^2 - 40*A*C*a^3*b^7*c^5*d^5*f^2 + 32*A*C*a
^6*b^4*c^2*d^8*f^2 + 32*A*C*a^2*b^8*c^6*d^4*f^2 - 32*A*C*a^2*b^8*c^2*d^8*f^
2 + 30*A*C*a^4*b^6*c^2*d^8*f^2 + 30*A*C*a^2*b^8*c^4*d^6*f^2 + 16*A*C*a^8*b^
2*c^4*d^6*f^2 + 16*A*C*a^4*b^6*c^8*d^2*f^2 - 488*A*B*a^6*b^4*c^3*d^7*f^2 -
488*A*B*a^3*b^7*c^6*d^4*f^2 + 440*A*B*a^7*b^3*c^4*d^6*f^2 + 440*A*B*a^4*b^6
*c^7*d^3*f^2 - 360*A*B*a^6*b^4*c^5*d^5*f^2 - 360*A*B*a^5*b^5*c^6*d^4*f^2 -
192*A*B*a^8*b^2*c^3*d^7*f^2 - 192*A*B*a^3*b^7*c^8*d^2*f^2 - 168*A*B*a^3*b^7
*c^2*d^8*f^2 - 168*A*B*a^2*b^8*c^3*d^7*f^2 - 152*A*B*a^4*b^6*c^3*d^7*f^2 -
152*A*B*a^3*b^7*c^4*d^6*f^2 - 120*A*B*a^8*b^2*c^5*d^5*f^2 + 120*A*B*a^7*b^3
*c^2*d^8*f^2 - 120*A*B*a^5*b^5*c^8*d^2*f^2 + 120*A*B*a^5*b^5*c^4*d^6*f^2 -
120*A*B*a^5*b^5*c^2*d^8*f^2 + 120*A*B*a^4*b^6*c^5*d^5*f^2 + 120*A*B*a^2*b^8

*c^7*d^3*f^2 - 120*A*B*a^2*b^8*c^5*d^5*f^2 + 40*A*B*a^7*b^3*c^6*d^4*f^2 + 4
0*A*B*a^6*b^4*c^7*d^3*f^2 - 72*B*C*a^9*b*c^4*d^6*f^2 - 72*B*C*a^4*b^6*c^9*d
*f^2 - 64*B*C*a^4*b^6*c*d^9*f^2 - 64*B*C*a*b^9*c^4*d^6*f^2 - 32*B*C*a^8*b^2
*c*d^9*f^2 - 32*B*C*a*b^9*c^8*d^2*f^2 - 16*B*C*a^2*b^8*c*d^9*f^2 - 16*B*C*a
*b^9*c^2*d^8*f^2 + 8*B*C*a^9*b*c^6*d^4*f^2 - 8*B*C*a^9*b*c^2*d^8*f^2 + 8*B*
C*a^6*b^4*c^9*d*f^2 - 8*B*C*a^2*b^8*c^9*d*f^2 + 104*A*C*a^7*b^3*c*d^9*f^2 +
104*A*C*a*b^9*c^7*d^3*f^2 + 96*A*C*a^3*b^7*c*d^9*f^2 + 96*A*C*a*b^9*c^3*d^
7*f^2 + 72*A*C*a^9*b*c^3*d^7*f^2 + 72*A*C*a^3*b^7*c^9*d*f^2 + 68*A*C*a^5*b^
5*c*d^9*f^2 + 68*A*C*a*b^9*c^5*d^5*f^2 - 28*A*C*a^9*b*c^5*d^5*f^2 - 28*A*C*
a^5*b^5*c^9*d*f^2 + 80*A*B*a^9*b*c^4*d^6*f^2 + 80*A*B*a^4*b^6*c^9*d*f^2 + 2
4*A*B*a^8*b^2*c*d^9*f^2 - 24*A*B*a^6*b^4*c*d^9*f^2 + 24*A*B*a^4*b^6*c*d^9*f
^2 - 24*A*B*a^2*b^8*c*d^9*f^2 + 24*A*B*a*b^9*c^8*d^2*f^2 - 24*A*B*a*b^9*c^6
*d^4*f^2 + 24*A*B*a*b^9*c^4*d^6*f^2 - 24*A*B*a*b^9*c^2*d^8*f^2 - 32*B*C*b^1
0*c^7*d^3*f^2 - 8*B*C*b^10*c^5*d^5*f^2 + 34*A*C*b^10*c^6*d^4*f^2 + 16*B*C*a
^10*c^3*d^7*f^2 + 16*A*C*b^10*c^4*d^6*f^2 - 12*A*C*b^10*c^8*d^2*f^2 - 96*A*
B*b^10*c^5*d^5*f^2 - 72*A*B*b^10*c^3*d^7*f^2 - 32*B*C*a^7*b^3*d^10*f^2 - 28
*A*C*a^10*c^2*d^8*f^2 - 24*A*B*b^10*c^7*d^3*f^2 - 8*B*C*a^5*b^5*d^10*f^2 +
2*A*C*a^10*c^4*d^6*f^2 + 34*A*C*a^6*b^4*d^10*f^2 + 16*B*C*a^3*b^7*c^10*f^2
+ 16*A*C*a^4*b^6*d^10*f^2 - 16*A*B*a^10*c^3*d^7*f^2 - 12*A*C*a^8*b^2*d^10*f
^2 - 96*A*B*a^5*b^5*d^10*f^2 - 72*A*B*a^3*b^7*d^10*f^2 - 28*A*C*a^2*b^8*c^1
0*f^2 - 24*A*B*a^7*b^3*d^10*f^2 + 2*A*C*a^4*b^6*c^10*f^2 - 16*A*B*a^3*b^7*c
^10*f^2 + 444*C^2*a^5*b^5*c^5*d^5*f^2 + 148*C^2*a^7*b^3*c^5*d^5*f^2 + 148*C
^2*a^5*b^5*c^7*d^3*f^2 + 148*C^2*a^5*b^5*c^3*d^7*f^2 + 148*C^2*a^3*b^7*c^5*
d^5*f^2 - 140*C^2*a^6*b^4*c^6*d^4*f^2 - 140*C^2*a^6*b^4*c^4*d^6*f^2 - 140*C
^2*a^4*b^6*c^6*d^4*f^2 - 140*C^2*a^4*b^6*c^4*d^6*f^2 + 109*C^2*a^8*b^2*c^2*
d^8*f^2 + 109*C^2*a^2*b^8*c^8*d^2*f^2 + 48*C^2*a^8*b^2*c^4*d^6*f^2 + 48*C^2
*a^6*b^4*c^2*d^8*f^2 + 48*C^2*a^4*b^6*c^8*d^2*f^2 + 48*C^2*a^2*b^8*c^6*d^4*
f^2 + 20*C^2*a^7*b^3*c^7*d^3*f^2 - 20*C^2*a^7*b^3*c^3*d^7*f^2 - 20*C^2*a^3*
b^7*c^7*d^3*f^2 + 20*C^2*a^3*b^7*c^3*d^7*f^2 + 17*C^2*a^8*b^2*c^6*d^4*f^2 +
17*C^2*a^6*b^4*c^8*d^2*f^2 + 17*C^2*a^4*b^6*c^2*d^8*f^2 + 17*C^2*a^2*b^8*c
^4*d^6*f^2 + 16*C^2*a^8*b^2*c^8*d^2*f^2 + 16*C^2*a^2*b^8*c^2*d^8*f^2 - 396*
B^2*a^5*b^5*c^5*d^5*f^2 + 308*B^2*a^6*b^4*c^4*d^6*f^2 + 308*B^2*a^4*b^6*c^6
*d^4*f^2 + 300*B^2*a^4*b^6*c^4*d^6*f^2 + 284*B^2*a^6*b^4*c^6*d^4*f^2 - 132*
B^2*a^7*b^3*c^5*d^5*f^2 - 132*B^2*a^5*b^5*c^7*d^3*f^2 - 84*B^2*a^5*b^5*c^3*
d^7*f^2 - 84*B^2*a^3*b^7*c^5*d^5*f^2 + 61*B^2*a^4*b^6*c^2*d^8*f^2 + 61*B^2*
a^2*b^8*c^4*d^6*f^2 - 59*B^2*a^8*b^2*c^2*d^8*f^2 - 59*B^2*a^2*b^8*c^8*d^2*f
^2 + 56*B^2*a^6*b^4*c^2*d^8*f^2 + 56*B^2*a^2*b^8*c^6*d^4*f^2 + 52*B^2*a^7*b
^3*c^3*d^7*f^2 + 52*B^2*a^3*b^7*c^7*d^3*f^2 + 44*B^2*a^3*b^7*c^3*d^7*f^2 +
33*B^2*a^8*b^2*c^6*d^4*f^2 + 33*B^2*a^6*b^4*c^8*d^2*f^2 + 20*B^2*a^8*b^2*c^
4*d^6*f^2 - 20*B^2*a^7*b^3*c^7*d^3*f^2 + 20*B^2*a^4*b^6*c^8*d^2*f^2 + 8*B^2
*a^2*b^8*c^2*d^8*f^2 + 337*A^2*a^4*b^6*c^2*d^8*f^2 + 337*A^2*a^2*b^8*c^4*d^
6*f^2 + 272*A^2*a^2*b^8*c^2*d^8*f^2 + 252*A^2*a^5*b^5*c^5*d^5*f^2 + 244*A^2
*a^4*b^6*c^4*d^6*f^2 - 236*A^2*a^3*b^7*c^3*d^7*f^2 + 176*A^2*a^6*b^4*c^2*d^
8*f^2 + 176*A^2*a^2*b^8*c^6*d^4*f^2 - 148*A^2*a^7*b^3*c^3*d^7*f^2 - 148*A^2
*a^3*b^7*c^7*d^3*f^2 - 140*A^2*a^6*b^4*c^6*d^4*f^2 + 109*A^2*a^8*b^2*c^2*d^

8*f^2 + 109*A^2*a^2*b^8*c^8*d^2*f^2 - 108*A^2*a^5*b^5*c^3*d^7*f^2 - 108*A^2
*a^3*b^7*c^5*d^5*f^2 + 84*A^2*a^7*b^3*c^5*d^5*f^2 + 84*A^2*a^5*b^5*c^7*d^3*
f^2 + 32*A^2*a^8*b^2*c^4*d^6*f^2 + 32*A^2*a^4*b^6*c^8*d^2*f^2 + 20*A^2*a^7*
b^3*c^7*d^3*f^2 - 15*A^2*a^8*b^2*c^6*d^4*f^2 - 15*A^2*a^6*b^4*c^8*d^2*f^2 -
12*A^2*a^6*b^4*c^4*d^6*f^2 - 12*A^2*a^4*b^6*c^6*d^4*f^2 + 8*B*C*b^10*c^9*d
*f^2 - 16*B*C*a^10*c*d^9*f^2 - 16*A*B*b^10*c^9*d*f^2 - 16*A*B*b^10*c*d^9*f^
2 + 8*B*C*a^9*b*d^10*f^2 - 16*B*C*a*b^9*c^10*f^2 + 16*A*B*a^10*c*d^9*f^2 -
16*A*B*a^9*b*d^10*f^2 - 16*A*B*a*b^9*d^10*f^2 + 16*A*B*a*b^9*c^10*f^2 + 22*
C^2*a^9*b*c^5*d^5*f^2 + 22*C^2*a^5*b^5*c^9*d*f^2 + 22*C^2*a^5*b^5*c*d^9*f^2
+ 22*C^2*a*b^9*c^5*d^5*f^2 - 20*C^2*a^9*b*c^3*d^7*f^2 - 20*C^2*a^7*b^3*c*d
^9*f^2 - 20*C^2*a^3*b^7*c^9*d*f^2 - 20*C^2*a*b^9*c^7*d^3*f^2 + 36*B^2*a^7*b
^3*c*d^9*f^2 + 36*B^2*a*b^9*c^7*d^3*f^2 + 28*B^2*a^9*b*c^3*d^7*f^2 + 28*B^2
*a^3*b^7*c^9*d*f^2 + 24*B^2*a^3*b^7*c*d^9*f^2 + 24*B^2*a*b^9*c^3*d^7*f^2 -
18*B^2*a^9*b*c^5*d^5*f^2 - 18*B^2*a^5*b^5*c^9*d*f^2 + 6*B^2*a^5*b^5*c*d^9*f
^2 + 6*B^2*a*b^9*c^5*d^5*f^2 - 96*A^2*a^3*b^7*c*d^9*f^2 - 96*A^2*a*b^9*c^3*
d^7*f^2 - 90*A^2*a^5*b^5*c*d^9*f^2 - 90*A^2*a*b^9*c^5*d^5*f^2 - 84*A^2*a^7*
b^3*c*d^9*f^2 - 84*A^2*a*b^9*c^7*d^3*f^2 - 52*A^2*a^9*b*c^3*d^7*f^2 - 52*A^
2*a^3*b^7*c^9*d*f^2 + 6*A^2*a^9*b*c^5*d^5*f^2 + 6*A^2*a^5*b^5*c^9*d*f^2 - 1
0*C^2*a^9*b*c*d^9*f^2 - 10*C^2*a*b^9*c^9*d*f^2 + 14*B^2*a^9*b*c*d^9*f^2 + 1
4*B^2*a*b^9*c^9*d*f^2 + 8*B^2*a*b^9*c*d^9*f^2 - 32*A^2*a*b^9*c*d^9*f^2 - 26
*A^2*a^9*b*c*d^9*f^2 - 26*A^2*a*b^9*c^9*d*f^2 + 2*A*C*b^10*c^10*f^2 + 2*A*C
*a^10*d^10*f^2 + 14*C^2*b^10*c^8*d^2*f^2 - C^2*b^10*c^6*d^4*f^2 + 31*B^2*b^
10*c^6*d^4*f^2 + 20*B^2*b^10*c^4*d^6*f^2 + 14*C^2*a^10*c^2*d^8*f^2 + 4*B^2*
b^10*c^2*d^8*f^2 + 2*B^2*b^10*c^8*d^2*f^2 - C^2*a^10*c^4*d^6*f^2 + 80*A^2*b
^10*c^4*d^6*f^2 + 64*A^2*b^10*c^2*d^8*f^2 + 31*A^2*b^10*c^6*d^4*f^2 + 14*C^
2*a^8*b^2*d^10*f^2 + 14*A^2*b^10*c^8*d^2*f^2 - 10*B^2*a^10*c^2*d^8*f^2 + 3*
B^2*a^10*c^4*d^6*f^2 - C^2*a^6*b^4*d^10*f^2 + 31*B^2*a^6*b^4*d^10*f^2 + 20*
B^2*a^4*b^6*d^10*f^2 + 14*C^2*a^2*b^8*c^10*f^2 + 14*A^2*a^10*c^2*d^8*f^2 +
4*B^2*a^2*b^8*d^10*f^2 + 2*B^2*a^8*b^2*d^10*f^2 - C^2*a^4*b^6*c^10*f^2 - A^
2*a^10*c^4*d^6*f^2 + 80*A^2*a^4*b^6*d^10*f^2 + 64*A^2*a^2*b^8*d^10*f^2 + 31
*A^2*a^6*b^4*d^10*f^2 + 14*A^2*a^8*b^2*d^10*f^2 - 10*B^2*a^2*b^8*c^10*f^2 +
3*B^2*a^4*b^6*c^10*f^2 + 14*A^2*a^2*b^8*c^10*f^2 - A^2*a^4*b^6*c^10*f^2 -
C^2*b^10*c^10*f^2 - C^2*a^10*d^10*f^2 + 16*A^2*b^10*d^10*f^2 + 3*B^2*b^10*c
^10*f^2 + 3*B^2*a^10*d^10*f^2 - A^2*b^10*c^10*f^2 - A^2*a^10*d^10*f^2 - 96*
A*B*C*a*b^7*c*d^7*f - 28*A*B*C*a^7*b*c*d^7*f - 28*A*B*C*a*b^7*c^7*d*f + 484
*A*B*C*a^4*b^4*c^4*d^4*f - 424*A*B*C*a^3*b^5*c^3*d^5*f + 320*A*B*C*a^2*b^6*
c^2*d^6*f - 176*A*B*C*a^6*b^2*c^2*d^6*f - 176*A*B*C*a^2*b^6*c^6*d^2*f + 158
*A*B*C*a^4*b^4*c^2*d^6*f + 158*A*B*C*a^2*b^6*c^4*d^4*f - 136*A*B*C*a^5*b^3*
c^5*d^3*f - 34*A*B*C*a^6*b^2*c^4*d^4*f - 34*A*B*C*a^4*b^4*c^6*d^2*f + 28*A*
B*C*a^5*b^3*c^3*d^5*f + 28*A*B*C*a^3*b^5*c^5*d^3*f + 308*A*B*C*a^5*b^3*c*d^
7*f + 308*A*B*C*a*b^7*c^5*d^3*f + 20*A*B*C*a^7*b*c^3*d^5*f + 20*A*B*C*a^3*b
^5*c^7*d*f + 30*B*C^2*a^7*b*c*d^7*f + 30*B*C^2*a*b^7*c^7*d*f + 160*A^2*B*a*
b^7*c*d^7*f - 2*A^2*B*a^7*b*c*d^7*f - 2*A^2*B*a*b^7*c^7*d*f - 96*A*B*C*b^8*
c^4*d^4*f + 34*A*B*C*b^8*c^6*d^2*f - 32*A*B*C*b^8*c^2*d^6*f + 2*A*B*C*a^8*c
^2*d^6*f - 96*A*B*C*a^4*b^4*d^8*f + 34*A*B*C*a^6*b^2*d^8*f - 32*A*B*C*a^2*b

^6*d^8*f + 2*A*B*C*a^2*b^6*c^8*f - 210*B*C^2*a^4*b^4*c^4*d^4*f - 182*B^2*C*
a^5*b^3*c^2*d^6*f - 182*B^2*C*a^2*b^6*c^5*d^3*f + 180*B*C^2*a^5*b^3*c^5*d^3
*f + 180*B*C^2*a^3*b^5*c^3*d^5*f - 166*B^2*C*a^5*b^3*c^4*d^4*f - 166*B^2*C*
a^4*b^4*c^5*d^3*f + 152*B*C^2*a^6*b^2*c^2*d^6*f + 152*B*C^2*a^2*b^6*c^6*d^2
*f - 112*B^2*C*a^3*b^5*c^2*d^6*f - 112*B^2*C*a^2*b^6*c^3*d^5*f + 94*B^2*C*a
^4*b^4*c^3*d^5*f + 94*B^2*C*a^3*b^5*c^4*d^4*f - 80*B*C^2*a^2*b^6*c^2*d^6*f
+ 66*B*C^2*a^5*b^3*c^3*d^5*f + 66*B*C^2*a^3*b^5*c^5*d^3*f + 46*B^2*C*a^6*b^
2*c^3*d^5*f + 46*B^2*C*a^3*b^5*c^6*d^2*f + 33*B*C^2*a^6*b^2*c^4*d^4*f + 33*
B*C^2*a^4*b^4*c^6*d^2*f + 24*B^2*C*a^6*b^2*c^5*d^3*f + 24*B^2*C*a^5*b^3*c^6
*d^2*f - 16*B*C^2*a^6*b^2*c^6*d^2*f - 15*B*C^2*a^4*b^4*c^2*d^6*f - 15*B*C^2
*a^2*b^6*c^4*d^4*f - 190*A^2*C*a^4*b^4*c^3*d^5*f - 190*A^2*C*a^3*b^5*c^4*d^
4*f + 182*A^2*C*a^5*b^3*c^2*d^6*f + 182*A^2*C*a^2*b^6*c^5*d^3*f + 160*A^2*C
*a^3*b^5*c^2*d^6*f + 160*A^2*C*a^2*b^6*c^3*d^5*f - 150*A*C^2*a^5*b^3*c^2*d^
6*f - 150*A*C^2*a^2*b^6*c^5*d^3*f - 126*A*C^2*a^5*b^3*c^4*d^4*f - 126*A*C^2
*a^4*b^4*c^5*d^3*f + 126*A*C^2*a^4*b^4*c^3*d^5*f + 126*A*C^2*a^3*b^5*c^4*d^
4*f - 96*A*C^2*a^3*b^5*c^2*d^6*f - 96*A*C^2*a^2*b^6*c^3*d^5*f + 94*A^2*C*a^
5*b^3*c^4*d^4*f + 94*A^2*C*a^4*b^4*c^5*d^3*f + 54*A*C^2*a^6*b^2*c^3*d^5*f +
54*A*C^2*a^3*b^5*c^6*d^2*f + 32*A*C^2*a^6*b^2*c^5*d^3*f + 32*A*C^2*a^5*b^3
*c^6*d^2*f - 22*A^2*C*a^6*b^2*c^3*d^5*f - 22*A^2*C*a^3*b^5*c^6*d^2*f + 500*
A^2*B*a^3*b^5*c^3*d^5*f - 290*A^2*B*a^4*b^4*c^4*d^4*f - 256*A^2*B*a^2*b^6*c
^2*d^6*f - 230*A*B^2*a^4*b^4*c^3*d^5*f - 230*A*B^2*a^3*b^5*c^4*d^4*f + 142*
A*B^2*a^5*b^3*c^2*d^6*f + 142*A*B^2*a^2*b^6*c^5*d^3*f - 127*A^2*B*a^4*b^4*c
^2*d^6*f - 127*A^2*B*a^2*b^6*c^4*d^4*f + 86*A*B^2*a^5*b^3*c^4*d^4*f + 86*A*
B^2*a^4*b^4*c^5*d^3*f + 80*A*B^2*a^3*b^5*c^2*d^6*f + 80*A*B^2*a^2*b^6*c^3*d
^5*f + 40*A^2*B*a^6*b^2*c^2*d^6*f + 40*A^2*B*a^2*b^6*c^6*d^2*f + 34*A^2*B*a
^5*b^3*c^3*d^5*f + 34*A^2*B*a^3*b^5*c^5*d^3*f - 30*A*B^2*a^6*b^2*c^3*d^5*f
- 30*A*B^2*a^3*b^5*c^6*d^2*f + 20*A^2*B*a^5*b^3*c^5*d^3*f - 15*A^2*B*a^6*b^
2*c^4*d^4*f - 15*A^2*B*a^4*b^4*c^6*d^2*f - 98*B^2*C*a^6*b^2*c*d^7*f - 98*B^
2*C*a*b^7*c^6*d^2*f - 90*B*C^2*a^5*b^3*c*d^7*f - 90*B*C^2*a*b^7*c^5*d^3*f +
48*B^2*C*a^4*b^4*c*d^7*f + 48*B^2*C*a*b^7*c^4*d^4*f + 40*B^2*C*a^2*b^6*c*d
^7*f + 40*B^2*C*a*b^7*c^2*d^6*f - 32*B*C^2*a^3*b^5*c*d^7*f - 32*B*C^2*a*b^7
*c^3*d^5*f + 26*B^2*C*a^7*b*c^2*d^6*f + 26*B^2*C*a^2*b^6*c^7*d*f - 26*B*C^2
*a^7*b*c^3*d^5*f - 26*B*C^2*a^3*b^5*c^7*d*f - 8*B^2*C*a^7*b*c^4*d^4*f - 8*B
^2*C*a^4*b^4*c^7*d*f - 224*A^2*C*a^4*b^4*c*d^7*f - 224*A^2*C*a*b^7*c^4*d^4*
f - 96*A^2*C*a^2*b^6*c*d^7*f - 96*A^2*C*a*b^7*c^2*d^6*f + 96*A*C^2*a^4*b^4*
c*d^7*f + 96*A*C^2*a*b^7*c^4*d^4*f - 66*A*C^2*a^6*b^2*c*d^7*f - 66*A*C^2*a*
b^7*c^6*d^2*f + 64*A*C^2*a^2*b^6*c*d^7*f + 64*A*C^2*a*b^7*c^2*d^6*f + 34*A^
2*C*a^6*b^2*c*d^7*f + 34*A^2*C*a*b^7*c^6*d^2*f + 34*A*C^2*a^7*b*c^2*d^6*f +
34*A*C^2*a^2*b^6*c^7*d*f - 2*A^2*C*a^7*b*c^2*d^6*f - 2*A^2*C*a^2*b^6*c^7*d
*f - 208*A*B^2*a^4*b^4*c*d^7*f - 208*A*B^2*a*b^7*c^4*d^4*f + 160*A^2*B*a^3*
b^5*c*d^7*f + 160*A^2*B*a*b^7*c^3*d^5*f - 154*A^2*B*a^5*b^3*c*d^7*f - 154*A
^2*B*a*b^7*c^5*d^3*f - 112*A*B^2*a^2*b^6*c*d^7*f - 112*A*B^2*a*b^7*c^2*d^6*
f + 58*A*B^2*a^6*b^2*c*d^7*f + 58*A*B^2*a*b^7*c^6*d^2*f - 10*A*B^2*a^7*b*c^
2*d^6*f - 10*A*B^2*a^2*b^6*c^7*d*f + 6*A^2*B*a^7*b*c^3*d^5*f + 6*A^2*B*a^3*
b^5*c^7*d*f + 32*B^2*C*b^8*c^5*d^3*f - 17*B*C^2*b^8*c^6*d^2*f + 8*B^2*C*b^8

```
*c^3*d^5*f + 64*A^2*C*b^8*c^3*d^5*f - 32*A^2*C*b^8*c^5*d^3*f + 32*A*C^2*b^8
*c^5*d^3*f - B*C^2*a^8*c^2*d^6*f + 112*A^2*B*b^8*c^4*d^4*f - 64*A*B^2*b^8*c
^5*d^3*f + 32*B^2*C*a^5*b^3*d^8*f - 17*B*C^2*a^6*b^2*d^8*f + 16*A^2*B*b^8*c
^2*d^6*f + 16*A*B^2*b^8*c^3*d^5*f + 8*B^2*C*a^3*b^5*d^8*f - A^2*B*b^8*c^6*d
^2*f + 64*A^2*C*a^3*b^5*d^8*f - 32*A^2*C*a^5*b^3*d^8*f + 32*A*C^2*a^5*b^3*d
^8*f - A^2*B*a^8*c^2*d^6*f - B*C^2*a^2*b^6*c^8*f + 112*A^2*B*a^4*b^4*d^8*f
- 64*A*B^2*a^5*b^3*d^8*f + 16*A^2*B*a^2*b^6*d^8*f + 16*A*B^2*a^3*b^5*d^8*f
- A^2*B*a^6*b^2*d^8*f - A^2*B*a^2*b^6*c^8*f - 8*B^3*a*b^7*c*d^7*f - 2*B^3*a
^7*b*c*d^7*f - 2*B^3*a*b^7*c^7*d*f - 6*B^2*C*b^8*c^7*d*f + 32*A^2*C*b^8*c*d
^7*f + 6*A^2*C*b^8*c^7*d*f - 6*A*C^2*b^8*c^7*d*f - 2*B^2*C*a^8*c*d^7*f + 16
*A*B^2*b^8*c*d^7*f - 6*B^2*C*a^7*b*d^8*f - 6*A^2*C*a^8*c*d^7*f + 6*A*C^2*a^
8*c*d^7*f - 2*A*B^2*b^8*c^7*d*f + 32*A^2*C*a*b^7*d^8*f + 6*A^2*C*a^7*b*d^8*
f - 6*A*C^2*a^7*b*d^8*f - 2*B^2*C*a*b^7*c^8*f + 2*A*B^2*a^8*c*d^7*f + 16*A*
B^2*a*b^7*d^8*f - 6*A^2*C*a*b^7*c^8*f + 6*A*C^2*a*b^7*c^8*f - 2*A*B^2*a^7*b
*d^8*f + 2*A*B^2*a*b^7*c^8*f - 50*C^3*a^6*b^2*c^3*d^5*f + 50*C^3*a^5*b^3*c^
2*d^6*f - 50*C^3*a^3*b^5*c^6*d^2*f + 50*C^3*a^2*b^6*c^5*d^3*f + 42*C^3*a^5*
b^3*c^4*d^4*f + 42*C^3*a^4*b^4*c^5*d^3*f - 42*C^3*a^4*b^4*c^3*d^5*f - 42*C^
3*a^3*b^5*c^4*d^4*f - 32*C^3*a^6*b^2*c^5*d^3*f - 32*C^3*a^5*b^3*c^6*d^2*f +
 32*C^3*a^3*b^5*c^2*d^6*f + 32*C^3*a^2*b^6*c^3*d^5*f + 94*B^3*a^4*b^4*c^4*d
^4*f + 48*B^3*a^2*b^6*c^2*d^6*f - 44*B^3*a^3*b^5*c^3*d^5*f - 32*B^3*a^6*b^2
*c^2*d^6*f - 32*B^3*a^2*b^6*c^6*d^2*f + 29*B^3*a^4*b^4*c^2*d^6*f + 29*B^3*a
^2*b^6*c^4*d^4*f - 20*B^3*a^5*b^3*c^5*d^3*f + 18*B^3*a^5*b^3*c^3*d^5*f + 18
*B^3*a^3*b^5*c^5*d^3*f - 3*B^3*a^6*b^2*c^4*d^4*f - 3*B^3*a^4*b^4*c^6*d^2*f
+ 106*A^3*a^4*b^4*c^3*d^5*f + 106*A^3*a^3*b^5*c^4*d^4*f - 96*A^3*a^3*b^5*c^
2*d^6*f - 96*A^3*a^2*b^6*c^3*d^5*f - 82*A^3*a^5*b^3*c^2*d^6*f - 82*A^3*a^2*
b^6*c^5*d^3*f + 18*A^3*a^6*b^2*c^3*d^5*f + 18*A^3*a^3*b^5*c^6*d^2*f - 10*A^
3*a^5*b^3*c^4*d^4*f - 10*A^3*a^4*b^4*c^5*d^3*f - 22*C^3*a^7*b*c^2*d^6*f + 2
2*C^3*a^6*b^2*c*d^7*f - 22*C^3*a^2*b^6*c^7*d*f + 22*C^3*a*b^7*c^6*d^2*f - 2
*A*B*C*b^8*c^8*f - 2*A*B*C*a^8*d^8*f + 62*B^3*a^5*b^3*c*d^7*f + 62*B^3*a*b^
7*c^5*d^3*f + 16*B^3*a^3*b^5*c*d^7*f + 16*B^3*a*b^7*c^3*d^5*f + 6*B^3*a^7*b
*c^3*d^5*f + 6*B^3*a^3*b^5*c^7*d*f + 128*A^3*a^4*b^4*c*d^7*f + 128*A^3*a*b^
7*c^4*d^4*f + 32*A^3*a^2*b^6*c*d^7*f + 32*A^3*a*b^7*c^2*d^6*f - 10*A^3*a^7*
b*c^2*d^6*f + 10*A^3*a^6*b^2*c*d^7*f - 10*A^3*a^2*b^6*c^7*d*f + 10*A^3*a*b^
7*c^6*d^2*f + 11*B^3*b^8*c^6*d^2*f - 8*B^3*b^8*c^4*d^4*f - 4*B^3*b^8*c^2*d^
6*f - 64*A^3*b^8*c^3*d^5*f - B^3*a^8*c^2*d^6*f + 11*B^3*a^6*b^2*d^8*f - 8*B
^3*a^4*b^4*d^8*f - 4*B^3*a^2*b^6*d^8*f - 64*A^3*a^3*b^5*d^8*f - B^3*a^2*b^6
*c^8*f + 2*C^3*b^8*c^7*d*f - 2*C^3*a^8*c*d^7*f - 32*A^3*b^8*c*d^7*f + 2*C^3
*a^7*b*d^8*f - 2*A^3*b^8*c^7*d*f - 2*C^3*a*b^7*c^8*f + 2*A^3*a^8*c*d^7*f -
32*A^3*a*b^7*d^8*f - 2*A^3*a^7*b*d^8*f + 2*A^3*a*b^7*c^8*f - 16*A^2*B*b^8*d
^8*f + B*C^2*b^8*c^8*f + B*C^2*a^8*d^8*f + A^2*B*b^8*c^8*f + A^2*B*a^8*d^8*
f + B^3*b^8*c^8*f + B^3*a^8*d^8*f - 4*A*B^2*C*a^5*b*c*d^5 - 4*A*B^2*C*a*b^5
*c^5*d + 4*A*B^2*C*a*b^5*c*d^5 + 22*A^2*B*C*a^3*b^3*c^2*d^4 + 22*A^2*B*C*a^
2*b^4*c^3*d^3 - 20*A*B^2*C*a^3*b^3*c^3*d^3 + 14*A*B^2*C*a^4*b^2*c^2*d^4 + 1
4*A*B^2*C*a^2*b^4*c^4*d^2 - 14*A*B*C^2*a^3*b^3*c^2*d^4 - 14*A*B*C^2*a^2*b^4
*c^3*d^3 + 12*A*B*C^2*a^4*b^2*c^3*d^3 + 12*A*B*C^2*a^3*b^3*c^4*d^2 - 6*A^2*
```

B*C*a^4*b^2*c^3*d^3 - 6*A^2*B*C*a^3*b^3*c^4*d^2 - 4*A*B^2*C*a^2*b^4*c^2*d^4
 + 22*A*B*C^2*a^4*b^2*c*d^5 + 22*A*B*C^2*a*b^5*c^4*d^2 - 20*A^2*B*C*a^4*b^2
*c*d^5 - 20*A^2*B*C*a*b^5*c^4*d^2 + 10*A*B*C^2*a^2*b^4*c*d^5 + 10*A*B*C^2*a
*b^5*c^2*d^4 - 8*A^2*B*C*a^2*b^4*c*d^5 - 8*A^2*B*C*a*b^5*c^2*d^4 + 4*A*B^2*
C*a^3*b^3*c*d^5 + 4*A*B^2*C*a*b^5*c^3*d^3 - 4*A*B*C^2*a^5*b*c^2*d^4 - 4*A*B
*C^2*a^2*b^4*c^5*d + 2*A^2*B*C*a^5*b*c^2*d^4 + 2*A^2*B*C*a^2*b^4*c^5*d - 8*
B^3*C*a^4*b^2*c*d^5 - 8*B^3*C*a*b^5*c^4*d^2 - 8*B*C^3*a^4*b^2*c*d^5 - 8*B*C
^3*a*b^5*c^4*d^2 - 4*B^3*C*a^2*b^4*c*d^5 - 4*B^3*C*a*b^5*c^2*d^4 + 4*B^2*C^
2*a^5*b*c*d^5 + 4*B^2*C^2*a*b^5*c^5*d - 4*B*C^3*a^2*b^4*c*d^5 - 4*B*C^3*a*b
^5*c^2*d^4 + 2*B^3*C*a^5*b*c^2*d^4 + 2*B^3*C*a^2*b^4*c^5*d + 2*B^2*C^2*a*b^
5*c*d^5 + 2*B*C^3*a^5*b*c^2*d^4 + 2*B*C^3*a^2*b^4*c^5*d + 24*A^3*C*a^3*b^3*
c*d^5 + 24*A^3*C*a*b^5*c^3*d^3 - 24*A^2*C^2*a*b^5*c*d^5 + 12*A^2*C^2*a^5*b*
c*d^5 + 12*A^2*C^2*a*b^5*c^5*d + 8*A*C^3*a^3*b^3*c*d^5 + 8*A*C^3*a*b^5*c^3*
d^3 + 6*A^3*B*a^4*b^2*c*d^5 + 6*A^3*B*a*b^5*c^4*d^2 - 6*A^2*B^2*a*b^5*c*d^5
 + 6*A*B^3*a^4*b^2*c*d^5 + 6*A*B^3*a*b^5*c^4*d^2 + 2*A^3*B*a^2*b^4*c*d^5 +
2*A^3*B*a*b^5*c^2*d^4 + 2*A*B^3*a^2*b^4*c*d^5 + 2*A*B^3*a*b^5*c^2*d^4 + 20*
A^2*B*C*b^6*c^3*d^3 - 10*A*B*C^2*b^6*c^3*d^3 - 2*A*B^2*C*b^6*c^4*d^2 - 2*A*
B^2*C*b^6*c^2*d^4 + 20*A^2*B*C*a^3*b^3*d^6 - 10*A*B*C^2*a^3*b^3*d^6 - 2*A*B
^2*C*a^4*b^2*d^6 - 2*A*B^2*C*a^2*b^4*d^6 + 10*B^2*C^2*a^3*b^3*c^3*d^3 + 4*B
^2*C^2*a^4*b^2*c^4*d^2 - 3*B^2*C^2*a^4*b^2*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*
d^2 + 2*B^2*C^2*a^2*b^4*c^2*d^4 + 40*A^2*C^2*a^2*b^4*c^2*d^4 - 16*A^2*C^2*a
^4*b^2*c^2*d^4 - 16*A^2*C^2*a^2*b^4*c^4*d^2 + 4*A^2*C^2*a^4*b^2*c^4*d^2 + 1
8*A^2*B^2*a^2*b^4*c^2*d^4 + 10*A^2*B^2*a^3*b^3*c^3*d^3 - 3*A^2*B^2*a^4*b^2*
c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 24*A^3*C*a*b^5*c*d^5 - 12*A*C^3*a^5*b
*c*d^5 - 12*A*C^3*a*b^5*c^5*d + 8*A*C^3*a*b^5*c*d^5 - 4*A^3*C*a^5*b*c*d^5 -
 4*A^3*C*a*b^5*c^5*d + 8*A^2*B*C*b^6*c*d^5 + 4*A*B*C^2*b^6*c^5*d - 4*A*B*C^
2*b^6*c*d^5 - 2*A^2*B*C*b^6*c^5*d + 8*A^2*B*C*a*b^5*d^6 + 4*A*B*C^2*a^5*b*d
^6 - 4*A*B*C^2*a*b^5*d^6 - 2*A^2*B*C*a^5*b*d^6 - 6*B^3*C*a^4*b^2*c^3*d^3 -
6*B^3*C*a^3*b^3*c^4*d^2 - 6*B*C^3*a^4*b^2*c^3*d^3 - 6*B*C^3*a^3*b^3*c^4*d^2
 + 2*B^3*C*a^3*b^3*c^2*d^4 + 2*B^3*C*a^2*b^4*c^3*d^3 + 2*B^2*C^2*a^3*b^3*c*
d^5 + 2*B^2*C^2*a*b^5*c^3*d^3 + 2*B*C^3*a^3*b^3*c^2*d^4 + 2*B*C^3*a^2*b^4*c
^3*d^3 - 48*A^3*C*a^2*b^4*c^2*d^4 - 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A^2*C^2*a
*b^5*c^3*d^3 - 16*A*C^3*a^2*b^4*c^2*d^4 + 8*A^3*C*a^4*b^2*c^2*d^4 + 8*A^3*C
*a^2*b^4*c^4*d^2 - 8*A*C^3*a^4*b^2*c^4*d^2 + 8*A*C^3*a^4*b^2*c^2*d^4 + 8*A*
C^3*a^2*b^4*c^4*d^2 - 10*A^3*B*a^3*b^3*c^2*d^4 - 10*A^3*B*a^2*b^4*c^3*d^3 -
 10*A*B^3*a^3*b^3*c^2*d^4 - 10*A*B^3*a^2*b^4*c^3*d^3 - 6*A^2*B^2*a^3*b^3*c*
d^5 - 6*A^2*B^2*a*b^5*c^3*d^3 + 3*B^2*C^2*b^6*c^4*d^2 - 8*A^2*C^2*b^6*c^4*d
^2 + 8*A^2*C^2*b^6*c^2*d^4 + 9*A^2*B^2*b^6*c^2*d^4 + 3*B^2*C^2*a^4*b^2*d^6
+ 3*A^2*B^2*b^6*c^4*d^2 - 8*A^2*C^2*a^4*b^2*d^6 + 8*A^2*C^2*a^2*b^4*d^6 + 9
*A^2*B^2*a^2*b^4*d^6 + 3*A^2*B^2*a^4*b^2*d^6 + 2*B^4*a^3*b^3*c*d^5 + 2*B^4*
a*b^5*c^3*d^3 - 8*A^4*a^3*b^3*c*d^5 - 8*A^4*a*b^5*c^3*d^3 - 16*A^3*C*b^6*c^
2*d^4 + 4*A^3*C*b^6*c^4*d^2 + 4*A*C^3*b^6*c^4*d^2 - 10*A^3*B*b^6*c^3*d^3 -
10*A*B^3*b^6*c^3*d^3 - 16*A^3*C*a^2*b^4*d^6 + 4*A^3*C*a^4*b^2*d^6 + 4*A*C^3
*a^4*b^2*d^6 - 10*A^3*B*a^3*b^3*d^6 - 10*A*B^3*a^3*b^3*d^6 + 4*C^4*a^5*b*c*
d^5 + 4*C^4*a*b^5*c^5*d + 2*B^4*a*b^5*c*d^5 - 8*A^4*a*b^5*c*d^5 - 2*B^3*C*b

^6*c^5*d - 2*B*C^3*b^6*c^5*d - 4*A^3*B*b^6*c*d^5 - 4*A*B^3*b^6*c*d^5 - 2*B^
3*C*a^5*b*d^6 - 2*B*C^3*a^5*b*d^6 - 4*A^3*B*a*b^5*d^6 - 4*A*B^3*a*b^5*d^6 +
 4*C^4*a^4*b^2*c^4*d^2 + 4*C^4*a^2*b^4*c^2*d^4 + 10*B^4*a^3*b^3*c^3*d^3 - 3
*B^4*a^4*b^2*c^2*d^4 - 3*B^4*a^2*b^4*c^4*d^2 - 2*B^4*a^2*b^4*c^2*d^4 + 20*A
^4*a^2*b^4*c^2*d^4 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^2*b^4*d^6 - 8*A^3*C*b^
6*d^6 + 3*B^4*b^6*c^4*d^2 + 8*A^4*b^6*c^2*d^4 + 3*B^4*a^4*b^2*d^6 + 8*A^4*a
^2*b^4*d^6 + 4*A^2*C^2*b^6*d^6 + 4*A^2*B^2*b^6*d^6 + 4*A^4*b^6*d^6 + B^4*b^
6*c^2*d^4 + B^4*a^2*b^4*d^6, f, k)*((4*a^5*b^8*d^13 + 4*a^7*b^6*d^13 - 4*a^
9*b^4*d^13 - 4*a^11*b^2*d^13 + 4*b^13*c^5*d^8 + 4*b^13*c^7*d^6 - 4*b^13*c^9
*d^4 - 4*b^13*c^11*d^2 - 12*a*b^12*c^4*d^9 + 4*a*b^12*c^6*d^7 + 52*a*b^12*c
^8*d^5 + 44*a*b^12*c^10*d^3 + 16*a^3*b^10*c^12*d - 12*a^4*b^9*c*d^12 + 8*a^
5*b^8*c^12*d + 4*a^6*b^7*c*d^12 + 52*a^8*b^5*c*d^12 + 44*a^10*b^3*c*d^12 +
16*a^12*b*c^3*d^10 + 8*a^12*b*c^5*d^8 + 8*a^2*b^11*c^3*d^10 - 36*a^2*b^11*c
^5*d^8 - 140*a^2*b^11*c^7*d^6 - 140*a^2*b^11*c^9*d^4 - 44*a^2*b^11*c^11*d^2
 + 8*a^3*b^10*c^2*d^11 + 28*a^3*b^10*c^4*d^9 + 148*a^3*b^10*c^6*d^7 + 260*a
^3*b^10*c^8*d^5 + 148*a^3*b^10*c^10*d^3 + 28*a^4*b^9*c^3*d^10 - 56*a^4*b^9*
c^5*d^8 - 320*a^4*b^9*c^7*d^6 - 300*a^4*b^9*c^9*d^4 - 76*a^4*b^9*c^11*d^2 -
 36*a^5*b^8*c^2*d^11 - 56*a^5*b^8*c^4*d^9 + 160*a^5*b^8*c^6*d^7 + 332*a^5*b
^8*c^8*d^5 + 164*a^5*b^8*c^10*d^3 + 148*a^6*b^7*c^3*d^10 + 160*a^6*b^7*c^5*
d^8 - 144*a^6*b^7*c^7*d^6 - 196*a^6*b^7*c^9*d^4 - 36*a^6*b^7*c^11*d^2 - 140
*a^7*b^6*c^2*d^11 - 320*a^7*b^6*c^4*d^9 - 144*a^7*b^6*c^6*d^7 + 92*a^7*b^6*
c^8*d^5 + 60*a^7*b^6*c^10*d^3 + 260*a^8*b^5*c^3*d^10 + 332*a^8*b^5*c^5*d^8
 + 92*a^8*b^5*c^7*d^6 - 32*a^8*b^5*c^9*d^4 - 140*a^9*b^4*c^2*d^11 - 300*a^9*
b^4*c^4*d^9 - 196*a^9*b^4*c^6*d^7 - 32*a^9*b^4*c^8*d^5 + 148*a^10*b^3*c^3*d
^10 + 164*a^10*b^3*c^5*d^8 + 60*a^10*b^3*c^7*d^6 - 44*a^11*b^2*c^2*d^11 - 7
6*a^11*b^2*c^4*d^9 - 36*a^11*b^2*c^6*d^7 + 8*a*b^12*c^12*d + 8*a^12*b*c*d^1
2)/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b
^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^
7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3
*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^
2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a
^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d
^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*
b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7*c^7*d - 4*a^7*b*c*d^7) + (tan(e +
 f*x)*(6*a^12*b*d^13 + 6*b^13*c^12*d + 8*a^4*b^9*d^13 + 22*a^6*b^7*d^13 + 2
6*a^8*b^5*d^13 + 18*a^10*b^3*d^13 + 8*b^13*c^4*d^9 + 22*b^13*c^6*d^7 + 26*b
^13*c^8*d^5 + 18*b^13*c^10*d^3 - 32*a*b^12*c^3*d^10 - 84*a*b^12*c^5*d^8 - 9
2*a*b^12*c^7*d^6 - 60*a*b^12*c^9*d^4 - 20*a*b^12*c^11*d^2 + 10*a^2*b^11*c^1
2*d - 32*a^3*b^10*c*d^12 + 2*a^4*b^9*c^12*d - 84*a^5*b^8*c*d^12 - 2*a^6*b^7
*c^12*d - 92*a^7*b^6*c*d^12 - 60*a^9*b^4*c*d^12 - 20*a^11*b^2*c*d^12 + 10*a
^12*b*c^2*d^11 + 2*a^12*b*c^4*d^9 - 2*a^12*b*c^6*d^7 + 48*a^2*b^11*c^2*d^11
 + 138*a^2*b^11*c^4*d^9 + 152*a^2*b^11*c^6*d^7 + 92*a^2*b^11*c^8*d^5 + 40*a
^2*b^11*c^10*d^3 - 152*a^3*b^10*c^3*d^10 - 196*a^3*b^10*c^5*d^8 - 92*a^3*b^
10*c^7*d^6 - 44*a^3*b^10*c^9*d^4 - 28*a^3*b^10*c^11*d^2 + 138*a^4*b^9*c^2*d
^11 + 220*a^4*b^9*c^4*d^9 + 50*a^4*b^9*c^6*d^7 - 46*a^4*b^9*c^8*d^5 - 4*a^4

*b^9*c^10*d^3 - 196*a^5*b^8*c^3*d^10 - 16*a^5*b^8*c^5*d^8 + 224*a^5*b^8*c^7
*d^6 + 132*a^5*b^8*c^9*d^4 + 4*a^5*b^8*c^11*d^2 + 152*a^6*b^7*c^2*d^11 + 50
*a^6*b^7*c^4*d^9 - 320*a^6*b^7*c^6*d^7 - 294*a^6*b^7*c^8*d^5 - 56*a^6*b^7*c
^10*d^3 - 92*a^7*b^6*c^3*d^10 + 224*a^7*b^6*c^5*d^8 + 368*a^7*b^6*c^7*d^6 +
156*a^7*b^6*c^9*d^4 + 12*a^7*b^6*c^11*d^2 + 92*a^8*b^5*c^2*d^11 - 46*a^8*b
^5*c^4*d^9 - 294*a^8*b^5*c^6*d^7 - 212*a^8*b^5*c^8*d^5 - 30*a^8*b^5*c^10*d^
3 - 44*a^9*b^4*c^3*d^10 + 132*a^9*b^4*c^5*d^8 + 156*a^9*b^4*c^7*d^6 + 40*a^
9*b^4*c^9*d^4 + 40*a^10*b^3*c^2*d^11 - 4*a^10*b^3*c^4*d^9 - 56*a^10*b^3*c^6
*d^7 - 30*a^10*b^3*c^8*d^5 - 28*a^11*b^2*c^3*d^10 + 4*a^11*b^2*c^5*d^8 + 12
*a^11*b^2*c^7*d^6))/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a^4*b^4*c^8 + a^4*
b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^8*c^4*d^4 + 2*b^8
*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c*d^7 - 8*a^3*b^5*
c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d
^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c^6*d^2 - 16*a^3*b
^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*a^4*b^4*c^4*d^4 +
14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*d^3 + 10*a^6*b^2*
c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7*c^7*d - 4*a^7*b*
c*d^7)) - (C*a^10*b*d^11 - A*b^11*c^10*d - A*a^10*b*d^11 + C*b^11*c^10*d -
8*A*a^2*b^9*d^11 - 16*A*a^4*b^7*d^11 - A*a^6*b^5*d^11 + 6*A*a^8*b^3*d^11 +
4*B*a^3*b^8*d^11 + 12*B*a^5*b^6*d^11 + 4*B*a^7*b^4*d^11 - 4*B*a^9*b^2*d^11
- 8*A*b^11*c^2*d^9 - 16*A*b^11*c^4*d^7 - A*b^11*c^6*d^5 + 6*A*b^11*c^8*d^3
- 7*C*a^6*b^5*d^11 - 6*C*a^8*b^3*d^11 + 4*B*b^11*c^3*d^8 + 12*B*b^11*c^5*d^
6 + 4*B*b^11*c^7*d^4 - 4*B*b^11*c^9*d^2 - 7*C*b^11*c^6*d^5 - 6*C*b^11*c^8*d
^3 + 56*A*a*b^10*c^3*d^8 + 54*A*a*b^10*c^5*d^6 + 12*A*a*b^10*c^7*d^4 - 2*A*
a*b^10*c^9*d^2 - 2*A*a^2*b^9*c^10*d + 56*A*a^3*b^8*c*d^10 - A*a^4*b^7*c^10*
d + 54*A*a^5*b^6*c*d^10 + 12*A*a^7*b^4*c*d^10 - 2*A*a^9*b^2*c*d^10 - 2*A*a^
10*b*c^2*d^9 - A*a^10*b*c^4*d^7 - 4*B*a*b^10*c^2*d^9 - 32*B*a*b^10*c^4*d^7
- 44*B*a*b^10*c^6*d^5 - 16*B*a*b^10*c^8*d^3 - 4*B*a^2*b^9*c*d^10 - 32*B*a^4
*b^7*c*d^10 - 44*B*a^6*b^5*c*d^10 - 16*B*a^8*b^3*c*d^10 - 8*C*a*b^10*c^3*d^
8 + 2*C*a*b^10*c^5*d^6 + 20*C*a*b^10*c^7*d^4 + 10*C*a*b^10*c^9*d^2 + 2*C*a^
2*b^9*c^10*d - 8*C*a^3*b^8*c*d^10 + C*a^4*b^7*c^10*d + 2*C*a^5*b^6*c*d^10 +
20*C*a^7*b^4*c*d^10 + 10*C*a^9*b^2*c*d^10 + 2*C*a^10*b*c^2*d^9 + C*a^10*b*
c^4*d^7 - 80*A*a^2*b^9*c^2*d^9 - 159*A*a^2*b^9*c^4*d^7 - 80*A*a^2*b^9*c^6*d
^5 + 5*A*a^2*b^9*c^8*d^3 + 212*A*a^3*b^8*c^3*d^8 + 228*A*a^3*b^8*c^5*d^6 +
76*A*a^3*b^8*c^7*d^4 + 4*A*a^3*b^8*c^9*d^2 - 159*A*a^4*b^7*c^2*d^9 - 332*A*
a^4*b^7*c^4*d^7 - 204*A*a^4*b^7*c^6*d^5 - 16*A*a^4*b^7*c^8*d^3 + 228*A*a^5*
b^6*c^3*d^8 + 252*A*a^5*b^6*c^5*d^6 + 84*A*a^5*b^6*c^7*d^4 + 6*A*a^5*b^6*c^
9*d^2 - 80*A*a^6*b^5*c^2*d^9 - 204*A*a^6*b^5*c^4*d^7 - 140*A*a^6*b^5*c^6*d^
5 - 15*A*a^6*b^5*c^8*d^3 + 76*A*a^7*b^4*c^3*d^8 + 84*A*a^7*b^4*c^5*d^6 + 20
*A*a^7*b^4*c^7*d^4 + 5*A*a^8*b^3*c^2*d^9 - 16*A*a^8*b^3*c^4*d^7 - 15*A*a^8*
b^3*c^6*d^5 + 4*A*a^9*b^2*c^3*d^8 + 6*A*a^9*b^2*c^5*d^6 + 20*B*a^2*b^9*c^3*
d^8 + 84*B*a^2*b^9*c^5*d^6 + 60*B*a^2*b^9*c^7*d^4 + 20*B*a^3*b^8*c^2*d^9 -
44*B*a^3*b^8*c^4*d^7 - 100*B*a^3*b^8*c^6*d^5 - 40*B*a^3*b^8*c^8*d^3 - 44*B*
a^4*b^7*c^3*d^8 + 60*B*a^4*b^7*c^5*d^6 + 76*B*a^4*b^7*c^7*d^4 + 4*B*a^4*b^7
*c^9*d^2 + 84*B*a^5*b^6*c^2*d^9 + 60*B*a^5*b^6*c^4*d^7 - 36*B*a^5*b^6*c^6*d

^5 - 24*B*a^5*b^6*c^8*d^3 - 100*B*a^6*b^5*c^3*d^8 - 36*B*a^6*b^5*c^5*d^6 + 20*B*a^6*b^5*c^7*d^4 + 60*B*a^7*b^4*c^2*d^9 + 76*B*a^7*b^4*c^4*d^7 + 20*B*a ^7*b^4*c^6*d^5 - 40*B*a^8*b^3*c^3*d^8 - 24*B*a^8*b^3*c^5*d^6 + 4*B*a^9*b^2* c^4*d^7 + 16*C*a^2*b^9*c^2*d^9 + 47*C*a^2*b^9*c^4*d^7 + 16*C*a^2*b^9*c^6*d^ 5 - 13*C*a^2*b^9*c^8*d^3 - 84*C*a^3*b^8*c^3*d^8 - 100*C*a^3*b^8*c^5*d^6 - 1 2*C*a^3*b^8*c^7*d^4 + 12*C*a^3*b^8*c^9*d^2 + 47*C*a^4*b^7*c^2*d^9 + 140*C*a ^4*b^7*c^4*d^7 + 92*C*a^4*b^7*c^6*d^5 - 100*C*a^5*b^6*c^3*d^8 - 156*C*a^5*b ^6*c^5*d^6 - 52*C*a^5*b^6*c^7*d^4 + 2*C*a^5*b^6*c^9*d^2 + 16*C*a^6*b^5*c^2* d^9 + 92*C*a^6*b^5*c^4*d^7 + 76*C*a^6*b^5*c^6*d^5 + 7*C*a^6*b^5*c^8*d^3 - 1 2*C*a^7*b^4*c^3*d^8 - 52*C*a^7*b^4*c^5*d^6 - 20*C*a^7*b^4*c^7*d^4 - 13*C*a^ 8*b^3*c^2*d^9 + 7*C*a^8*b^3*c^6*d^5 + 12*C*a^9*b^2*c^3*d^8 + 2*C*a^9*b^2*c^ 5*d^6 + 16*A*a*b^10*c*d^10)/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a^4*b^4*c^ 8 + a^4*b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c*d^7 - 8* a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^5 - 4*a^7 *b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*a^4*b^4*c ^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*d^3 + 10* a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7*c^7*d - 4*a^7*b*c*d^7) + (tan(e + f*x)*(3*B*a^10*b*d^11 + 3*B*b^11*c^10*d - 16*A*a^ 3*b^8*d^11 - 48*A*a^5*b^6*d^11 - 36*A*a^7*b^4*d^11 - 4*A*a^9*b^2*d^11 + 4*B *a^4*b^7*d^11 + 23*B*a^6*b^5*d^11 + 22*B*a^8*b^3*d^11 - 16*A*b^11*c^3*d^8 - 48*A*b^11*c^5*d^6 - 36*A*b^11*c^7*d^4 - 4*A*b^11*c^9*d^2 + 8*C*a^5*b^6*d^1 1 + 4*C*a^7*b^4*d^11 - 4*C*a^9*b^2*d^11 + 4*B*b^11*c^4*d^7 + 23*B*b^11*c^6* d^5 + 22*B*b^11*c^8*d^3 + 8*C*b^11*c^5*d^6 + 4*C*b^11*c^7*d^4 - 4*C*b^11*c^ 9*d^2 + 16*A*a*b^10*c^2*d^9 + 80*A*a*b^10*c^4*d^7 + 100*A*a*b^10*c^6*d^5 + 40*A*a*b^10*c^8*d^3 + 16*A*a^2*b^9*c*d^10 + 4*A*a^3*b^8*c^10*d + 80*A*a^4*b ^7*c*d^10 + 100*A*a^6*b^5*c*d^10 + 40*A*a^8*b^3*c*d^10 + 4*A*a^10*b*c^3*d^8 + 16*B*a*b^10*c^3*d^8 + 6*B*a*b^10*c^5*d^6 - 20*B*a*b^10*c^7*d^4 - 10*B*a* b^10*c^9*d^2 + 2*B*a^2*b^9*c^10*d + 16*B*a^3*b^8*c*d^10 - B*a^4*b^7*c^10*d + 6*B*a^5*b^6*c*d^10 - 20*B*a^7*b^4*c*d^10 - 10*B*a^9*b^2*c*d^10 + 2*B*a^10 *b*c^2*d^9 - B*a^10*b*c^4*d^7 - 40*C*a*b^10*c^4*d^7 - 68*C*a*b^10*c^6*d^5 - 32*C*a*b^10*c^8*d^3 - 4*C*a^3*b^8*c^10*d - 40*C*a^4*b^7*c*d^10 - 68*C*a^6* b^5*c*d^10 - 32*C*a^8*b^3*c*d^10 - 4*C*a^10*b*c^3*d^8 - 32*A*a^2*b^9*c^3*d^ 8 - 180*A*a^2*b^9*c^5*d^6 - 156*A*a^2*b^9*c^7*d^4 - 24*A*a^2*b^9*c^9*d^2 - 32*A*a^3*b^8*c^2*d^9 + 116*A*a^3*b^8*c^4*d^7 + 204*A*a^3*b^8*c^6*d^5 + 76*A *a^3*b^8*c^8*d^3 + 116*A*a^4*b^7*c^3*d^8 - 84*A*a^4*b^7*c^5*d^6 - 140*A*a^4 *b^7*c^7*d^4 - 20*A*a^4*b^7*c^9*d^2 - 180*A*a^5*b^6*c^2*d^9 - 84*A*a^5*b^6* c^4*d^7 + 84*A*a^5*b^6*c^6*d^5 + 36*A*a^5*b^6*c^8*d^3 + 204*A*a^6*b^5*c^3*d ^8 + 84*A*a^6*b^5*c^5*d^6 - 20*A*a^6*b^5*c^7*d^4 - 156*A*a^7*b^4*c^2*d^9 - 140*A*a^7*b^4*c^4*d^7 - 20*A*a^7*b^4*c^6*d^5 + 76*A*a^8*b^3*c^3*d^8 + 36*A* a^8*b^3*c^5*d^6 - 24*A*a^9*b^2*c^2*d^9 - 20*A*a^9*b^2*c^4*d^7 - 40*B*a^2*b^ 9*c^2*d^9 - 103*B*a^2*b^9*c^4*d^7 - 40*B*a^2*b^9*c^6*d^5 + 25*B*a^2*b^9*c^8 *d^3 + 148*B*a^3*b^8*c^3*d^8 + 180*B*a^3*b^8*c^5*d^6 + 44*B*a^3*b^8*c^7*d^4 - 4*B*a^3*b^8*c^9*d^2 - 103*B*a^4*b^7*c^2*d^9 - 284*B*a^4*b^7*c^4*d^7 - 18

8*B*a^4*b^7*c^6*d^5 - 12*B*a^4*b^7*c^8*d^3 + 180*B*a^5*b^6*c^3*d^8 + 252*B*a^5*b^6*c^5*d^6 + 84*B*a^5*b^6*c^7*d^4 + 6*B*a^5*b^6*c^9*d^2 - 40*B*a^6*b^5*c^2*d^9 - 188*B*a^6*b^5*c^4*d^7 - 140*B*a^6*b^5*c^6*d^5 - 15*B*a^6*b^5*c^8*d^3 + 44*B*a^7*b^4*c^3*d^8 + 84*B*a^7*b^4*c^5*d^6 + 20*B*a^7*b^4*c^7*d^4 + 25*B*a^8*b^3*c^2*d^9 - 12*B*a^8*b^3*c^4*d^7 - 15*B*a^8*b^3*c^6*d^5 - 4*B*a^9*b^2*c^3*d^8 + 6*B*a^9*b^2*c^5*d^6 + 32*C*a^2*b^9*c^3*d^8 + 116*C*a^2*b^9*c^5*d^6 + 92*C*a^2*b^9*c^7*d^4 + 8*C*a^2*b^9*c^9*d^2 + 32*C*a^3*b^8*c^2*d^9 - 52*C*a^3*b^8*c^4*d^7 - 140*C*a^3*b^8*c^6*d^5 - 60*C*a^3*b^8*c^8*d^3 - 52*C*a^4*b^7*c^3*d^8 + 84*C*a^4*b^7*c^5*d^6 + 108*C*a^4*b^7*c^7*d^4 + 12*C*a^4*b^7*c^9*d^2 + 116*C*a^5*b^6*c^2*d^9 + 84*C*a^5*b^6*c^4*d^7 - 52*C*a^5*b^6*c^6*d^5 - 28*C*a^5*b^6*c^8*d^3 - 140*C*a^6*b^5*c^3*d^8 - 52*C*a^6*b^5*c^5*d^6 + 20*C*a^6*b^5*c^7*d^4 + 92*C*a^7*b^4*c^2*d^9 + 108*C*a^7*b^4*c^4*d^7 + 20*C*a^7*b^4*c^6*d^5 - 60*C*a^8*b^3*c^3*d^8 - 28*C*a^8*b^3*c^5*d^6 + 8*C*a^9*b^2*c^2*d^9 + 12*C*a^9*b^2*c^4*d^7 + 4*A*a*b^10*c^10*d + 4*A*a^10*b*c*d^10 - 4*C*a*b^10*c^10*d - 4*C*a^10*b*c*d^10))/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2 - 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7*c^7*d - 4*a^7*b*c*d^7)) + (tan(e + f*x)*(8*A^2*b^9*d^9 + 8*A^2*a^2*b^7*d^9 + 18*A^2*a^4*b^5*d^9 + 2*A^2*a^6*b^3*d^9 + 2*B^2*a^2*b^7*d^9 - 6*B^2*a^4*b^5*d^9 + 9*B^2*a^6*b^3*d^9 + 8*A^2*b^9*c^2*d^7 + 18*A^2*b^9*c^4*d^5 + 2*A^2*b^9*c^6*d^3 + 2*C^2*a^4*b^5*d^9 - 14*C^2*a^6*b^3*d^9 + 2*B^2*b^9*c^2*d^7 - 6*B^2*b^9*c^4*d^5 + 9*B^2*b^9*c^6*d^3 + 2*C^2*b^9*c^4*d^5 - 14*C^2*b^9*c^6*d^3 + A^2*a^8*b*d^9 + A^2*b^9*c^8*d + C^2*a^8*b*d^9 + C^2*b^9*c^8*d + 28*A^2*a^2*b^7*c^2*d^7 + 54*A^2*a^2*b^7*c^4*d^5 + 6*A^2*a^2*b^7*c^6*d^3 - 96*A^2*a^3*b^6*c^3*d^6 - 20*A^2*a^3*b^6*c^5*d^4 + 54*A^2*a^4*b^5*c^2*d^7 + 42*A^2*a^4*b^5*c^4*d^5 - 20*A^2*a^5*b^4*c^3*d^6 + 6*A^2*a^6*b^3*c^2*d^7 - 20*B^2*a^2*b^7*c^2*d^7 - 37*B^2*a^2*b^7*c^4*d^5 + 14*B^2*a^2*b^7*c^6*d^3 - 4*B^2*a^3*b^6*c^3*d^6 - 14*B^2*a^3*b^6*c^5*d^4 - 6*B^2*a^3*b^6*c^7*d^2 - 37*B^2*a^4*b^5*c^2*d^7 - 28*B^2*a^4*b^5*c^4*d^5 + 9*B^2*a^4*b^5*c^6*d^3 - 14*B^2*a^5*b^4*c^3*d^6 + 14*B^2*a^6*b^3*c^2*d^7 + 9*B^2*a^6*b^3*c^4*d^5 - 6*B^2*a^7*b^2*c^3*d^6 + 20*C^2*a^2*b^7*c^2*d^7 + 22*C^2*a^2*b^7*c^4*d^5 - 26*C^2*a^2*b^7*c^6*d^3 - 48*C^2*a^3*b^6*c^3*d^6 - 28*C^2*a^3*b^6*c^5*d^4 + 8*C^2*a^3*b^6*c^7*d^2 + 22*C^2*a^4*b^5*c^2*d^7 + 18*C^2*a^4*b^5*c^4*d^5 - 8*C^2*a^4*b^5*c^6*d^3 - 28*C^2*a^5*b^4*c^3*d^6 - 32*C^2*a^5*b^4*c^5*d^4 - 26*C^2*a^6*b^3*c^2*d^7 - 8*C^2*a^6*b^3*c^4*d^5 + 8*C^2*a^6*b^3*c^6*d^3 + 8*C^2*a^7*b^2*c^3*d^6 + 4*A*B*a^3*b^6*d^9 - 20*A*B*a^5*b^4*d^9 + 2*A*B*a^7*b^2*d^9 - 28*A*C*a^4*b^5*d^9 + 4*A*C*a^6*b^3*d^9 + 4*A*B*b^9*c^3*d^6 - 20*A*B*b^9*c^5*d^4 + 2*A*B*b^9*c^7*d^2 + 28*B*C*a^5*b^4*d^9 - 6*B*C*a^7*b^2*d^9 - 28*A*C*b^9*c^4*d^5 + 4*A*C*b^9*c^6*d^3 + 28*B*C*b^9*c^5*d^4 - 6*B*C*b^9*c^7*d^2 - 48*A^2*a*b^8*c*d^8 + 4*B^2*a*b^8*c*d^8 - 72*A^2*a*b^8*c^3*d^6 - 24*A^2*

a*b^8*c^5*d^4 - 4*A^2*a*b^8*c^7*d^2 - 72*A^2*a^3*b^6*c*d^8 - 24*A^2*a^5*b^4
*c*d^8 - 4*A^2*a^7*b^2*c*d^8 - 10*B^2*a*b^8*c^5*d^4 - 2*B^2*a*b^8*c^7*d^2 +
B^2*a^2*b^7*c^8*d - 10*B^2*a^5*b^4*c*d^8 - 2*B^2*a^7*b^2*c*d^8 + B^2*a^8*b
*c^2*d^7 - 8*C^2*a*b^8*c^3*d^6 + 4*C^2*a*b^8*c^7*d^2 - 8*C^2*a^3*b^6*c*d^8
+ 4*C^2*a^7*b^2*c*d^8 - 8*A*B*a*b^8*d^9 - 2*A*C*a^8*b*d^9 - 8*A*B*b^9*c*d^8
- 2*A*C*b^9*c^8*d - 2*A*B*a*b^8*c^8*d - 2*A*B*a^8*b*c*d^8 + 16*A*C*a*b^8*c
*d^8 + 2*B*C*a*b^8*c^8*d + 2*B*C*a^8*b*c*d^8 + 28*A*B*a*b^8*c^2*d^7 + 48*A*
B*a*b^8*c^4*d^5 + 2*A*B*a*b^8*c^6*d^3 + 28*A*B*a^2*b^7*c*d^8 + 48*A*B*a^4*b
^5*c*d^8 + 2*A*B*a^6*b^3*c*d^8 + 16*A*C*a*b^8*c^3*d^6 - 8*A*C*a*b^8*c^5*d^4
+ 16*A*C*a^3*b^6*c*d^8 - 8*A*C*a^5*b^4*c*d^8 - 8*B*C*a*b^8*c^2*d^7 - 24*B*
C*a*b^8*c^4*d^5 - 6*B*C*a*b^8*c^6*d^3 - 8*B*C*a^2*b^7*c*d^8 - 24*B*C*a^4*b^
5*c*d^8 - 6*B*C*a^6*b^3*c*d^8 + 52*A*B*a^2*b^7*c^3*d^6 - 22*A*B*a^2*b^7*c^5
*d^4 + 10*A*B*a^2*b^7*c^7*d^2 + 52*A*B*a^3*b^6*c^2*d^7 + 50*A*B*a^3*b^6*c^4
*d^5 - 6*A*B*a^3*b^6*c^6*d^3 + 50*A*B*a^4*b^5*c^3*d^6 - 10*A*B*a^4*b^5*c^5*
d^4 - 22*A*B*a^5*b^4*c^2*d^7 - 10*A*B*a^5*b^4*c^4*d^5 - 6*A*B*a^6*b^3*c^3*d
^6 + 10*A*B*a^7*b^2*c^2*d^7 - 40*A*C*a^2*b^7*c^2*d^7 - 84*A*C*a^2*b^7*c^4*d
^5 + 12*A*C*a^2*b^7*c^6*d^3 + 16*A*C*a^3*b^6*c^3*d^6 - 16*A*C*a^3*b^6*c^5*d
^4 - 8*A*C*a^3*b^6*c^7*d^2 - 84*A*C*a^4*b^5*c^2*d^7 - 52*A*C*a^4*b^5*c^4*d^
5 + 16*A*C*a^4*b^5*c^6*d^3 - 16*A*C*a^5*b^4*c^3*d^6 + 12*A*C*a^6*b^3*c^2*d^
7 + 16*A*C*a^6*b^3*c^4*d^5 - 8*A*C*a^7*b^2*c^3*d^6 + 28*B*C*a^2*b^7*c^3*d^6
+ 82*B*C*a^2*b^7*c^5*d^4 - 10*B*C*a^2*b^7*c^7*d^2 + 28*B*C*a^3*b^6*c^2*d^7
+ 10*B*C*a^3*b^6*c^4*d^5 - 10*B*C*a^3*b^6*c^6*d^3 + 10*B*C*a^4*b^5*c^3*d^6
+ 50*B*C*a^4*b^5*c^5*d^4 + 4*B*C*a^4*b^5*c^7*d^2 + 82*B*C*a^5*b^4*c^2*d^7
+ 50*B*C*a^5*b^4*c^4*d^5 - 12*B*C*a^5*b^4*c^6*d^3 - 10*B*C*a^6*b^3*c^3*d^6
- 12*B*C*a^6*b^3*c^5*d^4 - 10*B*C*a^7*b^2*c^2*d^7 + 4*B*C*a^7*b^2*c^4*d^5))
/(a^8*d^8 + b^8*c^8 + 2*a^2*b^6*c^8 + a^4*b^4*c^8 + a^4*b^4*d^8 + 2*a^6*b^2
*d^8 + 2*a^8*c^2*d^6 + a^8*c^4*d^4 + b^8*c^4*d^4 + 2*b^8*c^6*d^2 - 4*a*b^7*
c^3*d^5 - 8*a*b^7*c^5*d^3 - 4*a^3*b^5*c*d^7 - 8*a^3*b^5*c^7*d - 8*a^5*b^3*c
*d^7 - 4*a^5*b^3*c^7*d - 8*a^7*b*c^3*d^5 - 4*a^7*b*c^5*d^3 + 6*a^2*b^6*c^2*
d^6 + 14*a^2*b^6*c^4*d^4 + 10*a^2*b^6*c^6*d^2 - 16*a^3*b^5*c^3*d^5 - 20*a^3
*b^5*c^5*d^3 + 14*a^4*b^4*c^2*d^6 + 26*a^4*b^4*c^4*d^4 + 14*a^4*b^4*c^6*d^2
- 20*a^5*b^3*c^3*d^5 - 16*a^5*b^3*c^5*d^3 + 10*a^6*b^2*c^2*d^6 + 14*a^6*b^
2*c^4*d^4 + 6*a^6*b^2*c^6*d^2 - 4*a*b^7*c^7*d - 4*a^7*b*c*d^7)))*root(144*a
^13*b*c^5*d^9*f^4 + 144*a^9*b^5*c*d^13*f^4 + 144*a^5*b^9*c^13*d*f^4 + 144*a
*b^13*c^9*d^5*f^4 + 96*a^13*b*c^7*d^7*f^4 + 96*a^13*b*c^3*d^11*f^4 + 96*a^1
1*b^3*c*d^13*f^4 + 96*a^7*b^7*c^13*d*f^4 + 96*a^7*b^7*c*d^13*f^4 + 96*a^3*b
^11*c^13*d*f^4 + 96*a*b^13*c^11*d^3*f^4 + 96*a*b^13*c^7*d^7*f^4 + 24*a^13*b
*c^9*d^5*f^4 + 24*a^9*b^5*c^13*d*f^4 + 24*a^5*b^9*c*d^13*f^4 + 24*a*b^13*c^
5*d^9*f^4 + 24*a^13*b*c*d^13*f^4 + 24*a*b^13*c^13*d*f^4 + 3648*a^7*b^7*c^7*
d^7*f^4 - 3188*a^8*b^6*c^6*d^8*f^4 - 3188*a^6*b^8*c^8*d^6*f^4 - 2912*a^8*b^
6*c^8*d^6*f^4 - 2912*a^6*b^8*c^6*d^8*f^4 + 2592*a^9*b^5*c^7*d^7*f^4 + 2592*
a^7*b^7*c^9*d^5*f^4 + 2592*a^7*b^7*c^5*d^9*f^4 + 2592*a^5*b^9*c^7*d^7*f^4 +
2168*a^9*b^5*c^5*d^9*f^4 + 2168*a^5*b^9*c^9*d^5*f^4 - 1776*a^10*b^4*c^6*d^
8*f^4 - 1776*a^8*b^6*c^4*d^10*f^4 - 1776*a^6*b^8*c^10*d^4*f^4 - 1776*a^4*b^
10*c^8*d^6*f^4 + 1568*a^9*b^5*c^9*d^5*f^4 + 1568*a^5*b^9*c^5*d^9*f^4 - 1344

*a^10*b^4*c^8*d^6*f^4 - 1344*a^8*b^6*c^10*d^4*f^4 - 1344*a^6*b^8*c^4*d^10*f
^4 - 1344*a^4*b^10*c^6*d^8*f^4 - 1164*a^10*b^4*c^4*d^10*f^4 - 1164*a^4*b^10
*c^10*d^4*f^4 + 896*a^11*b^3*c^5*d^9*f^4 + 896*a^9*b^5*c^3*d^11*f^4 + 896*a
^5*b^9*c^11*d^3*f^4 + 896*a^3*b^11*c^9*d^5*f^4 + 864*a^11*b^3*c^7*d^7*f^4 +
 864*a^7*b^7*c^11*d^3*f^4 + 864*a^7*b^7*c^3*d^11*f^4 + 864*a^3*b^11*c^7*d^7
*f^4 - 480*a^10*b^4*c^10*d^4*f^4 - 480*a^4*b^10*c^4*d^10*f^4 + 464*a^11*b^3
*c^3*d^11*f^4 + 464*a^3*b^11*c^11*d^3*f^4 - 424*a^12*b^2*c^6*d^8*f^4 - 424*
a^8*b^6*c^2*d^12*f^4 - 424*a^6*b^8*c^12*d^2*f^4 - 424*a^2*b^12*c^8*d^6*f^4
+ 416*a^11*b^3*c^9*d^5*f^4 + 416*a^9*b^5*c^11*d^3*f^4 + 416*a^5*b^9*c^3*d^1
1*f^4 + 416*a^3*b^11*c^5*d^9*f^4 - 336*a^12*b^2*c^4*d^10*f^4 - 336*a^10*b^4
*c^2*d^12*f^4 - 336*a^4*b^10*c^12*d^2*f^4 - 336*a^2*b^12*c^10*d^4*f^4 - 256
*a^12*b^2*c^8*d^6*f^4 - 256*a^8*b^6*c^12*d^2*f^4 - 256*a^6*b^8*c^2*d^12*f^4
 - 256*a^2*b^12*c^6*d^8*f^4 - 124*a^12*b^2*c^2*d^12*f^4 - 124*a^2*b^12*c^12
*d^2*f^4 + 80*a^11*b^3*c^11*d^3*f^4 + 80*a^3*b^11*c^3*d^11*f^4 - 60*a^12*b^
2*c^10*d^4*f^4 - 60*a^10*b^4*c^12*d^2*f^4 - 60*a^4*b^10*c^2*d^12*f^4 - 60*a
^2*b^12*c^4*d^10*f^4 - 24*b^14*c^10*d^4*f^4 - 16*b^14*c^12*d^2*f^4 - 16*b^1
4*c^8*d^6*f^4 - 4*b^14*c^6*d^8*f^4 - 24*a^14*c^4*d^10*f^4 - 16*a^14*c^6*d^8
*f^4 - 16*a^14*c^2*d^12*f^4 - 4*a^14*c^8*d^6*f^4 - 24*a^10*b^4*d^14*f^4 - 1
6*a^12*b^2*d^14*f^4 - 16*a^8*b^6*d^14*f^4 - 4*a^6*b^8*d^14*f^4 - 24*a^4*b^1
0*c^14*f^4 - 16*a^6*b^8*c^14*f^4 - 16*a^2*b^12*c^14*f^4 - 4*a^8*b^6*c^14*f^
4 - 4*b^14*c^14*f^4 - 4*a^14*d^14*f^4 + 36*A*C*a^9*b*c*d^9*f^2 + 36*A*C*a*b
^9*c^9*d*f^2 + 32*A*C*a*b^9*c*d^9*f^2 - 552*B*C*a^7*b^3*c^4*d^6*f^2 - 552*B
*C*a^4*b^6*c^7*d^3*f^2 - 408*B*C*a^5*b^5*c^4*d^6*f^2 - 408*B*C*a^4*b^6*c^5*
d^5*f^2 + 360*B*C*a^6*b^4*c^3*d^7*f^2 + 360*B*C*a^3*b^7*c^6*d^4*f^2 - 248*B
*C*a^7*b^3*c^2*d^8*f^2 - 248*B*C*a^2*b^8*c^7*d^3*f^2 + 184*B*C*a^6*b^4*c^5*
d^5*f^2 + 184*B*C*a^5*b^5*c^6*d^4*f^2 + 152*B*C*a^8*b^2*c^3*d^7*f^2 - 152*B
*C*a^5*b^5*c^2*d^8*f^2 + 152*B*C*a^3*b^7*c^8*d^2*f^2 - 152*B*C*a^2*b^8*c^5*
d^5*f^2 - 104*B*C*a^7*b^3*c^6*d^4*f^2 - 104*B*C*a^6*b^4*c^7*d^3*f^2 + 64*B*
C*a^8*b^2*c^5*d^5*f^2 + 64*B*C*a^5*b^5*c^8*d^2*f^2 - 56*B*C*a^4*b^6*c^3*d^7
*f^2 - 56*B*C*a^3*b^7*c^4*d^6*f^2 - 24*B*C*a^8*b^2*c^7*d^3*f^2 - 24*B*C*a^7
*b^3*c^8*d^2*f^2 - 24*B*C*a^3*b^7*c^2*d^8*f^2 - 24*B*C*a^2*b^8*c^3*d^7*f^2
- 696*A*C*a^5*b^5*c^5*d^5*f^2 + 536*A*C*a^6*b^4*c^6*d^4*f^2 + 536*A*C*a^6*b
^4*c^4*d^6*f^2 + 536*A*C*a^4*b^6*c^6*d^4*f^2 + 472*A*C*a^4*b^6*c^4*d^6*f^2
- 232*A*C*a^7*b^3*c^5*d^5*f^2 - 232*A*C*a^5*b^5*c^7*d^3*f^2 + 216*A*C*a^3*b
^7*c^3*d^7*f^2 + 168*A*C*a^7*b^3*c^3*d^7*f^2 + 168*A*C*a^3*b^7*c^7*d^3*f^2
- 154*A*C*a^8*b^2*c^2*d^8*f^2 - 154*A*C*a^2*b^8*c^8*d^2*f^2 + 62*A*C*a^8*b^
2*c^6*d^4*f^2 + 62*A*C*a^6*b^4*c^8*d^2*f^2 - 40*A*C*a^7*b^3*c^7*d^3*f^2 - 4
0*A*C*a^5*b^5*c^3*d^7*f^2 - 40*A*C*a^3*b^7*c^5*d^5*f^2 + 32*A*C*a^6*b^4*c^2
*d^8*f^2 + 32*A*C*a^2*b^8*c^6*d^4*f^2 - 32*A*C*a^2*b^8*c^2*d^8*f^2 + 30*A*C
*a^4*b^6*c^2*d^8*f^2 + 30*A*C*a^2*b^8*c^4*d^6*f^2 + 16*A*C*a^8*b^2*c^4*d^6*
f^2 + 16*A*C*a^4*b^6*c^8*d^2*f^2 - 488*A*B*a^6*b^4*c^3*d^7*f^2 - 488*A*B*a^
3*b^7*c^6*d^4*f^2 + 440*A*B*a^7*b^3*c^4*d^6*f^2 + 440*A*B*a^4*b^6*c^7*d^3*f
^2 - 360*A*B*a^6*b^4*c^5*d^5*f^2 - 360*A*B*a^5*b^5*c^6*d^4*f^2 - 192*A*B*a^
8*b^2*c^3*d^7*f^2 - 192*A*B*a^3*b^7*c^8*d^2*f^2 - 168*A*B*a^3*b^7*c^2*d^8*f
^2 - 168*A*B*a^2*b^8*c^3*d^7*f^2 - 152*A*B*a^4*b^6*c^3*d^7*f^2 - 152*A*B*a^

3*b^7*c^4*d^6*f^2 - 120*A*B*a^8*b^2*c^5*d^5*f^2 + 120*A*B*a^7*b^3*c^2*d^8*f
^2 - 120*A*B*a^5*b^5*c^8*d^2*f^2 + 120*A*B*a^5*b^5*c^4*d^6*f^2 - 120*A*B*a^
5*b^5*c^2*d^8*f^2 + 120*A*B*a^4*b^6*c^5*d^5*f^2 + 120*A*B*a^2*b^8*c^7*d^3*f
^2 - 120*A*B*a^2*b^8*c^5*d^5*f^2 + 40*A*B*a^7*b^3*c^6*d^4*f^2 + 40*A*B*a^6*
b^4*c^7*d^3*f^2 - 72*B*C*a^9*b*c^4*d^6*f^2 - 72*B*C*a^4*b^6*c^9*d*f^2 - 64*
B*C*a^4*b^6*c*d^9*f^2 - 64*B*C*a*b^9*c^4*d^6*f^2 - 32*B*C*a^8*b^2*c*d^9*f^2
 - 32*B*C*a*b^9*c^8*d^2*f^2 - 16*B*C*a^2*b^8*c*d^9*f^2 - 16*B*C*a*b^9*c^2*d
^8*f^2 + 8*B*C*a^9*b*c^6*d^4*f^2 - 8*B*C*a^9*b*c^2*d^8*f^2 + 8*B*C*a^6*b^4*
c^9*d*f^2 - 8*B*C*a^2*b^8*c^9*d*f^2 + 104*A*C*a^7*b^3*c*d^9*f^2 + 104*A*C*a
*b^9*c^7*d^3*f^2 + 96*A*C*a^3*b^7*c*d^9*f^2 + 96*A*C*a*b^9*c^3*d^7*f^2 + 72
*A*C*a^9*b*c^3*d^7*f^2 + 72*A*C*a^3*b^7*c^9*d*f^2 + 68*A*C*a^5*b^5*c*d^9*f^
2 + 68*A*C*a*b^9*c^5*d^5*f^2 - 28*A*C*a^9*b*c^5*d^5*f^2 - 28*A*C*a^5*b^5*c^
9*d*f^2 + 80*A*B*a^9*b*c^4*d^6*f^2 + 80*A*B*a^4*b^6*c^9*d*f^2 + 24*A*B*a^8*
b^2*c*d^9*f^2 - 24*A*B*a^6*b^4*c*d^9*f^2 + 24*A*B*a^4*b^6*c*d^9*f^2 - 24*A*
B*a^2*b^8*c*d^9*f^2 + 24*A*B*a*b^9*c^8*d^2*f^2 - 24*A*B*a*b^9*c^6*d^4*f^2 +
 24*A*B*a*b^9*c^4*d^6*f^2 - 24*A*B*a*b^9*c^2*d^8*f^2 - 32*B*C*b^10*c^7*d^3*
f^2 - 8*B*C*b^10*c^5*d^5*f^2 + 34*A*C*b^10*c^6*d^4*f^2 + 16*B*C*a^10*c^3*d^
7*f^2 + 16*A*C*b^10*c^4*d^6*f^2 - 12*A*C*b^10*c^8*d^2*f^2 - 96*A*B*b^10*c^5
*d^5*f^2 - 72*A*B*b^10*c^3*d^7*f^2 - 32*B*C*a^7*b^3*d^10*f^2 - 28*A*C*a^10*
c^2*d^8*f^2 - 24*A*B*b^10*c^7*d^3*f^2 - 8*B*C*a^5*b^5*d^10*f^2 + 2*A*C*a^10
*c^4*d^6*f^2 + 34*A*C*a^6*b^4*d^10*f^2 + 16*B*C*a^3*b^7*c^10*f^2 + 16*A*C*a
^4*b^6*d^10*f^2 - 16*A*B*a^10*c^3*d^7*f^2 - 12*A*C*a^8*b^2*d^10*f^2 - 96*A*
B*a^5*b^5*d^10*f^2 - 72*A*B*a^3*b^7*d^10*f^2 - 28*A*C*a^2*b^8*c^10*f^2 - 24
*A*B*a^7*b^3*d^10*f^2 + 2*A*C*a^4*b^6*c^10*f^2 - 16*A*B*a^3*b^7*c^10*f^2 +
444*C^2*a^5*b^5*c^5*d^5*f^2 + 148*C^2*a^7*b^3*c^5*d^5*f^2 + 148*C^2*a^5*b^5
*c^7*d^3*f^2 + 148*C^2*a^5*b^5*c^3*d^7*f^2 + 148*C^2*a^3*b^7*c^5*d^5*f^2 -
140*C^2*a^6*b^4*c^6*d^4*f^2 - 140*C^2*a^6*b^4*c^4*d^6*f^2 - 140*C^2*a^4*b^6
*c^6*d^4*f^2 - 140*C^2*a^4*b^6*c^4*d^6*f^2 + 109*C^2*a^8*b^2*c^2*d^8*f^2 +
109*C^2*a^2*b^8*c^8*d^2*f^2 + 48*C^2*a^8*b^2*c^4*d^6*f^2 + 48*C^2*a^6*b^4*c
^2*d^8*f^2 + 48*C^2*a^4*b^6*c^8*d^2*f^2 + 48*C^2*a^2*b^8*c^6*d^4*f^2 + 20*C
^2*a^7*b^3*c^7*d^3*f^2 - 20*C^2*a^7*b^3*c^3*d^7*f^2 - 20*C^2*a^3*b^7*c^7*d^
3*f^2 + 20*C^2*a^3*b^7*c^3*d^7*f^2 + 17*C^2*a^8*b^2*c^6*d^4*f^2 + 17*C^2*a^
6*b^4*c^8*d^2*f^2 + 17*C^2*a^4*b^6*c^2*d^8*f^2 + 17*C^2*a^2*b^8*c^4*d^6*f^2
 + 16*C^2*a^8*b^2*c^8*d^2*f^2 + 16*C^2*a^2*b^8*c^2*d^8*f^2 - 396*B^2*a^5*b^
5*c^5*d^5*f^2 + 308*B^2*a^6*b^4*c^4*d^6*f^2 + 308*B^2*a^4*b^6*c^6*d^4*f^2 +
 300*B^2*a^4*b^6*c^4*d^6*f^2 + 284*B^2*a^6*b^4*c^6*d^4*f^2 - 132*B^2*a^7*b^
3*c^5*d^5*f^2 - 132*B^2*a^5*b^5*c^7*d^3*f^2 - 84*B^2*a^5*b^5*c^3*d^7*f^2 -
84*B^2*a^3*b^7*c^5*d^5*f^2 + 61*B^2*a^4*b^6*c^2*d^8*f^2 + 61*B^2*a^2*b^8*c^
4*d^6*f^2 - 59*B^2*a^8*b^2*c^2*d^8*f^2 - 59*B^2*a^2*b^8*c^8*d^2*f^2 + 56*B^
2*a^6*b^4*c^2*d^8*f^2 + 56*B^2*a^2*b^8*c^6*d^4*f^2 + 52*B^2*a^7*b^3*c^3*d^7
*f^2 + 52*B^2*a^3*b^7*c^7*d^3*f^2 + 44*B^2*a^3*b^7*c^3*d^7*f^2 + 33*B^2*a^8
*b^2*c^6*d^4*f^2 + 33*B^2*a^6*b^4*c^8*d^2*f^2 + 20*B^2*a^8*b^2*c^4*d^6*f^2
- 20*B^2*a^7*b^3*c^7*d^3*f^2 + 20*B^2*a^4*b^6*c^8*d^2*f^2 + 8*B^2*a^2*b^8*c
^2*d^8*f^2 + 337*A^2*a^4*b^6*c^2*d^8*f^2 + 337*A^2*a^2*b^8*c^4*d^6*f^2 + 27
2*A^2*a^2*b^8*c^2*d^8*f^2 + 252*A^2*a^5*b^5*c^5*d^5*f^2 + 244*A^2*a^4*b^6*c

^4*d^6*f^2 - 236*A^2*a^3*b^7*c^3*d^7*f^2 + 176*A^2*a^6*b^4*c^2*d^8*f^2 + 17
6*A^2*a^2*b^8*c^6*d^4*f^2 - 148*A^2*a^7*b^3*c^3*d^7*f^2 - 148*A^2*a^3*b^7*c
^7*d^3*f^2 - 140*A^2*a^6*b^4*c^6*d^4*f^2 + 109*A^2*a^8*b^2*c^2*d^8*f^2 + 10
9*A^2*a^2*b^8*c^8*d^2*f^2 - 108*A^2*a^5*b^5*c^3*d^7*f^2 - 108*A^2*a^3*b^7*c
^5*d^5*f^2 + 84*A^2*a^7*b^3*c^5*d^5*f^2 + 84*A^2*a^5*b^5*c^7*d^3*f^2 + 32*A
^2*a^8*b^2*c^4*d^6*f^2 + 32*A^2*a^4*b^6*c^8*d^2*f^2 + 20*A^2*a^7*b^3*c^7*d^
3*f^2 - 15*A^2*a^8*b^2*c^6*d^4*f^2 - 15*A^2*a^6*b^4*c^8*d^2*f^2 - 12*A^2*a^
6*b^4*c^4*d^6*f^2 - 12*A^2*a^4*b^6*c^6*d^4*f^2 + 8*B*C*b^10*c^9*d*f^2 - 16*
B*C*a^10*c*d^9*f^2 - 16*A*B*b^10*c^9*d*f^2 - 16*A*B*b^10*c*d^9*f^2 + 8*B*C*
a^9*b*d^10*f^2 - 16*B*C*a*b^9*c^10*f^2 + 16*A*B*a^10*c*d^9*f^2 - 16*A*B*a^9
*b*d^10*f^2 - 16*A*B*a*b^9*d^10*f^2 + 16*A*B*a*b^9*c^10*f^2 + 22*C^2*a^9*b*
c^5*d^5*f^2 + 22*C^2*a^5*b^5*c^9*d*f^2 + 22*C^2*a^5*b^5*c*d^9*f^2 + 22*C^2*
a*b^9*c^5*d^5*f^2 - 20*C^2*a^9*b*c^3*d^7*f^2 - 20*C^2*a^7*b^3*c*d^9*f^2 - 2
0*C^2*a^3*b^7*c^9*d*f^2 - 20*C^2*a*b^9*c^7*d^3*f^2 + 36*B^2*a^7*b^3*c*d^9*f
^2 + 36*B^2*a*b^9*c^7*d^3*f^2 + 28*B^2*a^9*b*c^3*d^7*f^2 + 28*B^2*a^3*b^7*c
^9*d*f^2 + 24*B^2*a^3*b^7*c*d^9*f^2 + 24*B^2*a*b^9*c^3*d^7*f^2 - 18*B^2*a^9
*b*c^5*d^5*f^2 - 18*B^2*a^5*b^5*c^9*d*f^2 + 6*B^2*a^5*b^5*c*d^9*f^2 + 6*B^2
*a*b^9*c^5*d^5*f^2 - 96*A^2*a^3*b^7*c*d^9*f^2 - 96*A^2*a*b^9*c^3*d^7*f^2 -
90*A^2*a^5*b^5*c*d^9*f^2 - 90*A^2*a*b^9*c^5*d^5*f^2 - 84*A^2*a^7*b^3*c*d^9*
f^2 - 84*A^2*a*b^9*c^7*d^3*f^2 - 52*A^2*a^9*b*c^3*d^7*f^2 - 52*A^2*a^3*b^7*
c^9*d*f^2 + 6*A^2*a^9*b*c^5*d^5*f^2 + 6*A^2*a^5*b^5*c^9*d*f^2 - 10*C^2*a^9*
b*c*d^9*f^2 - 10*C^2*a*b^9*c^9*d*f^2 + 14*B^2*a^9*b*c*d^9*f^2 + 14*B^2*a*b^
9*c^9*d*f^2 + 8*B^2*a*b^9*c*d^9*f^2 - 32*A^2*a*b^9*c*d^9*f^2 - 26*A^2*a^9*b
*c*d^9*f^2 - 26*A^2*a*b^9*c^9*d*f^2 + 2*A*C*b^10*c^10*f^2 + 2*A*C*a^10*d^10
*f^2 + 14*C^2*b^10*c^8*d^2*f^2 - C^2*b^10*c^6*d^4*f^2 + 31*B^2*b^10*c^6*d^4
*f^2 + 20*B^2*b^10*c^4*d^6*f^2 + 14*C^2*a^10*c^2*d^8*f^2 + 4*B^2*b^10*c^2*d
^8*f^2 + 2*B^2*b^10*c^8*d^2*f^2 - C^2*a^10*c^4*d^6*f^2 + 80*A^2*b^10*c^4*d^
6*f^2 + 64*A^2*b^10*c^2*d^8*f^2 + 31*A^2*b^10*c^6*d^4*f^2 + 14*C^2*a^8*b^2*
d^10*f^2 + 14*A^2*b^10*c^8*d^2*f^2 - 10*B^2*a^10*c^2*d^8*f^2 + 3*B^2*a^10*c
^4*d^6*f^2 - C^2*a^6*b^4*d^10*f^2 + 31*B^2*a^6*b^4*d^10*f^2 + 20*B^2*a^4*b^
6*d^10*f^2 + 14*C^2*a^2*b^8*c^10*f^2 + 14*A^2*a^10*c^2*d^8*f^2 + 4*B^2*a^2*
b^8*d^10*f^2 + 2*B^2*a^8*b^2*d^10*f^2 - C^2*a^4*b^6*c^10*f^2 - A^2*a^10*c^4
*d^6*f^2 + 80*A^2*a^4*b^6*d^10*f^2 + 64*A^2*a^2*b^8*d^10*f^2 + 31*A^2*a^6*b
^4*d^10*f^2 + 14*A^2*a^8*b^2*d^10*f^2 - 10*B^2*a^2*b^8*c^10*f^2 + 3*B^2*a^4
*b^6*c^10*f^2 + 14*A^2*a^2*b^8*c^10*f^2 - A^2*a^4*b^6*c^10*f^2 - C^2*b^10*c
^10*f^2 - C^2*a^10*d^10*f^2 + 16*A^2*b^10*d^10*f^2 + 3*B^2*b^10*c^10*f^2 +
3*B^2*a^10*d^10*f^2 - A^2*b^10*c^10*f^2 - A^2*a^10*d^10*f^2 - 96*A*B*C*a*b^
7*c*d^7*f - 28*A*B*C*a^7*b*c*d^7*f - 28*A*B*C*a*b^7*c^7*d*f + 484*A*B*C*a^4
*b^4*c^4*d^4*f - 424*A*B*C*a^3*b^5*c^3*d^5*f + 320*A*B*C*a^2*b^6*c^2*d^6*f
- 176*A*B*C*a^6*b^2*c^2*d^6*f - 176*A*B*C*a^2*b^6*c^6*d^2*f + 158*A*B*C*a^4
*b^4*c^2*d^6*f + 158*A*B*C*a^2*b^6*c^4*d^4*f - 136*A*B*C*a^5*b^3*c^5*d^3*f
- 34*A*B*C*a^6*b^2*c^4*d^4*f - 34*A*B*C*a^4*b^4*c^6*d^2*f + 28*A*B*C*a^5*b^
3*c^3*d^5*f + 28*A*B*C*a^3*b^5*c^5*d^3*f + 308*A*B*C*a^5*b^3*c*d^7*f + 308*
A*B*C*a*b^7*c^5*d^3*f + 20*A*B*C*a^7*b*c^3*d^5*f + 20*A*B*C*a^3*b^5*c^7*d*f
 + 30*B*C^2*a^7*b*c*d^7*f + 30*B*C^2*a*b^7*c^7*d*f + 160*A^2*B*a*b^7*c*d^7*

f - 2*A^2*B*a^7*b*c*d^7*f - 2*A^2*B*a*b^7*c^7*d*f - 96*A*B*C*b^8*c^4*d^4*f
+ 34*A*B*C*b^8*c^6*d^2*f - 32*A*B*C*b^8*c^2*d^6*f + 2*A*B*C*a^8*c^2*d^6*f -
96*A*B*C*a^4*b^4*d^8*f + 34*A*B*C*a^6*b^2*d^8*f - 32*A*B*C*a^2*b^6*d^8*f +
2*A*B*C*a^2*b^6*c^8*f - 210*B*C^2*a^4*b^4*c^4*d^4*f - 182*B^2*C*a^5*b^3*c^
2*d^6*f - 182*B^2*C*a^2*b^6*c^5*d^3*f + 180*B*C^2*a^5*b^3*c^5*d^3*f + 180*B
*C^2*a^3*b^5*c^3*d^5*f - 166*B^2*C*a^5*b^3*c^4*d^4*f - 166*B^2*C*a^4*b^4*c^
5*d^3*f + 152*B*C^2*a^6*b^2*c^2*d^6*f + 152*B*C^2*a^2*b^6*c^6*d^2*f - 112*B
^2*C*a^3*b^5*c^2*d^6*f - 112*B^2*C*a^2*b^6*c^3*d^5*f + 94*B^2*C*a^4*b^4*c^3
*d^5*f + 94*B^2*C*a^3*b^5*c^4*d^4*f - 80*B*C^2*a^2*b^6*c^2*d^6*f + 66*B*C^2
*a^5*b^3*c^3*d^5*f + 66*B*C^2*a^3*b^5*c^5*d^3*f + 46*B^2*C*a^6*b^2*c^3*d^5*
f + 46*B^2*C*a^3*b^5*c^6*d^2*f + 33*B*C^2*a^6*b^2*c^4*d^4*f + 33*B*C^2*a^4*
b^4*c^6*d^2*f + 24*B^2*C*a^6*b^2*c^5*d^3*f + 24*B^2*C*a^5*b^3*c^6*d^2*f - 1
6*B*C^2*a^6*b^2*c^6*d^2*f - 15*B*C^2*a^4*b^4*c^2*d^6*f - 15*B*C^2*a^2*b^6*c
^4*d^4*f - 190*A^2*C*a^4*b^4*c^3*d^5*f - 190*A^2*C*a^3*b^5*c^4*d^4*f + 182*
A^2*C*a^5*b^3*c^2*d^6*f + 182*A^2*C*a^2*b^6*c^5*d^3*f + 160*A^2*C*a^3*b^5*c
^2*d^6*f + 160*A^2*C*a^2*b^6*c^3*d^5*f - 150*A*C^2*a^5*b^3*c^2*d^6*f - 150*
A*C^2*a^2*b^6*c^5*d^3*f - 126*A*C^2*a^5*b^3*c^4*d^4*f - 126*A*C^2*a^4*b^4*c
^5*d^3*f + 126*A*C^2*a^4*b^4*c^3*d^5*f + 126*A*C^2*a^3*b^5*c^4*d^4*f - 96*A
*C^2*a^3*b^5*c^2*d^6*f - 96*A*C^2*a^2*b^6*c^3*d^5*f + 94*A^2*C*a^5*b^3*c^4*
d^4*f + 94*A^2*C*a^4*b^4*c^5*d^3*f + 54*A*C^2*a^6*b^2*c^3*d^5*f + 54*A*C^2*
a^3*b^5*c^6*d^2*f + 32*A*C^2*a^6*b^2*c^5*d^3*f + 32*A*C^2*a^5*b^3*c^6*d^2*f
- 22*A^2*C*a^6*b^2*c^3*d^5*f - 22*A^2*C*a^3*b^5*c^6*d^2*f + 500*A^2*B*a^3*
b^5*c^3*d^5*f - 290*A^2*B*a^4*b^4*c^4*d^4*f - 256*A^2*B*a^2*b^6*c^2*d^6*f -
230*A*B^2*a^4*b^4*c^3*d^5*f - 230*A*B^2*a^3*b^5*c^4*d^4*f + 142*A*B^2*a^5*
b^3*c^2*d^6*f + 142*A*B^2*a^2*b^6*c^5*d^3*f - 127*A^2*B*a^4*b^4*c^2*d^6*f -
127*A^2*B*a^2*b^6*c^4*d^4*f + 86*A*B^2*a^5*b^3*c^4*d^4*f + 86*A*B^2*a^4*b^
4*c^5*d^3*f + 80*A*B^2*a^3*b^5*c^2*d^6*f + 80*A*B^2*a^2*b^6*c^3*d^5*f + 40*
A^2*B*a^6*b^2*c^2*d^6*f + 40*A^2*B*a^2*b^6*c^6*d^2*f + 34*A^2*B*a^5*b^3*c^3
*d^5*f + 34*A^2*B*a^3*b^5*c^5*d^3*f - 30*A*B^2*a^6*b^2*c^3*d^5*f - 30*A*B^2
*a^3*b^5*c^6*d^2*f + 20*A^2*B*a^5*b^3*c^5*d^3*f - 15*A^2*B*a^6*b^2*c^4*d^4*
f - 15*A^2*B*a^4*b^4*c^6*d^2*f - 98*B^2*C*a^6*b^2*c*d^7*f - 98*B^2*C*a*b^7*
c^6*d^2*f - 90*B*C^2*a^5*b^3*c*d^7*f - 90*B*C^2*a*b^7*c^5*d^3*f + 48*B^2*C*
a^4*b^4*c*d^7*f + 48*B^2*C*a*b^7*c^4*d^4*f + 40*B^2*C*a^2*b^6*c*d^7*f + 40*
B^2*C*a*b^7*c^2*d^6*f - 32*B*C^2*a^3*b^5*c*d^7*f - 32*B*C^2*a*b^7*c^3*d^5*f
+ 26*B^2*C*a^7*b*c^2*d^6*f + 26*B^2*C*a^2*b^6*c^7*d*f - 26*B*C^2*a^7*b*c^3
*d^5*f - 26*B*C^2*a^3*b^5*c^7*d*f - 8*B^2*C*a^7*b*c^4*d^4*f - 8*B^2*C*a^4*b
^4*c^7*d*f - 224*A^2*C*a^4*b^4*c*d^7*f - 224*A^2*C*a*b^7*c^4*d^4*f - 96*A^2
*C*a^2*b^6*c*d^7*f - 96*A^2*C*a*b^7*c^2*d^6*f + 96*A*C^2*a^4*b^4*c*d^7*f +
96*A*C^2*a*b^7*c^4*d^4*f - 66*A*C^2*a^6*b^2*c*d^7*f - 66*A*C^2*a*b^7*c^6*d^
2*f + 64*A*C^2*a^2*b^6*c*d^7*f + 64*A*C^2*a*b^7*c^2*d^6*f + 34*A^2*C*a^6*b^
2*c*d^7*f + 34*A^2*C*a*b^7*c^6*d^2*f + 34*A*C^2*a^7*b*c^2*d^6*f + 34*A*C^2*
a^2*b^6*c^7*d*f - 2*A^2*C*a^7*b*c^2*d^6*f - 2*A^2*C*a^2*b^6*c^7*d*f - 208*A
*B^2*a^4*b^4*c*d^7*f - 208*A*B^2*a*b^7*c^4*d^4*f + 160*A^2*B*a^3*b^5*c*d^7*
f + 160*A^2*B*a*b^7*c^3*d^5*f - 154*A^2*B*a^5*b^3*c*d^7*f - 154*A^2*B*a*b^7
*c^5*d^3*f - 112*A*B^2*a^2*b^6*c*d^7*f - 112*A*B^2*a*b^7*c^2*d^6*f + 58*A*B

^2*a^6*b^2*c*d^7*f + 58*A*B^2*a*b^7*c^6*d^2*f - 10*A*B^2*a^7*b*c^2*d^6*f -
10*A*B^2*a^2*b^6*c^7*d*f + 6*A^2*B*a^7*b*c^3*d^5*f + 6*A^2*B*a^3*b^5*c^7*d*
f + 32*B^2*C*b^8*c^5*d^3*f - 17*B*C^2*b^8*c^6*d^2*f + 8*B^2*C*b^8*c^3*d^5*f
 + 64*A^2*C*b^8*c^3*d^5*f - 32*A^2*C*b^8*c^5*d^3*f + 32*A*C^2*b^8*c^5*d^3*f
 - B*C^2*a^8*c^2*d^6*f + 112*A^2*B*b^8*c^4*d^4*f - 64*A*B^2*b^8*c^5*d^3*f +
 32*B^2*C*a^5*b^3*d^8*f - 17*B*C^2*a^6*b^2*d^8*f + 16*A^2*B*b^8*c^2*d^6*f +
 16*A*B^2*b^8*c^3*d^5*f + 8*B^2*C*a^3*b^5*d^8*f - A^2*B*b^8*c^6*d^2*f + 64*
A^2*C*a^3*b^5*d^8*f - 32*A^2*C*a^5*b^3*d^8*f + 32*A*C^2*a^5*b^3*d^8*f - A^2
*B*a^8*c^2*d^6*f - B*C^2*a^2*b^6*c^8*f + 112*A^2*B*a^4*b^4*d^8*f - 64*A*B^2
*a^5*b^3*d^8*f + 16*A^2*B*a^2*b^6*d^8*f + 16*A*B^2*a^3*b^5*d^8*f - A^2*B*a^
6*b^2*d^8*f - A^2*B*a^2*b^6*c^8*f - 8*B^3*a*b^7*c*d^7*f - 2*B^3*a^7*b*c*d^7
*f - 2*B^3*a*b^7*c^7*d*f - 6*B^2*C*b^8*c^7*d*f + 32*A^2*C*b^8*c*d^7*f + 6*A
^2*C*b^8*c^7*d*f - 6*A*C^2*b^8*c^7*d*f - 2*B^2*C*a^8*c*d^7*f + 16*A*B^2*b^8
*c*d^7*f - 6*B^2*C*a^7*b*d^8*f - 6*A^2*C*a^8*c*d^7*f + 6*A*C^2*a^8*c*d^7*f
 - 2*A*B^2*b^8*c^7*d*f + 32*A^2*C*a*b^7*d^8*f + 6*A^2*C*a^7*b*d^8*f - 6*A*C^
2*a^7*b*d^8*f - 2*B^2*C*a*b^7*c^8*f + 2*A*B^2*a^8*c*d^7*f + 16*A*B^2*a*b^7*
d^8*f - 6*A^2*C*a*b^7*c^8*f + 6*A*C^2*a*b^7*c^8*f - 2*A*B^2*a^7*b*d^8*f + 2
*A*B^2*a*b^7*c^8*f - 50*C^3*a^6*b^2*c^3*d^5*f + 50*C^3*a^5*b^3*c^2*d^6*f -
50*C^3*a^3*b^5*c^6*d^2*f + 50*C^3*a^2*b^6*c^5*d^3*f + 42*C^3*a^5*b^3*c^4*d^
4*f + 42*C^3*a^4*b^4*c^5*d^3*f - 42*C^3*a^4*b^4*c^3*d^5*f - 42*C^3*a^3*b^5*
c^4*d^4*f - 32*C^3*a^6*b^2*c^5*d^3*f - 32*C^3*a^5*b^3*c^6*d^2*f + 32*C^3*a^
3*b^5*c^2*d^6*f + 32*C^3*a^2*b^6*c^3*d^5*f + 94*B^3*a^4*b^4*c^4*d^4*f + 48*
B^3*a^2*b^6*c^2*d^6*f - 44*B^3*a^3*b^5*c^3*d^5*f - 32*B^3*a^6*b^2*c^2*d^6*f
 - 32*B^3*a^2*b^6*c^6*d^2*f + 29*B^3*a^4*b^4*c^2*d^6*f + 29*B^3*a^2*b^6*c^4
*d^4*f - 20*B^3*a^5*b^3*c^5*d^3*f + 18*B^3*a^5*b^3*c^3*d^5*f + 18*B^3*a^3*b
^5*c^5*d^3*f - 3*B^3*a^6*b^2*c^4*d^4*f - 3*B^3*a^4*b^4*c^6*d^2*f + 106*A^3*
a^4*b^4*c^3*d^5*f + 106*A^3*a^3*b^5*c^4*d^4*f - 96*A^3*a^3*b^5*c^2*d^6*f -
96*A^3*a^2*b^6*c^3*d^5*f - 82*A^3*a^5*b^3*c^2*d^6*f - 82*A^3*a^2*b^6*c^5*d^
3*f + 18*A^3*a^6*b^2*c^3*d^5*f + 18*A^3*a^3*b^5*c^6*d^2*f - 10*A^3*a^5*b^3*
c^4*d^4*f - 10*A^3*a^4*b^4*c^5*d^3*f - 22*C^3*a^7*b*c^2*d^6*f + 22*C^3*a^6*
b^2*c*d^7*f - 22*C^3*a^2*b^6*c^7*d*f + 22*C^3*a*b^7*c^6*d^2*f - 2*A*B*C*b^8
*c^8*f - 2*A*B*C*a^8*d^8*f + 62*B^3*a^5*b^3*c*d^7*f + 62*B^3*a*b^7*c^5*d^3*
f + 16*B^3*a^3*b^5*c*d^7*f + 16*B^3*a*b^7*c^3*d^5*f + 6*B^3*a^7*b*c^3*d^5*f
 + 6*B^3*a^3*b^5*c^7*d*f + 128*A^3*a^4*b^4*c*d^7*f + 128*A^3*a*b^7*c^4*d^4*
f + 32*A^3*a^2*b^6*c*d^7*f + 32*A^3*a*b^7*c^2*d^6*f - 10*A^3*a^7*b*c^2*d^6*
f + 10*A^3*a^6*b^2*c*d^7*f - 10*A^3*a^2*b^6*c^7*d*f + 10*A^3*a*b^7*c^6*d^2*
f + 11*B^3*b^8*c^6*d^2*f - 8*B^3*b^8*c^4*d^4*f - 4*B^3*b^8*c^2*d^6*f - 64*A
^3*b^8*c^3*d^5*f - B^3*a^8*c^2*d^6*f + 11*B^3*a^6*b^2*d^8*f - 8*B^3*a^4*b^4
*d^8*f - 4*B^3*a^2*b^6*d^8*f - 64*A^3*a^3*b^5*d^8*f - B^3*a^2*b^6*c^8*f + 2
*C^3*b^8*c^7*d*f - 2*C^3*a^8*c*d^7*f - 32*A^3*b^8*c*d^7*f + 2*C^3*a^7*b*d^8
*f - 2*A^3*b^8*c^7*d*f - 2*C^3*a*b^7*c^8*f + 2*A^3*a^8*c*d^7*f - 32*A^3*a*b
^7*d^8*f - 2*A^3*a^7*b*d^8*f + 2*A^3*a*b^7*c^8*f - 16*A^2*B*b^8*d^8*f + B*C
^2*b^8*c^8*f + B*C^2*a^8*d^8*f + A^2*B*b^8*c^8*f + A^2*B*a^8*d^8*f + B^3*b^
8*c^8*f + B^3*a^8*d^8*f - 4*A*B^2*C*a^5*b*c*d^5 - 4*A*B^2*C*a*b^5*c^5*d + 4
*A*B^2*C*a*b^5*c*d^5 + 22*A^2*B*C*a^3*b^3*c^2*d^4 + 22*A^2*B*C*a^2*b^4*c^3*

```
d^3 - 20*A*B^2*C*a^3*b^3*c^3*d^3 + 14*A*B^2*C*a^4*b^2*c^2*d^4 + 14*A*B^2*C*
a^2*b^4*c^4*d^2 - 14*A*B*C^2*a^3*b^3*c^2*d^4 - 14*A*B*C^2*a^2*b^4*c^3*d^3 +
 12*A*B*C^2*a^4*b^2*c^3*d^3 + 12*A*B*C^2*a^3*b^3*c^4*d^2 - 6*A^2*B*C*a^4*b^
2*c^3*d^3 - 6*A^2*B*C*a^3*b^3*c^4*d^2 - 4*A*B^2*C*a^2*b^4*c^2*d^4 + 22*A*B*
C^2*a^4*b^2*c*d^5 + 22*A*B*C^2*a*b^5*c^4*d^2 - 20*A^2*B*C*a^4*b^2*c*d^5 - 2
0*A^2*B*C*a*b^5*c^4*d^2 + 10*A*B*C^2*a^2*b^4*c*d^5 + 10*A*B*C^2*a*b^5*c^2*d
^4 - 8*A^2*B*C*a^2*b^4*c*d^5 - 8*A^2*B*C*a*b^5*c^2*d^4 + 4*A*B^2*C*a^3*b^3*
c*d^5 + 4*A*B^2*C*a*b^5*c^3*d^3 - 4*A*B*C^2*a^5*b*c^2*d^4 - 4*A*B*C^2*a^2*b
^4*c^5*d + 2*A^2*B*C*a^5*b*c^2*d^4 + 2*A^2*B*C*a^2*b^4*c^5*d - 8*B^3*C*a^4*
b^2*c*d^5 - 8*B^3*C*a*b^5*c^4*d^2 - 8*B*C^3*a^4*b^2*c*d^5 - 8*B*C^3*a*b^5*c
^4*d^2 - 4*B^3*C*a^2*b^4*c*d^5 - 4*B^3*C*a*b^5*c^2*d^4 + 4*B^2*C^2*a^5*b*c*
d^5 + 4*B^2*C^2*a*b^5*c^5*d - 4*B*C^3*a^2*b^4*c*d^5 - 4*B*C^3*a*b^5*c^2*d^4
 + 2*B^3*C*a^5*b*c^2*d^4 + 2*B^3*C*a^2*b^4*c^5*d + 2*B^2*C^2*a*b^5*c*d^5 +
2*B*C^3*a^5*b*c^2*d^4 + 2*B*C^3*a^2*b^4*c^5*d + 24*A^3*C*a^3*b^3*c*d^5 + 24
*A^3*C*a*b^5*c^3*d^3 - 24*A^2*C^2*a*b^5*c*d^5 + 12*A^2*C^2*a^5*b*c*d^5 + 12
*A^2*C^2*a*b^5*c^5*d + 8*A*C^3*a^3*b^3*c*d^5 + 8*A*C^3*a*b^5*c^3*d^3 + 6*A^
3*B*a^4*b^2*c*d^5 + 6*A^3*B*a*b^5*c^4*d^2 - 6*A^2*B^2*a*b^5*c*d^5 + 6*A*B^3
*a^4*b^2*c*d^5 + 6*A*B^3*a*b^5*c^4*d^2 + 2*A^3*B*a^2*b^4*c*d^5 + 2*A^3*B*a*
b^5*c^2*d^4 + 2*A*B^3*a^2*b^4*c*d^5 + 2*A*B^3*a*b^5*c^2*d^4 + 20*A^2*B*C*b^
6*c^3*d^3 - 10*A*B*C^2*b^6*c^3*d^3 - 2*A*B^2*C*b^6*c^4*d^2 - 2*A*B^2*C*b^6*
c^2*d^4 + 20*A^2*B*C*a^3*b^3*d^6 - 10*A*B*C^2*a^3*b^3*d^6 - 2*A*B^2*C*a^4*b
^2*d^6 - 2*A*B^2*C*a^2*b^4*d^6 + 10*B^2*C^2*a^3*b^3*c^3*d^3 + 4*B^2*C^2*a^4
*b^2*c^4*d^2 - 3*B^2*C^2*a^4*b^2*c^2*d^4 - 3*B^2*C^2*a^2*b^4*c^4*d^2 + 2*B^
2*C^2*a^2*b^4*c^2*d^4 + 40*A^2*C^2*a^2*b^4*c^2*d^4 - 16*A^2*C^2*a^4*b^2*c^2
*d^4 - 16*A^2*C^2*a^2*b^4*c^4*d^2 + 4*A^2*C^2*a^4*b^2*c^4*d^2 + 18*A^2*B^2*
a^2*b^4*c^2*d^4 + 10*A^2*B^2*a^3*b^3*c^3*d^3 - 3*A^2*B^2*a^4*b^2*c^2*d^4 -
3*A^2*B^2*a^2*b^4*c^4*d^2 + 24*A^3*C*a*b^5*c*d^5 - 12*A*C^3*a^5*b*c*d^5 - 1
2*A*C^3*a*b^5*c^5*d + 8*A*C^3*a*b^5*c*d^5 - 4*A^3*C*a^5*b*c*d^5 - 4*A^3*C*a
*b^5*c^5*d + 8*A^2*B*C*b^6*c*d^5 + 4*A*B*C^2*b^6*c^5*d - 4*A*B*C^2*b^6*c*d^
5 - 2*A^2*B*C*b^6*c^5*d + 8*A^2*B*C*a*b^5*d^6 + 4*A*B*C^2*a^5*b*d^6 - 4*A*B
*C^2*a*b^5*d^6 - 2*A^2*B*C*a^5*b*d^6 - 6*B^3*C*a^4*b^2*c^3*d^3 - 6*B^3*C*a^
3*b^3*c^4*d^2 - 6*B*C^3*a^4*b^2*c^3*d^3 - 6*B*C^3*a^3*b^3*c^4*d^2 + 2*B^3*C
*a^3*b^3*c^2*d^4 + 2*B^3*C*a^2*b^4*c^3*d^3 + 2*B^2*C^2*a^3*b^3*c*d^5 + 2*B^
2*C^2*a*b^5*c^3*d^3 + 2*B*C^3*a^3*b^3*c^2*d^4 + 2*B*C^3*a^2*b^4*c^3*d^3 - 4
8*A^3*C*a^2*b^4*c^2*d^4 - 24*A^2*C^2*a^3*b^3*c*d^5 - 24*A^2*C^2*a*b^5*c^3*d
^3 - 16*A*C^3*a^2*b^4*c^2*d^4 + 8*A^3*C*a^4*b^2*c^2*d^4 + 8*A^3*C*a^2*b^4*c
^4*d^2 - 8*A*C^3*a^4*b^2*c^4*d^2 + 8*A*C^3*a^4*b^2*c^2*d^4 + 8*A*C^3*a^2*b^
4*c^4*d^2 - 10*A^3*B*a^3*b^3*c^2*d^4 - 10*A^3*B*a^2*b^4*c^3*d^3 - 10*A*B^3*
a^3*b^3*c^2*d^4 - 10*A*B^3*a^2*b^4*c^3*d^3 - 6*A^2*B^2*a^3*b^3*c*d^5 - 6*A^
2*B^2*a*b^5*c^3*d^3 + 3*B^2*C^2*b^6*c^4*d^2 - 8*A^2*C^2*b^6*c^4*d^2 + 8*A^2
*C^2*b^6*c^2*d^4 + 9*A^2*B^2*b^6*c^2*d^4 + 3*B^2*C^2*a^4*b^2*d^6 + 3*A^2*B^
2*b^6*c^4*d^2 - 8*A^2*C^2*a^4*b^2*d^6 + 8*A^2*C^2*a^2*b^4*d^6 + 9*A^2*B^2*a
^2*b^4*d^6 + 3*A^2*B^2*a^4*b^2*d^6 + 2*B^4*a^3*b^3*c*d^5 + 2*B^4*a*b^5*c^3*
d^3 - 8*A^4*a^3*b^3*c*d^5 - 8*A^4*a*b^5*c^3*d^3 - 16*A^3*C*b^6*c^2*d^4 + 4*
A^3*C*b^6*c^4*d^2 + 4*A*C^3*b^6*c^4*d^2 - 10*A^3*B*b^6*c^3*d^3 - 10*A*B^3*b
```

```
^6*c^3*d^3 - 16*A^3*C*a^2*b^4*d^6 + 4*A^3*C*a^4*b^2*d^6 + 4*A*C^3*a^4*b^2*d
^6 - 10*A^3*B*a^3*b^3*d^6 - 10*A*B^3*a^3*b^3*d^6 + 4*C^4*a^5*b*c*d^5 + 4*C^
4*a*b^5*c^5*d + 2*B^4*a*b^5*c*d^5 - 8*A^4*a*b^5*c*d^5 - 2*B^3*C*b^6*c^5*d -
 2*B*C^3*b^6*c^5*d - 4*A^3*B*b^6*c*d^5 - 4*A*B^3*b^6*c*d^5 - 2*B^3*C*a^5*b*
d^6 - 2*B*C^3*a^5*b*d^6 - 4*A^3*B*a*b^5*d^6 - 4*A*B^3*a*b^5*d^6 + 4*C^4*a^4
*b^2*c^4*d^2 + 4*C^4*a^2*b^4*c^2*d^4 + 10*B^4*a^3*b^3*c^3*d^3 - 3*B^4*a^4*b
^2*c^2*d^4 - 3*B^4*a^2*b^4*c^4*d^2 - 2*B^4*a^2*b^4*c^2*d^4 + 20*A^4*a^2*b^4
*c^2*d^4 + B^2*C^2*b^6*c^2*d^4 + B^2*C^2*a^2*b^4*d^6 - 8*A^3*C*b^6*d^6 + 3*
B^4*b^6*c^4*d^2 + 8*A^4*b^6*c^2*d^4 + 3*B^4*a^4*b^2*d^6 + 8*A^4*a^2*b^4*d^6
 + 4*A^2*C^2*b^6*d^6 + 4*A^2*B^2*b^6*d^6 + 4*A^4*b^6*d^6 + B^4*b^6*c^2*d^4
+ B^4*a^2*b^4*d^6, f, k), k, 1, 4) - ((A*a^3*d^3 + A*b^3*c^3 + A*a*b^2*d^3
- B*a*b^2*c^3 + C*a^2*b*c^3 + A*b^3*c*d^2 - B*a^3*c*d^2 + C*a^3*c^2*d - 2*B
*a*b^2*c*d^2 + C*a*b^2*c^2*d + C*a^2*b*c*d^2)/((a^2*d^2 + b^2*c^2 - 2*a*b*c
*d)*(a^2*c^2 + a^2*d^2 + b^2*c^2 + b^2*d^2)) + (tan(e + f*x)*(2*A*b^3*d^3 +
 A*a^2*b*d^3 - B*a*b^2*d^3 + A*b^3*c^2*d + C*a^2*b*d^3 - B*b^3*c*d^2 + C*b^
3*c^2*d - B*a*b^2*c^2*d - B*a^2*b*c*d^2 + 2*C*a^2*b*c^2*d))/((a^2*d^2 + b^2
*c^2 - 2*a*b*c*d)*(a^2*c^2 + a^2*d^2 + b^2*c^2 + b^2*d^2)))/(a*c + tan(e +
f*x)*(a*d + b*c) + b*d*tan(e + f*x)^2))/f
```

sympy [**F(-2)**] time = 0.00, size = 0, normalized size = 0.00

Exception raised: NotImplementedError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**2/(c+d*tan(f*x
+e))**2,x)
```

```
[Out] Exception raised: NotImplementedError
```

3.83 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx$

Optimal. Leaf size=841

$$-\frac{\left(b\left(3Cc^4-4Bdc^3+(5A+C)d^2c^2-2Bd^3c+3Ad^4\right)-ad^2\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)\log(c\cos(e+fx)+d\sin}{(bc-ad)^4\left(c^2+d^2\right)^2f}$$

```
[Out] -(a^3*(c^2*C-2*B*c*d-C*d^2-A*(c^2-d^2))-3*a*b^2*(c^2*C-2*B*c*d-C*d^2-A*(c^2
-d^2))+3*a^2*b*(2*c*(A-C)*d-B*(c^2-d^2))-b^3*(2*c*(A-C)*d-B*(c^2-d^2)))*x/(
a^2+b^2)^3/(c^2+d^2)^2-b*(6*a^5*b*B*d^2-3*a^6*C*d^2-a^4*b^2*d*(4*B*c+(10*A-
C)*d)-b^6*(c*(-2*B*d+C*c)-A*(c^2-3*d^2))+a*b^5*(2*c*(A-C)*d-B*(3*c^2-d^2))+
3*a^2*b^4*(c*(2*B*d+C*c)-A*(c^2+3*d^2))+a^3*b^3*(10*c*(A-C)*d+B*(c^2+3*d^2)
))*ln(a*cos(f*x+e)+b*sin(f*x+e))/(a^2+b^2)^3/(-a*d+b*c)^4/f-d^2*(b*(3*c^4*C
-4*B*c^3*d+c^2*(5*A+C)*d^2-2*B*c*d^3+3*A*d^4)-a*d^2*(2*c*(A-C)*d-B*(c^2-d^2
)))*ln(c*cos(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)^4/(c^2+d^2)^2/f-d*(3*a^3*b*B*d
*(c^2+d^2)+a*b^3*(2*A*c+B*d-2*C*c)*(c^2+d^2)-a^4*d*(3*c^2*C-B*c*d+(A+2*C)*d
^2)-a^2*b^2*(4*A*c^2*d+6*A*d^3+B*c^3-B*c*d^2+2*C*c^2*d)-b^4*(d*(2*A*c^2+3*A
*d^2+C*c^2)-B*(c^3+2*c*d^2)))/(a^2+b^2)^2/(-a*d+b*c)^3/(c^2+d^2)/f/(c+d*tan
(f*x+e))+1/2*(-A*b^2+a*(B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))^2
/(c+d*tan(f*x+e))+1/2*(-5*a^3*b*B*d+3*a^4*C*d-b^4*(-3*A*d+2*B*c)-a*b^3*(4*A
*c+B*d-4*C*c)+a^2*b^2*(2*B*c+(7*A-C)*d))/(a^2+b^2)^2/(-a*d+b*c)^2/f/(a+b*ta
n(f*x+e))/(c+d*tan(f*x+e))
```

Rubi [A] time = 4.08, antiderivative size = 841, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 3, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3649, 3651, 3530}

$$-\frac{\left(b\left(3Cc^4-4Bdc^3+(5A+C)d^2c^2-2Bd^3c+3Ad^4\right)-ad^2\left(2c(A-C)d-B\left(c^2-d^2\right)\right)\right)\log(c\cos(e+fx)+d\sin}{(bc-ad)^4\left(c^2+d^2\right)^2f}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^3*(c + d*
Tan[e + f*x])^2),x]
```

```
[Out] -(((a^3*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - 3*a*b^2*(c^2*C - 2*B*c*
d - C*d^2 - A*(c^2 - d^2)) + 3*a^2*b*(2*c*(A - C)*d - B*(c^2 - d^2)) - b^3*
(2*c*(A - C)*d - B*(c^2 - d^2)))*x)/((a^2 + b^2)^3*(c^2 + d^2)^2)) - (b*(6*
a^5*b*B*d^2 - 3*a^6*C*d^2 - a^4*b^2*d*(4*B*c + (10*A - C)*d) - b^6*(c*(c*C
- 2*B*d) - A*(c^2 - 3*d^2)) + a*b^5*(2*c*(A - C)*d - B*(3*c^2 - d^2)) + 3*a
^2*b^4*(c*(c*C + 2*B*d) - A*(c^2 + 3*d^2)) + a^3*b^3*(10*c*(A - C)*d + B*(c
^2 + 3*d^2)))*Log[a*Cos[e + f*x] + b*Sin[e + f*x]])/((a^2 + b^2)^3*(b*c - a
```

```
*d)^4*f) - (d^2*(b*(3*c^4*C - 4*B*c^3*d + c^2*(5*A + C)*d^2 - 2*B*c*d^3 + 3
*A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Log[c*Cos[e + f*x] + d*Sin
[e + f*x]])/((b*c - a*d)^4*(c^2 + d^2)^2*f) - (d*(3*a^3*b*B*d*(c^2 + d^2) +
 a*b^3*(2*A*c - 2*c*C + B*d)*(c^2 + d^2) - a^4*d*(3*c^2*C - B*c*d + (A + 2*
C)*d^2) - a^2*b^2*(B*c^3 + 4*A*c^2*d + 2*c^2*C*d - B*c*d^2 + 6*A*d^3) - b^4
*(d*(2*A*c^2 + c^2*C + 3*A*d^2) - B*(c^3 + 2*c*d^2))))/((a^2 + b^2)^2*(b*c
- a*d)^3*(c^2 + d^2)*f*(c + d*Tan[e + f*x])) - (A*b^2 - a*(b*B - a*C))/(2*(
a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])) - (5*
a^3*b*B*d - 3*a^4*C*d + b^4*(2*B*c - 3*A*d) + a*b^3*(4*A*c - 4*c*C + B*d) -
 a^2*b^2*(2*B*c + (7*A - C)*d))/(2*(a^2 + b^2)^2*(b*c - a*d)^2*f*(a + b*Tan
[e + f*x])*(c + d*Tan[e + f*x]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^3(c+d\tan(e+fx))^2}\,dx = -\frac{Ab^2 - a(bB-aC)}{2\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^2(c+d\tan(e+fx))} -$$

$$= -\frac{Ab^2 - a(bB-aC)}{2\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^2(c+d\tan(e+fx))} -$$

$$= -\frac{d\left(3a^3bBd\left(c^2+d^2\right) + ab^3(2Ac-2cC+Bd)\left(c^2+d^2\right) - a^4d\left(3c\right.\right.}{}$$

$$= -\frac{\left(a^3\left(c^2C - 2Bcd - Cd^2 - A\left(c^2-d^2\right)\right) - 3ab^2\left(c^2C - 2Bcd - C\right.\right.}{}$$

$$= -\frac{\left(a^3\left(c^2C - 2Bcd - Cd^2 - A\left(c^2-d^2\right)\right) - 3ab^2\left(c^2C - 2Bcd - C\right.\right.}{}$$

Mathematica [B] time = 8.62, size = 1758, normalized size = 2.09

$$-\frac{Ab^2 - a(bB-aC)}{2\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^2(c+d\tan(e+fx))} - \frac{-\frac{b^2\left(3Adb^2-2aA(bc-ad)-(bB-aC)(2bc+ad)\right)-a\left(2b(Ab-Cb-aB)(b\right.}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))(c+d\tan\right.}}{}$$

Warning: Unable to verify antiderivative.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^3*(
c + d*Tan[e + f*x])^2),x]
```

```
[Out] -1/2*(A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]
)^2*(c + d*Tan[e + f*x])) - (-((-(a*(-3*a*(A*b^2 - a*(b*B - a*C))*d + 2*b*(
A*b - a*B - b*C)*(b*c - a*d))) + b^2*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B
- a*C)*(2*b*c + a*d)))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])*(c +
 d*Tan[e + f*x]))) - (-((-(((b*c - a*d)^3*(-(b^2*(-3*a^2*A*b*c^2 + A*b^3*c^
2 + a^3*B*c^2 - 3*a*b^2*B*c^2 + 3*a^2*b*c^2*C - b^3*c^2*C - 2*a^3*A*c*d + 6
*a*A*b^2*c*d - 6*a^2*b*B*c*d + 2*b^3*B*c*d + 2*a^3*c*C*d - 6*a*b^2*c*C*d +
3*a^2*A*b*d^2 - A*b^3*d^2 - a^3*B*d^2 + 3*a*b^2*B*d^2 - 3*a^2*b*C*d^2 + b^3
*C*d^2)) + Sqrt[-b^2]*(a^3*A*b*c^2 - 3*a*A*b^3*c^2 + 3*a^2*b^2*B*c^2 - b^4*
B*c^2 - a^3*b*c^2*C + 3*a*b^3*c^2*C - 6*a^2*A*b^2*c*d + 2*A*b^4*c*d + 2*a^3
```

```
*b*B*c*d - 6*a*b^3*B*c*d + 6*a^2*b^2*c*C*d - 2*b^4*c*C*d - a^3*A*b*d^2 + 3*
a*A*b^3*d^2 - 3*a^2*b^2*B*d^2 + b^4*B*d^2 + a^3*b*C*d^2 - 3*a*b^3*C*d^2))*L
og[Sqrt[-b^2] - b*Tan[e + f*x]])/(b*(a^2 + b^2)*(c^2 + d^2))) - (2*b^2*(c^2
 + d^2)*(6*a^5*b*B*d^2 - 3*a^6*C*d^2 - a^4*b^2*d*(4*B*c + (10*A - C)*d) - b
^6*(c*(c*C - 2*B*d) - A*(c^2 - 3*d^2)) + a*b^5*(2*c*(A - C)*d - B*(3*c^2 -
d^2)) + 3*a^2*b^4*(c*(c*C + 2*B*d) - A*(c^2 + 3*d^2)) + a^3*b^3*(10*c*(A -
C)*d + B*(c^2 + 3*d^2)))*Log[a + b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d))
 + ((b*c - a*d)^3*(b^2*(-3*a^2*A*b*c^2 + A*b^3*c^2 + a^3*B*c^2 - 3*a*b^2*B*
c^2 + 3*a^2*b*c^2*C - b^3*c^2*C - 2*a^3*A*c*d + 6*a*A*b^2*c*d - 6*a^2*b*B*c
*d + 2*b^3*B*c*d + 2*a^3*c*C*d - 6*a*b^2*c*C*d + 3*a^2*A*b*d^2 - A*b^3*d^2
- a^3*B*d^2 + 3*a*b^2*B*d^2 - 3*a^2*b*C*d^2 + b^3*C*d^2) + Sqrt[-b^2]*(a^3*
A*b*c^2 - 3*a*A*b^3*c^2 + 3*a^2*b^2*B*c^2 - b^4*B*c^2 - a^3*b*c^2*C + 3*a*b
^3*c^2*C - 6*a^2*A*b^2*c*d + 2*A*b^4*c*d + 2*a^3*b*B*c*d - 6*a*b^3*B*c*d +
6*a^2*b^2*c*C*d - 2*b^4*c*C*d - a^3*A*b*d^2 + 3*a*A*b^3*d^2 - 3*a^2*b^2*B*d
^2 + b^4*B*d^2 + a^3*b*C*d^2 - 3*a*b^3*C*d^2))*Log[Sqrt[-b^2] + b*Tan[e + f
*x]])/(b*(a^2 + b^2)*(c^2 + d^2)) - (2*b*(a^2 + b^2)^2*d^2*(b*(3*c^4*C - 4*
B*c^3*d + c^2*(5*A + C)*d^2 - 2*B*c*d^3 + 3*A*d^4) - a*d^2*(2*c*(A - C)*d -
 B*(c^2 - d^2)))*Log[c + d*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/(b*(-(
b*c) + a*d)*(c^2 + d^2)*f)) - (d^2*((-(b*c) - a*d)*(-3*a*(A*b^2 - a*(b*B -
a*C))*d + 2*b*(A*b - a*B - b*C)*(b*c - a*d)) + (2*b^2*d - a*(b*c - a*d))*(3
*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d))) - c*(d*(b*c - a*
d)*(-3*b*(A*b^2 - a*(b*B - a*C))*d - 2*a*(A*b - a*B - b*C)*(b*c - a*d) + b*
(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d))) - 2*c*d*(-(a*(
-3*a*(A*b^2 - a*(b*B - a*C))*d + 2*b*(A*b - a*B - b*C)*(b*c - a*d))) + b^2*
(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))))/((-(b*c) + a
*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])))/((a^2 + b^2)*(b*c - a*d)))/(2*(a^2
 + b^2)*(b*c - a*d))
```

fricas [B] time = 17.36, size = 9594, normalized size = 11.41

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] -1/2*((3*C*a^4*b^5 - 5*B*a^3*b^6 + (7*A - 3*C)*a^2*b^7 + B*a*b^8 + A*b^9)*c
^7 - 2*(5*C*a^5*b^4 - 7*B*a^4*b^5 + (9*A - C)*a^3*b^6 - B*a^2*b^7 + 3*A*a*b
^8)*c^6*d + (7*C*a^6*b^3 - 9*B*a^5*b^4 + (11*A + 7*C)*a^4*b^5 - 13*B*a^3*b^
6 + (19*A - 6*C)*a^2*b^7 + 2*B*a*b^8 + 2*A*b^9)*c^5*d^2 - 4*(5*C*a^5*b^4 -
7*B*a^4*b^5 + (9*A - C)*a^3*b^6 - B*a^2*b^7 + 3*A*a*b^8)*c^4*d^3 - (2*C*a^8
*b - 8*C*a^6*b^3 + 18*B*a^5*b^4 - (22*A - C)*a^4*b^5 + 11*B*a^3*b^6 - (17*A
 - 5*C)*a^2*b^7 - B*a*b^8 - A*b^9)*c^3*d^4 + 2*(C*a^9 + B*a^8*b + 3*C*a^7*b
^2 + 3*B*a^6*b^3 - 2*C*a^5*b^4 + 10*B*a^4*b^5 - (9*A - 2*C)*a^3*b^6 + 2*B*a
^2*b^7 - 3*A*a*b^8)*c^2*d^5 - (2*B*a^9 + 2*A*a^8*b + 6*B*a^7*b^2 + (6*A - 7
```

*C)*a^6*b^3 + 15*B*a^5*b^4 - (5*A + C)*a^4*b^5 + 5*B*a^3*b^6 - 3*A*a^2*b^7)
*c*d^6 + 2*(A*a^9 + 3*A*a^7*b^2 + 3*A*a^5*b^4 + A*a^3*b^6)*d^7 - ((C*a^4*b^
5 - 3*B*a^3*b^6 + 5*(A - C)*a^2*b^7 + 3*B*a*b^8 - A*b^9)*c^6*d - 2*(3*C*a^5
*b^4 - 5*B*a^4*b^5 + (7*A - 3*C)*a^3*b^6 + B*a^2*b^7 + A*a*b^8)*c^5*d^2 + (
3*C*a^6*b^3 - 7*B*a^5*b^4 + (9*A - 5*C)*a^4*b^5 - 7*B*a^3*b^6 + (13*A - 16*
C)*a^2*b^7 + 6*B*a*b^8 - 2*(A + C)*b^9)*c^4*d^3 + 2*(C*a^7*b^2 + B*a^6*b^3
- 3*C*a^5*b^4 + 13*B*a^4*b^5 - (14*A - 9*C)*a^3*b^6 + B*a^2*b^7 - (2*A - C)
*a*b^8 + B*b^9)*c^3*d^4 - (2*B*a^7*b^2 + 2*(A - 5*C)*a^6*b^3 + 20*B*a^5*b^4
- (12*A - C)*a^4*b^5 + 11*B*a^3*b^6 - 5*(A - C)*a^2*b^7 - B*a*b^8 + 3*A*b^
9)*c^2*d^5 + 2*(A*a^7*b^2 + 3*(A - C)*a^5*b^4 + 5*B*a^4*b^5 - (4*A - 3*C)*a
^3*b^6 - B*a^2*b^7)*c*d^6 + (5*C*a^6*b^3 - 7*B*a^5*b^4 + (9*A - C)*a^4*b^5
- B*a^3*b^6 + 3*A*a^2*b^7)*d^7 + 2*(((A - C)*a^3*b^6 + 3*B*a^2*b^7 - 3*(A -
C)*a*b^8 - B*b^9)*c^6*d - 2*(2*(A - C)*a^4*b^5 + 5*B*a^3*b^6 - 3*(A - C)*a
^2*b^7 + B*a*b^8 - (A - C)*b^9)*c^5*d^2 + (6*(A - C)*a^5*b^4 + 10*B*a^4*b^5
+ 5*(A - C)*a^3*b^6 + 15*B*a^2*b^7 - 5*(A - C)*a*b^8 + B*b^9)*c^4*d^3 - 4*
((A - C)*a^6*b^3 + 5*(A - C)*a^4*b^5 + 5*B*a^3*b^6 + B*a*b^8)*c^3*d^4 + ((A
- C)*a^7*b^2 - 5*B*a^6*b^3 + 15*(A - C)*a^5*b^4 + 5*B*a^4*b^5 + 10*(A - C)
*a^3*b^6 + 6*B*a^2*b^7)*c^2*d^5 + 2*(B*a^7*b^2 - (A - C)*a^6*b^3 + 3*B*a^5*
b^4 - 5*(A - C)*a^4*b^5 - 2*B*a^3*b^6)*c*d^6 - ((A - C)*a^7*b^2 + 3*B*a^6*b
^3 - 3*(A - C)*a^5*b^4 - B*a^4*b^5)*d^7)*f*x)*tan(f*x + e)^3 - 2*(((A - C)*
a^5*b^4 + 3*B*a^4*b^5 - 3*(A - C)*a^3*b^6 - B*a^2*b^7)*c^7 - 2*(2*(A - C)*a
^6*b^3 + 5*B*a^5*b^4 - 3*(A - C)*a^4*b^5 + B*a^3*b^6 - (A - C)*a^2*b^7)*c^6
d + (6(A - C)*a^7*b^2 + 10*B*a^6*b^3 + 5*(A - C)*a^5*b^4 + 15*B*a^4*b^5 -
5*(A - C)*a^3*b^6 + B*a^2*b^7)*c^5*d^2 - 4*((A - C)*a^8*b + 5*(A - C)*a^6*
b^3 + 5*B*a^5*b^4 + B*a^3*b^6)*c^4*d^3 + ((A - C)*a^9 - 5*B*a^8*b + 15*(A -
C)*a^7*b^2 + 5*B*a^6*b^3 + 10*(A - C)*a^5*b^4 + 6*B*a^4*b^5)*c^3*d^4 + 2*(
B*a^9 - (A - C)*a^8*b + 3*B*a^7*b^2 - 5*(A - C)*a^6*b^3 - 2*B*a^5*b^4)*c^2*
d^5 - ((A - C)*a^9 + 3*B*a^8*b - 3*(A - C)*a^7*b^2 - B*a^6*b^3)*c*d^6)*f*x
- ((C*a^4*b^5 - 3*B*a^3*b^6 + 5*(A - C)*a^2*b^7 + 3*B*a*b^8 - A*b^9)*c^7 -
2*(2*C*a^5*b^4 - 3*B*a^4*b^5 + 4*A*a^3*b^6 - 2*B*a^2*b^7 + 2*(2*A - C)*a*b^
8 + B*b^9)*c^6*d - (3*C*a^6*b^3 - 5*B*a^5*b^4 + (7*A - 13*C)*a^4*b^5 + 19*B
*a^3*b^6 - (25*A - 14*C)*a^2*b^7 - 6*B*a*b^8 - 2*A*b^9)*c^5*d^2 + 2*(C*a^7*
b^2 - 4*B*a^6*b^3 + (5*A - 13*C)*a^5*b^4 + 9*B*a^4*b^5 - (11*A + 6*C)*a^3*b
^6 + 5*B*a^2*b^7 - 2*(5*A - C)*a*b^8 - 2*B*b^9)*c^4*d^3 + (4*C*a^8*b + 4*B*
a^7*b^2 + 8*C*a^6*b^3 + 22*B*a^5*b^4 - (14*A - 41*C)*a^4*b^5 - 17*B*a^3*b^6
+ (35*A - 3*C)*a^2*b^7 + 7*B*a*b^8 + (7*A + 2*C)*b^9)*c^3*d^4 - 2*(2*B*a^8
*b + (2*A - 5*C)*a^7*b^2 + 15*B*a^6*b^3 - (4*A - 11*C)*a^5*b^4 + (16*A + 3*
C)*a^3*b^6 + B*a^2*b^7 + (10*A - C)*a*b^8 + 2*B*b^9)*c^2*d^5 + (4*A*a^8*b +
2*B*a^7*b^2 + (14*A - 3*C)*a^6*b^3 + 11*B*a^5*b^4 + 11*(A + C)*a^4*b^5 - 7
*B*a^3*b^6 + (25*A - 4*C)*a^2*b^7 + 2*B*a*b^8 + 6*A*b^9)*c*d^6 - 2*((A - 3*
C)*a^7*b^2 + 4*B*a^6*b^3 - (2*A - 3*C)*a^5*b^4 - 3*B*a^4*b^5 + 6*A*a^3*b^6
- B*a^2*b^7 + 3*A*a*b^8)*d^7 + 2*(((A - C)*a^3*b^6 + 3*B*a^2*b^7 - 3*(A - C
)*a*b^8 - B*b^9)*c^7 - 2*((A - C)*a^4*b^5 + 2*B*a^3*b^6 + 2*B*a*b^8 - (A -
C)*b^9)*c^6*d - (2*(A - C)*a^5*b^4 + 10*B*a^4*b^5 - 17*(A - C)*a^3*b^6 - 11
*B*a^2*b^7 + (A - C)*a*b^8 - B*b^9)*c^5*d^2 + 2*(4*(A - C)*a^6*b^3 + 10*B*a

```
^5*b^4 - 5*(A - C)*a^4*b^5 + 5*B*a^3*b^6 - 5*(A - C)*a^2*b^7 - B*a*b^8)*c^4
*d^3 - (7*(A - C)*a^7*b^2 + 5*B*a^6*b^3 + 25*(A - C)*a^5*b^4 + 35*B*a^4*b^5
 - 10*(A - C)*a^3*b^6 + 2*B*a^2*b^7)*c^3*d^4 + 2*((A - C)*a^8*b - 4*B*a^7*b
^2 + 14*(A - C)*a^6*b^3 + 8*B*a^5*b^4 + 5*(A - C)*a^4*b^5 + 4*B*a^3*b^6)*c^
2*d^5 + (4*B*a^8*b - 5*(A - C)*a^7*b^2 + 9*B*a^6*b^3 - 17*(A - C)*a^5*b^4 -
 7*B*a^4*b^5)*c*d^6 - 2*((A - C)*a^8*b + 3*B*a^7*b^2 - 3*(A - C)*a^6*b^3 -
B*a^5*b^4)*d^7)*f*x)*tan(f*x + e)^2 + ((B*a^5*b^4 - 3*(A - C)*a^4*b^5 - 3*B
*a^3*b^6 + (A - C)*a^2*b^7)*c^7 - 2*(2*B*a^6*b^3 - 5*(A - C)*a^5*b^4 - 3*B*
a^4*b^5 - (A - C)*a^3*b^6 - B*a^2*b^7)*c^6*d - (3*C*a^8*b - 6*B*a^7*b^2 + (
10*A - C)*a^6*b^3 - 5*B*a^5*b^4 + 3*(5*A - 2*C)*a^4*b^5 + 5*B*a^3*b^6 + (A
+ 2*C)*a^2*b^7)*c^5*d^2 - 4*(2*B*a^6*b^3 - 5*(A - C)*a^5*b^4 - 3*B*a^4*b^5
- (A - C)*a^3*b^6 - B*a^2*b^7)*c^4*d^3 - (6*C*a^8*b - 12*B*a^7*b^2 + 2*(10*
A - C)*a^6*b^3 - 7*B*a^5*b^4 + 3*(7*A - C)*a^4*b^5 + B*a^3*b^6 + (5*A + C)*
a^2*b^7)*c^3*d^4 - 2*(2*B*a^6*b^3 - 5*(A - C)*a^5*b^4 - 3*B*a^4*b^5 - (A -
C)*a^3*b^6 - B*a^2*b^7)*c^2*d^5 - (3*C*a^8*b - 6*B*a^7*b^2 + (10*A - C)*a^6
*b^3 - 3*B*a^5*b^4 + 9*A*a^4*b^5 - B*a^3*b^6 + 3*A*a^2*b^7)*c*d^6 + ((B*a^3
*b^6 - 3*(A - C)*a^2*b^7 - 3*B*a*b^8 + (A - C)*b^9)*c^6*d - 2*(2*B*a^4*b^5
- 5*(A - C)*a^3*b^6 - 3*B*a^2*b^7 - (A - C)*a*b^8 - B*b^9)*c^5*d^2 - (3*C*a
^6*b^3 - 6*B*a^5*b^4 + (10*A - C)*a^4*b^5 - 5*B*a^3*b^6 + 3*(5*A - 2*C)*a^2
*b^7 + 5*B*a*b^8 + (A + 2*C)*b^9)*c^4*d^3 - 4*(2*B*a^4*b^5 - 5*(A - C)*a^3*
b^6 - 3*B*a^2*b^7 - (A - C)*a*b^8 - B*b^9)*c^3*d^4 - (6*C*a^6*b^3 - 12*B*a^
5*b^4 + 2*(10*A - C)*a^4*b^5 - 7*B*a^3*b^6 + 3*(7*A - C)*a^2*b^7 + B*a*b^8
+ (5*A + C)*b^9)*c^2*d^5 - 2*(2*B*a^4*b^5 - 5*(A - C)*a^3*b^6 - 3*B*a^2*b^7
 - (A - C)*a*b^8 - B*b^9)*c*d^6 - (3*C*a^6*b^3 - 6*B*a^5*b^4 + (10*A - C)*a
^4*b^5 - 3*B*a^3*b^6 + 9*A*a^2*b^7 - B*a*b^8 + 3*A*b^9)*d^7)*tan(f*x + e)^3
 + ((B*a^3*b^6 - 3*(A - C)*a^2*b^7 - 3*B*a*b^8 + (A - C)*b^9)*c^7 - 2*(B*a^
4*b^5 - 2*(A - C)*a^3*b^6 - 2*(A - C)*a*b^8 - B*b^9)*c^6*d - (3*C*a^6*b^3 +
 2*B*a^5*b^4 - (10*A - 19*C)*a^4*b^5 - 17*B*a^3*b^6 + (11*A - 2*C)*a^2*b^7
+ B*a*b^8 + (A + 2*C)*b^9)*c^5*d^2 - 2*(3*C*a^7*b^2 - 6*B*a^6*b^3 + (10*A -
 C)*a^5*b^4 - B*a^4*b^5 + (5*A + 4*C)*a^3*b^6 - B*a^2*b^7 - (A - 4*C)*a*b^8
 - 2*B*b^9)*c^4*d^3 - (6*C*a^6*b^3 + 4*B*a^5*b^4 - 2*(10*A - 19*C)*a^4*b^5
- 31*B*a^3*b^6 + (13*A + 5*C)*a^2*b^7 - 7*B*a*b^8 + (5*A + C)*b^9)*c^3*d^4
- 2*(6*C*a^7*b^2 - 12*B*a^6*b^3 + 2*(10*A - C)*a^5*b^4 - 5*B*a^4*b^5 + 2*(8
*A + C)*a^3*b^6 - 2*B*a^2*b^7 + 2*(2*A + C)*a*b^8 - B*b^9)*c^2*d^5 - (3*C*a
^6*b^3 + 2*B*a^5*b^4 - (10*A - 19*C)*a^4*b^5 - 15*B*a^3*b^6 + (5*A + 4*C)*a
^2*b^7 - 5*B*a*b^8 + 3*A*b^9)*c*d^6 - 2*(3*C*a^7*b^2 - 6*B*a^6*b^3 + (10*A
- C)*a^5*b^4 - 3*B*a^4*b^5 + 9*A*a^3*b^6 - B*a^2*b^7 + 3*A*a*b^8)*d^7)*tan(
f*x + e)^2 + (2*(B*a^4*b^5 - 3*(A - C)*a^3*b^6 - 3*B*a^2*b^7 + (A - C)*a*b^
8)*c^7 - (7*B*a^5*b^4 - 17*(A - C)*a^4*b^5 - 9*B*a^3*b^6 - 5*(A - C)*a^2*b^
7 - 4*B*a*b^8)*c^6*d - 2*(3*C*a^7*b^2 - 4*B*a^6*b^3 + (5*A + 4*C)*a^5*b^4 -
 8*B*a^4*b^5 + (14*A - 5*C)*a^3*b^6 + 4*B*a^2*b^7 + (A + 2*C)*a*b^8)*c^5*d^
2 - (3*C*a^8*b - 6*B*a^7*b^2 + (10*A - C)*a^6*b^3 + 11*B*a^5*b^4 - (25*A -
34*C)*a^4*b^5 - 19*B*a^3*b^6 - (7*A - 10*C)*a^2*b^7 - 8*B*a*b^8)*c^4*d^3 -
2*(6*C*a^7*b^2 - 8*B*a^6*b^3 + 2*(5*A + 4*C)*a^5*b^4 - 13*B*a^4*b^5 + (19*A
 - C)*a^3*b^6 - B*a^2*b^7 + (5*A + C)*a*b^8)*c^3*d^4 - (6*C*a^8*b - 12*B*a^
```

```
7*b^2 + 2*(10*A - C)*a^6*b^3 + B*a^5*b^4 + (A + 17*C)*a^4*b^5 - 11*B*a^3*b^
6 + (A + 5*C)*a^2*b^7 - 4*B*a*b^8)*c^2*d^5 - 2*(3*C*a^7*b^2 - 4*B*a^6*b^3 +
 (5*A + 4*C)*a^5*b^4 - 6*B*a^4*b^5 + (8*A + C)*a^3*b^6 - 2*B*a^2*b^7 + 3*A*
a*b^8)*c*d^6 - (3*C*a^8*b - 6*B*a^7*b^2 + (10*A - C)*a^6*b^3 - 3*B*a^5*b^4
+ 9*A*a^4*b^5 - B*a^3*b^6 + 3*A*a^2*b^7)*d^7)*tan(f*x + e))*log((b^2*tan(f*
x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) + (3*(C*a^8*b +
3*C*a^6*b^3 + 3*C*a^4*b^5 + C*a^2*b^7)*c^5*d^2 - 4*(B*a^8*b + 3*B*a^6*b^3 +
 3*B*a^4*b^5 + B*a^2*b^7)*c^4*d^3 + (B*a^9 + (5*A + C)*a^8*b + 3*B*a^7*b^2
+ 3*(5*A + C)*a^6*b^3 + 3*B*a^5*b^4 + 3*(5*A + C)*a^4*b^5 + B*a^3*b^6 + (5*
A + C)*a^2*b^7)*c^3*d^4 - 2*((A - C)*a^9 + B*a^8*b + 3*(A - C)*a^7*b^2 + 3*
B*a^6*b^3 + 3*(A - C)*a^5*b^4 + 3*B*a^4*b^5 + (A - C)*a^3*b^6 + B*a^2*b^7)*
c^2*d^5 - (B*a^9 - 3*A*a^8*b + 3*B*a^7*b^2 - 9*A*a^6*b^3 + 3*B*a^5*b^4 - 9*
A*a^4*b^5 + B*a^3*b^6 - 3*A*a^2*b^7)*c*d^6 + (3*(C*a^6*b^3 + 3*C*a^4*b^5 +
3*C*a^2*b^7 + C*b^9)*c^4*d^3 - 4*(B*a^6*b^3 + 3*B*a^4*b^5 + 3*B*a^2*b^7 + B
*b^9)*c^3*d^4 + (B*a^7*b^2 + (5*A + C)*a^6*b^3 + 3*B*a^5*b^4 + 3*(5*A + C)*
a^4*b^5 + 3*B*a^3*b^6 + 3*(5*A + C)*a^2*b^7 + B*a*b^8 + (5*A + C)*b^9)*c^2*
d^5 - 2*((A - C)*a^7*b^2 + B*a^6*b^3 + 3*(A - C)*a^5*b^4 + 3*B*a^4*b^5 + 3*
(A - C)*a^3*b^6 + 3*B*a^2*b^7 + (A - C)*a*b^8 + B*b^9)*c*d^6 - (B*a^7*b^2 -
 3*A*a^6*b^3 + 3*B*a^5*b^4 - 9*A*a^4*b^5 + 3*B*a^3*b^6 - 9*A*a^2*b^7 + B*a*
b^8 - 3*A*b^9)*d^7)*tan(f*x + e)^3 + (3*(C*a^6*b^3 + 3*C*a^4*b^5 + 3*C*a^2*
b^7 + C*b^9)*c^5*d^2 + 2*(3*C*a^7*b^2 - 2*B*a^6*b^3 + 9*C*a^5*b^4 - 6*B*a^4
*b^5 + 9*C*a^3*b^6 - 6*B*a^2*b^7 + 3*C*a*b^8 - 2*B*b^9)*c^4*d^3 - (7*B*a^7*
b^2 - (5*A + C)*a^6*b^3 + 21*B*a^5*b^4 - 3*(5*A + C)*a^4*b^5 + 21*B*a^3*b^6
 - 3*(5*A + C)*a^2*b^7 + 7*B*a*b^8 - (5*A + C)*b^9)*c^3*d^4 + 2*(B*a^8*b +
2*(2*A + C)*a^7*b^2 + 2*B*a^6*b^3 + 6*(2*A + C)*a^5*b^4 + 6*(2*A + C)*a^3*b
^6 - 2*B*a^2*b^7 + 2*(2*A + C)*a*b^8 - B*b^9)*c^2*d^5 - (4*(A - C)*a^8*b +
5*B*a^7*b^2 + 3*(3*A - 4*C)*a^6*b^3 + 15*B*a^5*b^4 + 3*(A - 4*C)*a^4*b^5 +
15*B*a^3*b^6 - (5*A + 4*C)*a^2*b^7 + 5*B*a*b^8 - 3*A*b^9)*c*d^6 - 2*(B*a^8*
b - 3*A*a^7*b^2 + 3*B*a^6*b^3 - 9*A*a^5*b^4 + 3*B*a^4*b^5 - 9*A*a^3*b^6 + B
*a^2*b^7 - 3*A*a*b^8)*d^7)*tan(f*x + e)^2 + (6*(C*a^7*b^2 + 3*C*a^5*b^4 + 3
*C*a^3*b^6 + C*a*b^8)*c^5*d^2 + (3*C*a^8*b - 8*B*a^7*b^2 + 9*C*a^6*b^3 - 24
*B*a^5*b^4 + 9*C*a^4*b^5 - 24*B*a^3*b^6 + 3*C*a^2*b^7 - 8*B*a*b^8)*c^4*d^3
- 2*(B*a^8*b - (5*A + C)*a^7*b^2 + 3*B*a^6*b^3 - 3*(5*A + C)*a^5*b^4 + 3*B*
a^4*b^5 - 3*(5*A + C)*a^3*b^6 + B*a^2*b^7 - (5*A + C)*a*b^8)*c^3*d^4 + (B*a
^9 + (A + 5*C)*a^8*b - B*a^7*b^2 + 3*(A + 5*C)*a^6*b^3 - 9*B*a^5*b^4 + 3*(A
 + 5*C)*a^4*b^5 - 11*B*a^3*b^6 + (A + 5*C)*a^2*b^7 - 4*B*a*b^8)*c^2*d^5 - 2
*((A - C)*a^9 + 2*B*a^8*b - 3*C*a^7*b^2 + 6*B*a^6*b^3 - 3*(2*A + C)*a^5*b^4
 + 6*B*a^4*b^5 - (8*A + C)*a^3*b^6 + 2*B*a^2*b^7 - 3*A*a*b^8)*c*d^6 - (B*a^
9 - 3*A*a^8*b + 3*B*a^7*b^2 - 9*A*a^6*b^3 + 3*B*a^5*b^4 - 9*A*a^4*b^5 + B*a
^3*b^6 - 3*A*a^2*b^7)*d^7)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*ta
n(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) - (2*(C*a^5*b^4 - 2*B*a^4*b^5 + 3*(
A - C)*a^3*b^6 + 3*B*a^2*b^7 - (3*A - 2*C)*a*b^8 - B*b^9)*c^7 - (8*C*a^6*b^
3 - 12*B*a^5*b^4 + (16*A - 9*C)*a^4*b^5 + 7*B*a^3*b^6 - (5*A - C)*a^2*b^7 +
 B*a*b^8 - 3*A*b^9)*c^6*d + 2*(3*C*a^7*b^2 - 4*B*a^6*b^3 + (5*A + 4*C)*a^5*
b^4 - 8*B*a^4*b^5 + (12*A - 7*C)*a^3*b^6 + 6*B*a^2*b^7 - (5*A - 4*C)*a*b^8
```

- 2*B*b^9)*c^5*d^2 - (2*C*a^8*b + 29*C*a^6*b^3 - 33*B*a^5*b^4 + (43*A - 11*
C)*a^4*b^5 + 11*B*a^3*b^6 - (5*A - 4*C)*a^2*b^7 + 2*B*a*b^8 - 6*A*b^9)*c^4*
d^3 + 2*(C*a^9 + B*a^8*b + 11*C*a^7*b^2 - 5*B*a^6*b^3 + 2*(5*A + 7*C)*a^5*b
^4 - 7*B*a^4*b^5 + (15*A + 2*C)*a^3*b^6 + 4*B*a^2*b^7 - (A - 4*C)*a*b^8 - B
*b^9)*c^3*d^4 - (2*B*a^9 + 2*(A + 2*C)*a^8*b + 10*B*a^7*b^2 + 2*(3*A + 17*C
)*a^6*b^3 - 12*B*a^5*b^4 + (44*A + 5*C)*a^4*b^5 + 15*B*a^3*b^6 + (7*A + 5*C
)*a^2*b^7 + 5*B*a*b^8 - 3*A*b^9)*c^2*d^5 + 2*(A*a^9 + 2*B*a^8*b + (5*A + 3*
C)*a^7*b^2 + 2*B*a^6*b^3 + 2*(7*A + C)*a^5*b^4 + 2*B*a^4*b^5 + (13*A - C)*a
^3*b^6 + 2*B*a^2*b^7 + 3*A*a*b^8)*c*d^6 - (4*A*a^8*b + (12*A + 7*C)*a^6*b^3
 - 9*B*a^5*b^4 + (23*A + C)*a^4*b^5 - 3*B*a^3*b^6 + 9*A*a^2*b^7)*d^7 + 2*(2
*((A - C)*a^4*b^5 + 3*B*a^3*b^6 - 3*(A - C)*a^2*b^7 - B*a*b^8)*c^7 - (7*(A
- C)*a^5*b^4 + 17*B*a^4*b^5 - 9*(A - C)*a^3*b^6 + 5*B*a^2*b^7 - 4*(A - C)*a
*b^8)*c^6*d + 2*(4*(A - C)*a^6*b^3 + 5*B*a^5*b^4 + 8*(A - C)*a^4*b^5 + 14*B
*a^3*b^6 - 4*(A - C)*a^2*b^7 + B*a*b^8)*c^5*d^2 - (2*(A - C)*a^7*b^2 - 10*B
*a^6*b^3 + 35*(A - C)*a^5*b^4 + 25*B*a^4*b^5 + 5*(A - C)*a^3*b^6 + 7*B*a^2*
b^7)*c^4*d^3 - 2*((A - C)*a^8*b + 5*B*a^7*b^2 - 5*(A - C)*a^6*b^3 + 5*B*a^5
b^4 - 10(A - C)*a^4*b^5 - 4*B*a^3*b^6)*c^3*d^4 + ((A - C)*a^9 - B*a^8*b +
 11*(A - C)*a^7*b^2 + 17*B*a^6*b^3 - 10*(A - C)*a^5*b^4 - 2*B*a^4*b^5)*c^2*
d^5 + 2*(B*a^9 - 2*(A - C)*a^8*b - 2*(A - C)*a^6*b^3 - B*a^5*b^4)*c*d^6 - (
(A - C)*a^9 + 3*B*a^8*b - 3*(A - C)*a^7*b^2 - B*a^6*b^3)*d^7)*f*x)*tan(f*x
+ e))/(((a^6*b^6 + 3*a^4*b^8 + 3*a^2*b^10 + b^12)*c^8*d - 4*(a^7*b^5 + 3*a^
5*b^7 + 3*a^3*b^9 + a*b^11)*c^7*d^2 + 2*(3*a^8*b^4 + 10*a^6*b^6 + 12*a^4*b^
8 + 6*a^2*b^10 + b^12)*c^6*d^3 - 4*(a^9*b^3 + 5*a^7*b^5 + 9*a^5*b^7 + 7*a^3
*b^9 + 2*a*b^11)*c^5*d^4 + (a^10*b^2 + 15*a^8*b^4 + 40*a^6*b^6 + 40*a^4*b^8
 + 15*a^2*b^10 + b^12)*c^4*d^5 - 4*(2*a^9*b^3 + 7*a^7*b^5 + 9*a^5*b^7 + 5*a
^3*b^9 + a*b^11)*c^3*d^6 + 2*(a^10*b^2 + 6*a^8*b^4 + 12*a^6*b^6 + 10*a^4*b^
8 + 3*a^2*b^10)*c^2*d^7 - 4*(a^9*b^3 + 3*a^7*b^5 + 3*a^5*b^7 + a^3*b^9)*c*d
^8 + (a^10*b^2 + 3*a^8*b^4 + 3*a^6*b^6 + a^4*b^8)*d^9)*f*tan(f*x + e)^3 + (
(a^6*b^6 + 3*a^4*b^8 + 3*a^2*b^10 + b^12)*c^9 - 2*(a^7*b^5 + 3*a^5*b^7 + 3*
a^3*b^9 + a*b^11)*c^8*d - 2*(a^8*b^4 + 2*a^6*b^6 - 2*a^2*b^10 - b^12)*c^7*d
^2 + 4*(2*a^9*b^3 + 5*a^7*b^5 + 3*a^5*b^7 - a^3*b^9 - a*b^11)*c^6*d^3 - (7*
a^10*b^2 + 25*a^8*b^4 + 32*a^6*b^6 + 16*a^4*b^8 + a^2*b^10 - b^12)*c^5*d^4
+ 2*(a^11*b + 11*a^9*b^3 + 26*a^7*b^5 + 22*a^5*b^7 + 5*a^3*b^9 - a*b^11)*c^
4*d^5 - 2*(7*a^10*b^2 + 22*a^8*b^4 + 24*a^6*b^6 + 10*a^4*b^8 + a^2*b^10)*c^
3*d^6 + 4*(a^11*b + 5*a^9*b^3 + 9*a^7*b^5 + 7*a^5*b^7 + 2*a^3*b^9)*c^2*d^7
- 7*(a^10*b^2 + 3*a^8*b^4 + 3*a^6*b^6 + a^4*b^8)*c*d^8 + 2*(a^11*b + 3*a^9*
b^3 + 3*a^7*b^5 + a^5*b^7)*d^9)*f*tan(f*x + e)^2 + (2*(a^7*b^5 + 3*a^5*b^7
+ 3*a^3*b^9 + a*b^11)*c^9 - 7*(a^8*b^4 + 3*a^6*b^6 + 3*a^4*b^8 + a^2*b^10)*
c^8*d + 4*(2*a^9*b^3 + 7*a^7*b^5 + 9*a^5*b^7 + 5*a^3*b^9 + a*b^11)*c^7*d^2
- 2*(a^10*b^2 + 10*a^8*b^4 + 24*a^6*b^6 + 22*a^4*b^8 + 7*a^2*b^10)*c^6*d^3
- 2*(a^11*b - 5*a^9*b^3 - 22*a^7*b^5 - 26*a^5*b^7 - 11*a^3*b^9 - a*b^11)*c^
5*d^4 + (a^12 - a^10*b^2 - 16*a^8*b^4 - 32*a^6*b^6 - 25*a^4*b^8 - 7*a^2*b^1
0)*c^4*d^5 - 4*(a^11*b + a^9*b^3 - 3*a^7*b^5 - 5*a^5*b^7 - 2*a^3*b^9)*c^3*d
^6 + 2*(a^12 + 2*a^10*b^2 - 2*a^6*b^6 - a^4*b^8)*c^2*d^7 - 2*(a^11*b + 3*a^
9*b^3 + 3*a^7*b^5 + a^5*b^7)*c*d^8 + (a^12 + 3*a^10*b^2 + 3*a^8*b^4 + a^6*b

```
^6)*d^9)*f*tan(f*x + e) + ((a^8*b^4 + 3*a^6*b^6 + 3*a^4*b^8 + a^2*b^10)*c^9
 - 4*(a^9*b^3 + 3*a^7*b^5 + 3*a^5*b^7 + a^3*b^9)*c^8*d + 2*(3*a^10*b^2 + 10
*a^8*b^4 + 12*a^6*b^6 + 6*a^4*b^8 + a^2*b^10)*c^7*d^2 - 4*(a^11*b + 5*a^9*b
^3 + 9*a^7*b^5 + 7*a^5*b^7 + 2*a^3*b^9)*c^6*d^3 + (a^12 + 15*a^10*b^2 + 40*
a^8*b^4 + 40*a^6*b^6 + 15*a^4*b^8 + a^2*b^10)*c^5*d^4 - 4*(2*a^11*b + 7*a^9
*b^3 + 9*a^7*b^5 + 5*a^5*b^7 + a^3*b^9)*c^4*d^5 + 2*(a^12 + 6*a^10*b^2 + 12
*a^8*b^4 + 10*a^6*b^6 + 3*a^4*b^8)*c^3*d^6 - 4*(a^11*b + 3*a^9*b^3 + 3*a^7*
b^5 + a^5*b^7)*c^2*d^7 + (a^12 + 3*a^10*b^2 + 3*a^8*b^4 + a^6*b^6)*c*d^8)*f
)
```

giac [B] time = 25.98, size = 3176, normalized size = 3.78

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^3*c^2 - C*a^3*c^2 + 3*B*a^2*b*c^2 - 3*A*a*b^2*c^2 + 3*C*a*b^2*c
^2 - B*b^3*c^2 + 2*B*a^3*c*d - 6*A*a^2*b*c*d + 6*C*a^2*b*c*d - 6*B*a*b^2*c*
d + 2*A*b^3*c*d - 2*C*b^3*c*d - A*a^3*d^2 + C*a^3*d^2 - 3*B*a^2*b*d^2 + 3*A
*a*b^2*d^2 - 3*C*a*b^2*d^2 + B*b^3*d^2)*(f*x + e)/(a^6*c^4 + 3*a^4*b^2*c^4
+ 3*a^2*b^4*c^4 + b^6*c^4 + 2*a^6*c^2*d^2 + 6*a^4*b^2*c^2*d^2 + 6*a^2*b^4*c
^2*d^2 + 2*b^6*c^2*d^2 + a^6*d^4 + 3*a^4*b^2*d^4 + 3*a^2*b^4*d^4 + b^6*d^4)
 + (B*a^3*c^2 - 3*A*a^2*b*c^2 + 3*C*a^2*b*c^2 - 3*B*a*b^2*c^2 + A*b^3*c^2 -
 C*b^3*c^2 - 2*A*a^3*c*d + 2*C*a^3*c*d - 6*B*a^2*b*c*d + 6*A*a*b^2*c*d - 6*
C*a*b^2*c*d + 2*B*b^3*c*d - B*a^3*d^2 + 3*A*a^2*b*d^2 - 3*C*a^2*b*d^2 + 3*B
*a*b^2*d^2 - A*b^3*d^2 + C*b^3*d^2)*log(tan(f*x + e)^2 + 1)/(a^6*c^4 + 3*a^
4*b^2*c^4 + 3*a^2*b^4*c^4 + b^6*c^4 + 2*a^6*c^2*d^2 + 6*a^4*b^2*c^2*d^2 + 6
*a^2*b^4*c^2*d^2 + 2*b^6*c^2*d^2 + a^6*d^4 + 3*a^4*b^2*d^4 + 3*a^2*b^4*d^4
+ b^6*d^4) - 2*(B*a^3*b^5*c^2 - 3*A*a^2*b^6*c^2 + 3*C*a^2*b^6*c^2 - 3*B*a*b
^7*c^2 + A*b^8*c^2 - C*b^8*c^2 - 4*B*a^4*b^4*c*d + 10*A*a^3*b^5*c*d - 10*C*
a^3*b^5*c*d + 6*B*a^2*b^6*c*d + 2*A*a*b^7*c*d - 2*C*a*b^7*c*d + 2*B*b^8*c*d
 - 3*C*a^6*b^2*d^2 + 6*B*a^5*b^3*d^2 - 10*A*a^4*b^4*d^2 + C*a^4*b^4*d^2 + 3
*B*a^3*b^5*d^2 - 9*A*a^2*b^6*d^2 + B*a*b^7*d^2 - 3*A*b^8*d^2)*log(abs(b*tan
(f*x + e) + a))/(a^6*b^5*c^4 + 3*a^4*b^7*c^4 + 3*a^2*b^9*c^4 + b^11*c^4 - 4
*a^7*b^4*c^3*d - 12*a^5*b^6*c^3*d - 12*a^3*b^8*c^3*d - 4*a*b^10*c^3*d + 6*a
^8*b^3*c^2*d^2 + 18*a^6*b^5*c^2*d^2 + 18*a^4*b^7*c^2*d^2 + 6*a^2*b^9*c^2*d^
2 - 4*a^9*b^2*c*d^3 - 12*a^7*b^4*c*d^3 - 12*a^5*b^6*c*d^3 - 4*a^3*b^8*c*d^3
 + a^10*b*d^4 + 3*a^8*b^3*d^4 + 3*a^6*b^5*d^4 + a^4*b^7*d^4) - 2*(3*C*b*c^4
*d^3 - 4*B*b*c^3*d^4 + B*a*c^2*d^5 + 5*A*b*c^2*d^5 + C*b*c^2*d^5 - 2*A*a*c*
d^6 + 2*C*a*c*d^6 - 2*B*b*c*d^6 - B*a*d^7 + 3*A*b*d^7)*log(abs(d*tan(f*x +
e) + c))/(b^4*c^8*d - 4*a*b^3*c^7*d^2 + 6*a^2*b^2*c^6*d^3 + 2*b^4*c^6*d^3 -
 4*a^3*b*c^5*d^4 - 8*a*b^3*c^5*d^4 + a^4*c^4*d^5 + 12*a^2*b^2*c^4*d^5 + b^4
*c^4*d^5 - 8*a^3*b*c^3*d^6 - 4*a*b^3*c^3*d^6 + 2*a^4*c^2*d^7 + 6*a^2*b^2*c^
```

```
2*d^7 - 4*a^3*b*c*d^8 + a^4*d^9) + 2*(3*C*b*c^4*d^3*tan(f*x + e) - 4*B*b*c^
3*d^4*tan(f*x + e) + B*a*c^2*d^5*tan(f*x + e) + 5*A*b*c^2*d^5*tan(f*x + e)
+ C*b*c^2*d^5*tan(f*x + e) - 2*A*a*c*d^6*tan(f*x + e) + 2*C*a*c*d^6*tan(f*x
 + e) - 2*B*b*c*d^6*tan(f*x + e) - B*a*d^7*tan(f*x + e) + 3*A*b*d^7*tan(f*x
 + e) + 4*C*b*c^5*d^2 - C*a*c^4*d^3 - 5*B*b*c^4*d^3 + 2*B*a*c^3*d^4 + 6*A*b
*c^3*d^4 + 2*C*b*c^3*d^4 - 3*A*a*c^2*d^5 + C*a*c^2*d^5 - 3*B*b*c^2*d^5 + 4*
A*b*c*d^6 - A*a*d^7)/((b^4*c^8 - 4*a*b^3*c^7*d + 6*a^2*b^2*c^6*d^2 + 2*b^4*
c^6*d^2 - 4*a^3*b*c^5*d^3 - 8*a*b^3*c^5*d^3 + a^4*c^4*d^4 + 12*a^2*b^2*c^4*
d^4 + b^4*c^4*d^4 - 8*a^3*b*c^3*d^5 - 4*a*b^3*c^3*d^5 + 2*a^4*c^2*d^6 + 6*a
^2*b^2*c^2*d^6 - 4*a^3*b*c*d^7 + a^4*d^8)*(d*tan(f*x + e) + c)) + (3*B*a^3*
b^6*c^2*tan(f*x + e)^2 - 9*A*a^2*b^7*c^2*tan(f*x + e)^2 + 9*C*a^2*b^7*c^2*t
an(f*x + e)^2 - 9*B*a*b^8*c^2*tan(f*x + e)^2 + 3*A*b^9*c^2*tan(f*x + e)^2 -
 3*C*b^9*c^2*tan(f*x + e)^2 - 12*B*a^4*b^5*c*d*tan(f*x + e)^2 + 30*A*a^3*b^
6*c*d*tan(f*x + e)^2 - 30*C*a^3*b^6*c*d*tan(f*x + e)^2 + 18*B*a^2*b^7*c*d*t
an(f*x + e)^2 + 6*A*a*b^8*c*d*tan(f*x + e)^2 - 6*C*a*b^8*c*d*tan(f*x + e)^2
 + 6*B*b^9*c*d*tan(f*x + e)^2 - 9*C*a^6*b^3*d^2*tan(f*x + e)^2 + 18*B*a^5*b
^4*d^2*tan(f*x + e)^2 - 30*A*a^4*b^5*d^2*tan(f*x + e)^2 + 3*C*a^4*b^5*d^2*t
an(f*x + e)^2 + 9*B*a^3*b^6*d^2*tan(f*x + e)^2 - 27*A*a^2*b^7*d^2*tan(f*x +
 e)^2 + 3*B*a*b^8*d^2*tan(f*x + e)^2 - 9*A*b^9*d^2*tan(f*x + e)^2 + 8*B*a^4
*b^5*c^2*tan(f*x + e) - 22*A*a^3*b^6*c^2*tan(f*x + e) + 22*C*a^3*b^6*c^2*ta
n(f*x + e) - 18*B*a^2*b^7*c^2*tan(f*x + e) + 2*A*a*b^8*c^2*tan(f*x + e) - 2
*C*a*b^8*c^2*tan(f*x + e) - 2*B*b^9*c^2*tan(f*x + e) + 4*C*a^6*b^3*c*d*tan(
f*x + e) - 32*B*a^5*b^4*c*d*tan(f*x + e) + 72*A*a^4*b^5*c*d*tan(f*x + e) -
60*C*a^4*b^5*c*d*tan(f*x + e) + 28*B*a^3*b^6*c*d*tan(f*x + e) + 28*A*a^2*b^
7*c*d*tan(f*x + e) - 16*C*a^2*b^7*c*d*tan(f*x + e) + 12*B*a*b^8*c*d*tan(f*x
 + e) + 4*A*b^9*c*d*tan(f*x + e) - 22*C*a^7*b^2*d^2*tan(f*x + e) + 42*B*a^6
*b^3*d^2*tan(f*x + e) - 68*A*a^5*b^4*d^2*tan(f*x + e) + 2*C*a^5*b^4*d^2*tan
(f*x + e) + 26*B*a^4*b^5*d^2*tan(f*x + e) - 66*A*a^3*b^6*d^2*tan(f*x + e) +
 8*B*a^2*b^7*d^2*tan(f*x + e) - 22*A*a*b^8*d^2*tan(f*x + e) - C*a^6*b^3*c^2
 + 6*B*a^5*b^4*c^2 - 14*A*a^4*b^5*c^2 + 11*C*a^4*b^5*c^2 - 7*B*a^3*b^6*c^2
- 3*A*a^2*b^7*c^2 - B*a*b^8*c^2 - A*b^9*c^2 + 6*C*a^7*b^2*c*d - 22*B*a^6*b^
3*c*d + 44*A*a^5*b^4*c*d - 26*C*a^5*b^4*c*d + 6*B*a^4*b^5*c*d + 26*A*a^3*b^
6*c*d - 8*C*a^3*b^6*c*d + 4*B*a^2*b^7*c*d + 6*A*a*b^8*c*d - 14*C*a^8*b*d^2
+ 25*B*a^7*b^2*d^2 - 39*A*a^6*b^3*d^2 - 3*C*a^6*b^3*d^2 + 19*B*a^5*b^4*d^2
- 41*A*a^4*b^5*d^2 - C*a^4*b^5*d^2 + 6*B*a^3*b^6*d^2 - 14*A*a^2*b^7*d^2)/((
a^6*b^4*c^4 + 3*a^4*b^6*c^4 + 3*a^2*b^8*c^4 + b^10*c^4 - 4*a^7*b^3*c^3*d -
12*a^5*b^5*c^3*d - 12*a^3*b^7*c^3*d - 4*a*b^9*c^3*d + 6*a^8*b^2*c^2*d^2 + 1
8*a^6*b^4*c^2*d^2 + 18*a^4*b^6*c^2*d^2 + 6*a^2*b^8*c^2*d^2 - 4*a^9*b*c*d^3
- 12*a^7*b^3*c*d^3 - 12*a^5*b^5*c*d^3 - 4*a^3*b^7*c*d^3 + a^10*d^4 + 3*a^8*
b^2*d^4 + 3*a^6*b^4*d^4 + a^4*b^6*d^4)*(b*tan(f*x + e) + a)^2))/f
```

maple [B] time = 0.62, size = 3364, normalized size = 4.00

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e))^2,x
)
```

```
[Out] -1/f*d^2/(a*d-b*c)^3/(c^2+d^2)/(c+d*tan(f*x+e))*c^2*C+1/f/(a^2+b^2)^3/(c^2+
d^2)^2*A*arctan(tan(f*x+e))*a^3*c^2-1/f/(a^2+b^2)^3/(c^2+d^2)^2*A*arctan(ta
n(f*x+e))*a^3*d^2-1/f/(a^2+b^2)^3/(c^2+d^2)^2*B*arctan(tan(f*x+e))*b^3*c^2+
1/f/(a^2+b^2)^3/(c^2+d^2)^2*B*arctan(tan(f*x+e))*b^3*d^2+2/f*b^4/(a*d-b*c)^
3/(a^2+b^2)^2/(a+b*tan(f*x+e))*A*a*c+3/f*b^2/(a*d-b*c)^3/(a^2+b^2)^2/(a+b*t
an(f*x+e))*a^3*B*d-1/f*b^3/(a*d-b*c)^3/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a^2*c
+1/f*b^4/(a*d-b*c)^3/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*a*d-2/f*b/(a*d-b*c)^3/(
a^2+b^2)^2/(a+b*tan(f*x+e))*a^4*C*d-2/f*b^4/(a*d-b*c)^3/(a^2+b^2)^2/(a+b*ta
n(f*x+e))*C*a*c+10/f*b^3/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a^4*d
^2-2/f*d^5/(a*d-b*c)^4/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*a*c-3/f*d^2/(a*d-b*
c)^4/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*C*b*c^4-1/f*d^4/(a*d-b*c)^4/(c^2+d^2)^2
*ln(c+d*tan(f*x+e))*C*b*c^2+2/f*d^5/(a*d-b*c)^4/(c^2+d^2)^2*ln(c+d*tan(f*x+
e))*B*b*c+2/f/(a^2+b^2)^3/(c^2+d^2)^2*A*arctan(tan(f*x+e))*b^3*c*d+2/f/(a^2
+b^2)^3/(c^2+d^2)^2*B*arctan(tan(f*x+e))*a^3*c*d+3/f/(a^2+b^2)^3/(c^2+d^2)^
2*B*arctan(tan(f*x+e))*a^2*b*c^2-3/f/(a^2+b^2)^3/(c^2+d^2)^2*B*arctan(tan(f
*x+e))*a^2*b*d^2+3/f/(a^2+b^2)^3/(c^2+d^2)^2*C*arctan(tan(f*x+e))*a*b^2*c^2
-3/f/(a^2+b^2)^3/(c^2+d^2)^2*C*arctan(tan(f*x+e))*a*b^2*d^2-2/f/(a^2+b^2)^3
/(c^2+d^2)^2*C*arctan(tan(f*x+e))*b^3*c*d-1/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1+
tan(f*x+e)^2)*A*a^3*c*d-3/2/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*
a*b^2*c^2+3/2/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a*b^2*d^2+1/f/
(a^2+b^2)^3/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*b^3*c*d+3/2/f/(a^2+b^2)^3/(c^2
+d^2)^2*ln(1+tan(f*x+e)^2)*C*a^2*b*c^2-3/2/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1+t
an(f*x+e)^2)*C*a^2*b*d^2-3/f/(a^2+b^2)^3/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a
*b^2*c^2+3/f/(a^2+b^2)^3/(c^2+d^2)^2*A*arctan(tan(f*x+e))*a*b^2*d^2+1/f/(a^
2+b^2)^3/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*a^3*c*d-4/f*b^3/(a*d-b*c)^3/(a^2+
b^2)^2/(a+b*tan(f*x+e))*A*a^2*d+2/f*d^5/(a*d-b*c)^4/(c^2+d^2)^2*ln(c+d*tan(
f*x+e))*A*a*c-5/f*d^4/(a*d-b*c)^4/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*A*b*c^2-1/
f*d^4/(a*d-b*c)^4/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*a*c^2+4/f*d^3/(a*d-b*c)^
4/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*b*c^3-6/f*b^2/(a*d-b*c)^4/(a^2+b^2)^3*ln
(a+b*tan(f*x+e))*a^5*B*d^2-1/f*b^4/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e
))*B*a^3*c^2-3/f*b^4/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^3*d^2+3
/f*b^6/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*a*B*c^2-1/f*b^6/(a*d-b*c)
^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a*d^2-2/f*b^7/(a*d-b*c)^4/(a^2+b^2)^3*l
n(a+b*tan(f*x+e))*B*c*d+3/f*b/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*a^
6*C*d^2-1/f*b^3/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*a^4*C*d^2-3/f*b^
5/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*a^2*c^2-3/2/f/(a^2+b^2)^3/(c
^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*a^2*b*c^2+3/2/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1
+tan(f*x+e)^2)*A*a^2*b*d^2+3/f*b^5/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e
))*A*a^2*c^2+9/f*b^5/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a^2*d^2+6
/f/(a^2+b^2)^3/(c^2+d^2)^2*C*arctan(tan(f*x+e))*a^2*b*c*d-10/f*b^4/(a*d-b*c
)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a^3*c*d+4/f*b^3/(a*d-b*c)^4/(a^2+b^2)^
3*ln(a+b*tan(f*x+e))*B*a^4*c*d-6/f/(a^2+b^2)^3/(c^2+d^2)^2*B*arctan(tan(f*x
```

```
+e))*a*b^2*c*d-3/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*a*b^2*c*d-2
/f*b^6/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*a*c*d+3/f/(a^2+b^2)^3/(
c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*a*b^2*c*d-3/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1+
tan(f*x+e)^2)*B*a^2*b*c*d-6/f/(a^2+b^2)^3/(c^2+d^2)^2*A*arctan(tan(f*x+e))*
a^2*b*c*d+10/f*b^4/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*C*a^3*c*d-6/f
*b^5/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*B*a^2*c*d+3/f*b^7/(a*d-b*c)
^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*A*d^2+1/f*b^7/(a*d-b*c)^4/(a^2+b^2)^3*ln(
a+b*tan(f*x+e))*C*c^2+1/2/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*b^
3*c^2-1/2/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*A*b^3*d^2+1/2/f/(a^2
+b^2)^3/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*B*a^3*c^2-1/2/f/(a^2+b^2)^3/(c^2+d^2
)^2*ln(1+tan(f*x+e)^2)*B*a^3*d^2-1/2/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1+tan(f*x
+e)^2)*C*b^3*c^2+1/2/f/(a^2+b^2)^3/(c^2+d^2)^2*ln(1+tan(f*x+e)^2)*C*b^3*d^2
+1/2/f*b^2/(a*d-b*c)^2/(a^2+b^2)/(a+b*tan(f*x+e))^2*B*a-1/2/f*b/(a*d-b*c)^2
/(a^2+b^2)/(a+b*tan(f*x+e))^2*a^2*C-3/f*d^6/(a*d-b*c)^4/(c^2+d^2)^2*ln(c+d*
tan(f*x+e))*A*b+1/f*d^6/(a*d-b*c)^4/(c^2+d^2)^2*ln(c+d*tan(f*x+e))*B*a+1/f*
d^3/(a*d-b*c)^3/(c^2+d^2)/(c+d*tan(f*x+e))*B*c-1/f/(a^2+b^2)^3/(c^2+d^2)^2*
C*arctan(tan(f*x+e))*a^3*c^2+1/f/(a^2+b^2)^3/(c^2+d^2)^2*C*arctan(tan(f*x+e
))*a^3*d^2-2/f*b^5/(a*d-b*c)^3/(a^2+b^2)^2/(a+b*tan(f*x+e))*A*d+1/f*b^5/(a*
d-b*c)^3/(a^2+b^2)^2/(a+b*tan(f*x+e))*B*c-1/f*b^7/(a*d-b*c)^4/(a^2+b^2)^3*l
n(a+b*tan(f*x+e))*A*c^2+2/f*b^6/(a*d-b*c)^4/(a^2+b^2)^3*ln(a+b*tan(f*x+e))*
C*a*c*d-1/2/f*b^3/(a*d-b*c)^2/(a^2+b^2)/(a+b*tan(f*x+e))^2*A-1/f*d^4/(a*d-b
*c)^3/(c^2+d^2)/(c+d*tan(f*x+e))*A
```

maxima [B] time = 0.70, size = 2519, normalized size = 3.00

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^3/(c+d*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^3 + 3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c^2 + 2*(B*a^3
- 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c*d - ((A - C)*a^3 + 3*B*a^2*b
 - 3*(A - C)*a*b^2 - B*b^3)*d^2)*(f*x + e)/((a^6 + 3*a^4*b^2 + 3*a^2*b^4 +
b^6)*c^4 + 2*(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)*c^2*d^2 + (a^6 + 3*a^4*b^2
 + 3*a^2*b^4 + b^6)*d^4) - 2*((B*a^3*b^4 - 3*(A - C)*a^2*b^5 - 3*B*a*b^6 +
(A - C)*b^7)*c^2 - 2*(2*B*a^4*b^3 - 5*(A - C)*a^3*b^4 - 3*B*a^2*b^5 - (A -
C)*a*b^6 - B*b^7)*c*d - (3*C*a^6*b - 6*B*a^5*b^2 + (10*A - C)*a^4*b^3 - 3*B
*a^3*b^4 + 9*A*a^2*b^5 - B*a*b^6 + 3*A*b^7)*d^2)*log(b*tan(f*x + e) + a)/((
a^6*b^4 + 3*a^4*b^6 + 3*a^2*b^8 + b^10)*c^4 - 4*(a^7*b^3 + 3*a^5*b^5 + 3*a^
3*b^7 + a*b^9)*c^3*d + 6*(a^8*b^2 + 3*a^6*b^4 + 3*a^4*b^6 + a^2*b^8)*c^2*d^
2 - 4*(a^9*b + 3*a^7*b^3 + 3*a^5*b^5 + a^3*b^7)*c*d^3 + (a^10 + 3*a^8*b^2 +
 3*a^6*b^4 + a^4*b^6)*d^4) - 2*(3*C*b*c^4*d^2 - 4*B*b*c^3*d^3 + (B*a + (5*A
 + C)*b)*c^2*d^4 - 2*((A - C)*a + B*b)*c*d^5 - (B*a - 3*A*b)*d^6)*log(d*tan
(f*x + e) + c)/(b^4*c^8 - 4*a*b^3*c^7*d - 4*a^3*b*c*d^7 + a^4*d^8 + 2*(3*a^
```

```
2*b^2 + b^4)*c^6*d^2 - 4*(a^3*b + 2*a*b^3)*c^5*d^3 + (a^4 + 12*a^2*b^2 + b^
4)*c^4*d^4 - 4*(2*a^3*b + a*b^3)*c^3*d^5 + 2*(a^4 + 3*a^2*b^2)*c^2*d^6) + (
(B*a^3 - 3*(A - C)*a^2*b - 3*B*a*b^2 + (A - C)*b^3)*c^2 - 2*((A - C)*a^3 +
3*B*a^2*b - 3*(A - C)*a*b^2 - B*b^3)*c*d - (B*a^3 - 3*(A - C)*a^2*b - 3*B*a
*b^2 + (A - C)*b^3)*d^2)*log(tan(f*x + e)^2 + 1)/((a^6 + 3*a^4*b^2 + 3*a^2*
b^4 + b^6)*c^4 + 2*(a^6 + 3*a^4*b^2 + 3*a^2*b^4 + b^6)*c^2*d^2 + (a^6 + 3*a
^4*b^2 + 3*a^2*b^4 + b^6)*d^4) - ((C*a^4*b^2 - 3*B*a^3*b^3 + (5*A - 3*C)*a^
2*b^4 + B*a*b^5 + A*b^6)*c^4 - (5*C*a^5*b - 7*B*a^4*b^2 + (9*A + C)*a^3*b^3
 - 3*B*a^2*b^4 + 5*A*a*b^5)*c^3*d - (2*C*a^6 + 3*C*a^4*b^2 + 3*B*a^3*b^3 -
5*(A - C)*a^2*b^4 - B*a*b^5 - A*b^6)*c^2*d^2 + (2*B*a^6 - 5*C*a^5*b + 11*B*
a^4*b^2 - (9*A + C)*a^3*b^3 + 5*B*a^2*b^4 - 5*A*a*b^5)*c*d^3 - 2*(A*a^6 + 2
*A*a^4*b^2 + A*a^2*b^4)*d^4 - 2*((B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^3*
d + (3*C*a^4*b^2 - 3*B*a^3*b^3 + 2*(2*A + C)*a^2*b^4 - B*a*b^5 + (2*A + C)*
b^6)*c^2*d^2 - (B*a^4*b^2 + B*a^2*b^4 + 2*(A - C)*a*b^5 + 2*B*b^6)*c*d^3 +
((A + 2*C)*a^4*b^2 - 3*B*a^3*b^3 + 6*A*a^2*b^4 - B*a*b^5 + 3*A*b^6)*d^4)*ta
n(f*x + e)^2 - (2*(B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c^4 + 3*(C*a^4*b^2
- B*a^3*b^3 + (A + C)*a^2*b^4 - B*a*b^5 + A*b^6)*c^3*d + (9*C*a^5*b - 7*B*a
^4*b^2 + 9*(A + C)*a^3*b^3 - B*a^2*b^4 + (A + 8*C)*a*b^5 - 2*B*b^6)*c^2*d^2
 - (4*B*a^5*b - 3*C*a^4*b^2 + 11*B*a^3*b^3 - 3*(A + C)*a^2*b^4 + 7*B*a*b^5
- 3*A*b^6)*c*d^3 + ((4*A + 5*C)*a^5*b - 7*B*a^4*b^2 + (17*A + C)*a^3*b^3 -
3*B*a^2*b^4 + 9*A*a*b^5)*d^4)*tan(f*x + e))/((a^6*b^3 + 2*a^4*b^5 + a^2*b^7
)*c^6 - 3*(a^7*b^2 + 2*a^5*b^4 + a^3*b^6)*c^5*d + (3*a^8*b + 7*a^6*b^3 + 5*
a^4*b^5 + a^2*b^7)*c^4*d^2 - (a^9 + 5*a^7*b^2 + 7*a^5*b^4 + 3*a^3*b^6)*c^3*
d^3 + 3*(a^8*b + 2*a^6*b^3 + a^4*b^5)*c^2*d^4 - (a^9 + 2*a^7*b^2 + a^5*b^4)
*c*d^5 + ((a^4*b^5 + 2*a^2*b^7 + b^9)*c^5*d - 3*(a^5*b^4 + 2*a^3*b^6 + a*b^
8)*c^4*d^2 + (3*a^6*b^3 + 7*a^4*b^5 + 5*a^2*b^7 + b^9)*c^3*d^3 - (a^7*b^2 +
 5*a^5*b^4 + 7*a^3*b^6 + 3*a*b^8)*c^2*d^4 + 3*(a^6*b^3 + 2*a^4*b^5 + a^2*b^
7)*c*d^5 - (a^7*b^2 + 2*a^5*b^4 + a^3*b^6)*d^6)*tan(f*x + e)^3 + ((a^4*b^5
+ 2*a^2*b^7 + b^9)*c^6 - (a^5*b^4 + 2*a^3*b^6 + a*b^8)*c^5*d - (3*a^6*b^3 +
 5*a^4*b^5 + a^2*b^7 - b^9)*c^4*d^2 + (5*a^7*b^2 + 9*a^5*b^4 + 3*a^3*b^6 -
a*b^8)*c^3*d^3 - (2*a^8*b + 7*a^6*b^3 + 8*a^4*b^5 + 3*a^2*b^7)*c^2*d^4 + 5*
(a^7*b^2 + 2*a^5*b^4 + a^3*b^6)*c*d^5 - 2*(a^8*b + 2*a^6*b^3 + a^4*b^5)*d^6
)*tan(f*x + e)^2 + (2*(a^5*b^4 + 2*a^3*b^6 + a*b^8)*c^6 - 5*(a^6*b^3 + 2*a^
4*b^5 + a^2*b^7)*c^5*d + (3*a^7*b^2 + 8*a^5*b^4 + 7*a^3*b^6 + 2*a*b^8)*c^4*
d^2 + (a^8*b - 3*a^6*b^3 - 9*a^4*b^5 - 5*a^2*b^7)*c^3*d^3 - (a^9 - a^7*b^2
- 5*a^5*b^4 - 3*a^3*b^6)*c^2*d^4 + (a^8*b + 2*a^6*b^3 + a^4*b^5)*c*d^5 - (a
^9 + 2*a^7*b^2 + a^5*b^4)*d^6)*tan(f*x + e)))/f
```

mupad [B] time = 58.47, size = 128667, normalized size = 152.99

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^3*(c + d*
tan(e + f*x))^2),x)
```

```
[Out] (symsum(log((24*A^3*a^3*b^7*d^9 + 27*A^3*a^5*b^5*d^9 + B^3*a^2*b^8*d^9 + 4*
B^3*a^4*b^6*d^9 + 7*B^3*a^6*b^4*d^9 + 3*A^3*b^10*c^3*d^6 - A^3*b^10*c^5*d^4
 + 4*B^3*b^10*c^2*d^7 + 6*B^3*b^10*c^4*d^5 + C^3*b^10*c^5*d^4 + 9*A^2*B*b^1
0*d^9 + 9*A^3*a*b^9*d^9 + 16*A^3*a^2*b^8*c^3*d^6 + 3*A^3*a^2*b^8*c^5*d^4 +
26*A^3*a^3*b^7*c^2*d^7 - 6*A^3*a^3*b^7*c^4*d^5 - 11*A^3*a^4*b^6*c^3*d^6 + 3
1*A^3*a^5*b^5*c^2*d^7 + 5*B^3*a^2*b^8*c^2*d^7 + 6*B^3*a^2*b^8*c^4*d^5 + 28*
B^3*a^3*b^7*c^3*d^6 + 7*B^3*a^3*b^7*c^5*d^4 - 14*B^3*a^4*b^6*c^2*d^7 - 20*B
^3*a^4*b^6*c^4*d^5 + 19*B^3*a^5*b^5*c^3*d^6 + 9*B^3*a^6*b^4*c^2*d^7 - 7*C^3
*a^2*b^8*c^3*d^6 - 3*C^3*a^2*b^8*c^5*d^4 + C^3*a^3*b^7*c^2*d^7 + 15*C^3*a^3
*b^7*c^4*d^5 + 6*C^3*a^3*b^7*c^6*d^3 - 28*C^3*a^4*b^6*c^3*d^6 - 24*C^3*a^4*
b^6*c^5*d^4 - 4*C^3*a^5*b^5*c^2*d^7 + 3*C^3*a^6*b^4*c^3*d^6 - 9*C^3*a^7*b^3
*c^2*d^7 - 9*C^3*a^7*b^3*c^4*d^5 - 6*A*B^2*a*b^9*d^9 - 9*A^2*C*a*b^9*d^9 -
12*A*B^2*b^10*c*d^8 + 4*B^3*a*b^9*c*d^8 - 20*A*B^2*a^3*b^7*d^9 - 28*A*B^2*a
^5*b^5*d^9 + 6*A*B^2*a^7*b^3*d^9 + 21*A^2*B*a^2*b^8*d^9 + 13*A^2*B*a^4*b^6*
d^9 - 27*A^2*B*a^6*b^4*d^9 - 3*A*C^2*a^3*b^7*d^9 - 9*A*C^2*a^7*b^3*d^9 - 21
*A^2*C*a^3*b^7*d^9 - 27*A^2*C*a^5*b^5*d^9 + 9*A^2*C*a^7*b^3*d^9 - 17*A*B^2*
b^10*c^3*d^6 + 3*A*B^2*b^10*c^5*d^4 + B*C^2*a^4*b^6*d^9 + 3*B*C^2*a^8*b^2*d
^9 + 12*A^2*B*b^10*c^2*d^7 - 7*A^2*B*b^10*c^4*d^5 - B^2*C*a^3*b^7*d^9 - 2*B
^2*C*a^5*b^5*d^9 - 9*B^2*C*a^7*b^3*d^9 + 3*A*C^2*b^10*c^3*d^6 - 3*A*C^2*b^1
0*c^5*d^4 - 6*A^2*C*b^10*c^3*d^6 + 3*A^2*C*b^10*c^5*d^4 - B*C^2*b^10*c^4*d^
5 + 3*B*C^2*b^10*c^6*d^3 - 4*B^2*C*b^10*c^3*d^6 - 9*B^2*C*b^10*c^5*d^4 + 3*
A^3*a*b^9*c^2*d^7 - 10*A^3*a*b^9*c^4*d^5 - 3*A^3*a^2*b^8*c*d^8 - 31*A^3*a^4
*b^6*c*d^8 - 8*A^3*a^6*b^4*c*d^8 + B^3*a*b^9*c^3*d^6 - 5*B^3*a*b^9*c^5*d^4
+ 11*B^3*a^3*b^7*c*d^8 + 5*B^3*a^5*b^5*c*d^8 - 6*B^3*a^7*b^3*c*d^8 - 2*C^3*
a*b^9*c^4*d^5 - 6*C^3*a*b^9*c^6*d^3 - 2*C^3*a^4*b^6*c*d^8 - C^3*a^6*b^4*c*d
^8 - 3*C^3*a^8*b^2*c*d^8 - 60*A*B^2*a^2*b^8*c^3*d^6 - 21*A*B^2*a^2*b^8*c^5*
d^4 - 4*A*B^2*a^3*b^7*c^2*d^7 + 44*A*B^2*a^3*b^7*c^4*d^5 + 25*A*B^2*a^4*b^6
*c^3*d^6 + 4*A*B^2*a^4*b^6*c^5*d^4 - 77*A*B^2*a^5*b^5*c^2*d^7 - 17*A*B^2*a^
5*b^5*c^4*d^5 + 28*A*B^2*a^6*b^4*c^3*d^6 - 6*A*B^2*a^7*b^3*c^2*d^7 + 71*A^2
*B*a^2*b^8*c^2*d^7 + 16*A^2*B*a^2*b^8*c^4*d^5 - 116*A^2*B*a^3*b^7*c^3*d^6 -
 9*A^2*B*a^3*b^7*c^5*d^4 + 86*A^2*B*a^4*b^6*c^2*d^7 + 35*A^2*B*a^4*b^6*c^4*
d^5 - 37*A^2*B*a^5*b^5*c^3*d^6 - 13*A^2*B*a^6*b^4*c^2*d^7 + 30*A*C^2*a^2*b^
8*c^3*d^6 + 9*A*C^2*a^2*b^8*c^5*d^4 - 30*A*C^2*a^3*b^7*c^2*d^7 - 63*A*C^2*a
^3*b^7*c^4*d^5 - 12*A*C^2*a^3*b^7*c^6*d^3 + 45*A*C^2*a^4*b^6*c^3*d^6 + 48*A
*C^2*a^4*b^6*c^5*d^4 - 15*A*C^2*a^5*b^5*c^2*d^7 - 27*A*C^2*a^5*b^5*c^4*d^5
- 6*A*C^2*a^6*b^4*c^3*d^6 + 9*A*C^2*a^7*b^3*c^4*d^5 - 39*A^2*C*a^2*b^8*c^3*
d^6 - 9*A^2*C*a^2*b^8*c^5*d^4 + 3*A^2*C*a^3*b^7*c^2*d^7 + 54*A^2*C*a^3*b^7*
c^4*d^5 + 6*A^2*C*a^3*b^7*c^6*d^3 - 6*A^2*C*a^4*b^6*c^3*d^6 - 24*A^2*C*a^4*
b^6*c^5*d^4 - 12*A^2*C*a^5*b^5*c^2*d^7 + 27*A^2*C*a^5*b^5*c^4*d^5 + 3*A^2*C
*a^6*b^4*c^3*d^6 + 9*A^2*C*a^7*b^3*c^2*d^7 + 11*B*C^2*a^2*b^8*c^2*d^7 - 17*
B*C^2*a^2*b^8*c^4*d^5 - 18*B*C^2*a^2*b^8*c^6*d^3 + 16*B*C^2*a^3*b^7*c^3*d^6
 + 39*B*C^2*a^3*b^7*c^5*d^4 + 47*B*C^2*a^4*b^6*c^2*d^7 + 47*B*C^2*a^4*b^6*c
^4*d^5 + 3*B*C^2*a^4*b^6*c^6*d^3 - 25*B*C^2*a^5*b^5*c^3*d^6 - 12*B*C^2*a^5*
b^5*c^5*d^4 + 17*B*C^2*a^6*b^4*c^2*d^7 + 27*B*C^2*a^6*b^4*c^4*d^5 + 12*B*C^
2*a^7*b^3*c^3*d^6 - 3*B*C^2*a^8*b^2*c^2*d^7 + 9*B^2*C*a^2*b^8*c^3*d^6 + 9*B
```

^2*C*a^2*b^8*c^5*d^4 - 35*B^2*C*a^3*b^7*c^2*d^7 - 68*B^2*C*a^3*b^7*c^4*d^5 - 6*B^2*C*a^3*b^7*c^6*d^3 - 16*B^2*C*a^4*b^6*c^3*d^6 + 14*B^2*C*a^4*b^6*c^5*d^4 + 26*B^2*C*a^5*b^5*c^2*d^7 - 4*B^2*C*a^5*b^5*c^4*d^5 - 37*B^2*C*a^6*b^4*c^3*d^6 + 3*B^2*C*a^7*b^3*c^2*d^7 + 6*A*B*C*a^2*b^8*d^9 + 13*A*B*C*a^4*b^6*d^9 + 36*A*B*C*a^6*b^4*d^9 - 3*A*B*C*a^8*b^2*d^9 + 6*A*B*C*b^10*c^2*d^7 + 17*A*B*C*b^10*c^4*d^5 - 3*A*B*C*b^10*c^6*d^3 - 24*A^2*B*a*b^9*c*d^8 + 11*A*B^2*a*b^9*c^2*d^7 + 25*A*B^2*a*b^9*c^4*d^5 - 19*A*B^2*a^2*b^8*c*d^8 + 37*A*B^2*a^4*b^6*c*d^8 + 32*A*B^2*a^6*b^4*c*d^8 - 23*A^2*B*a*b^9*c^3*d^6 + 11*A^2*B*a*b^9*c^5*d^4 - 81*A^2*B*a^3*b^7*c*d^8 - 15*A^2*B*a^5*b^5*c*d^8 + 6*A^2*B*a^7*b^3*c*d^8 - 15*A*C^2*a*b^9*c^2*d^7 - 15*A*C^2*a*b^9*c^4*d^5 + 12*A*C^2*a*b^9*c^6*d^3 - 3*A*C^2*a^2*b^8*c*d^8 - 27*A*C^2*a^4*b^6*c*d^8 - 6*A*C^2*a^6*b^4*c*d^8 + 6*A*C^2*a^8*b^2*c*d^8 + 12*A^2*C*a*b^9*c^2*d^7 + 27*A^2*C*a*b^9*c^4*d^5 - 6*A^2*C*a*b^9*c^6*d^3 + 6*A^2*C*a^2*b^8*c*d^8 + 60*A^2*C*a^4*b^6*c*d^8 + 15*A^2*C*a^6*b^4*c*d^8 - 3*A^2*C*a^8*b^2*c*d^8 + 13*B*C^2*a*b^9*c^3*d^6 + 23*B*C^2*a*b^9*c^5*d^4 + 3*B*C^2*a^3*b^7*c*d^8 + 9*B*C^2*a^5*b^5*c*d^8 + 18*B*C^2*a^7*b^3*c*d^8 - 14*B^2*C*a*b^9*c^2*d^7 - 16*B^2*C*a*b^9*c^4*d^5 + 6*B^2*C*a*b^9*c^6*d^3 - 8*B^2*C*a^2*b^8*c*d^8 - 28*B^2*C*a^4*b^6*c*d^8 - 29*B^2*C*a^6*b^4*c*d^8 + 3*B^2*C*a^8*b^2*c*d^8 - 28*A*B*C*a^2*b^8*c^2*d^7 + 28*A*B*C*a^2*b^8*c^4*d^5 + 18*A*B*C*a^2*b^8*c^6*d^3 + 100*A*B*C*a^3*b^7*c^3*d^6 - 30*A*B*C*a^3*b^7*c^5*d^4 - 79*A*B*C*a^4*b^6*c^2*d^7 - 55*A*B*C*a^4*b^6*c^4*d^5 - 3*A*B*C*a^4*b^6*c^6*d^3 + 62*A*B*C*a^5*b^5*c^3*d^6 + 12*A*B*C*a^5*b^5*c^5*d^4 + 14*A*B*C*a^6*b^4*c^2*d^7 - 18*A*B*C*a^6*b^4*c^4*d^5 - 12*A*B*C*a^7*b^3*c^3*d^6 + 3*A*B*C*a^8*b^2*c^2*d^7 + 24*A*B*C*a*b^9*c*d^8 + 10*A*B*C*a*b^9*c^3*d^6 - 34*A*B*C*a*b^9*c^5*d^4 + 78*A*B*C*a^3*b^7*c*d^8 + 6*A*B*C*a^5*b^5*c*d^8 - 24*A*B*C*a^7*b^3*c*d^8)/(a^14*d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^11*c^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8 + 90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^5*b^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 152*a^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 68*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12*b^2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*a^13*b*c*d^9) - root(640*a^13*b^7*c*d^15*f^4 + 640*a^7*b^13*c^15*d*f^4 + 480*a^15*b^5*c*d^15*f^4 + 480*a^11*b^9*c*d^15*f^4 + 480*a^9*b^11*c^15*d*f^4 + 480*a^5*b^15*c^15*d*f^4 + 192*a^19*b*c^5*d^11*f^4 + 192*a^17*b^3*c*d^15*f^4 + 192*a

^11*b^9*c^15*d*f^4 + 192*a^9*b^11*c*d^15*f^4 + 192*a^3*b^17*c^15*d*f^4 + 19
2*a*b^19*c^11*d^5*f^4 + 128*a^19*b*c^7*d^9*f^4 + 128*a^19*b*c^3*d^13*f^4 +
128*a*b^19*c^13*d^3*f^4 + 128*a*b^19*c^9*d^7*f^4 + 32*a^19*b*c^9*d^7*f^4 +
32*a^13*b^7*c^15*d*f^4 + 32*a^7*b^13*c*d^15*f^4 + 32*a*b^19*c^7*d^9*f^4 + 3
2*a^19*b*c*d^15*f^4 + 32*a*b^19*c^15*d*f^4 - 47088*a^10*b^10*c^8*d^8*f^4 +
42432*a^11*b^9*c^7*d^9*f^4 + 42432*a^9*b^11*c^9*d^7*f^4 + 39328*a^11*b^9*c^
9*d^7*f^4 + 39328*a^9*b^11*c^7*d^9*f^4 - 36912*a^12*b^8*c^8*d^8*f^4 - 36912
*a^8*b^12*c^8*d^8*f^4 - 34256*a^10*b^10*c^10*d^6*f^4 - 34256*a^10*b^10*c^6*
d^10*f^4 - 31152*a^12*b^8*c^6*d^10*f^4 - 31152*a^8*b^12*c^10*d^6*f^4 + 2812
8*a^13*b^7*c^7*d^9*f^4 + 28128*a^7*b^13*c^9*d^7*f^4 + 24160*a^11*b^9*c^5*d^
11*f^4 + 24160*a^9*b^11*c^11*d^5*f^4 - 23088*a^12*b^8*c^10*d^6*f^4 - 23088*
a^8*b^12*c^6*d^10*f^4 + 22272*a^13*b^7*c^9*d^7*f^4 + 22272*a^7*b^13*c^7*d^9
*f^4 + 19072*a^11*b^9*c^11*d^5*f^4 + 19072*a^9*b^11*c^5*d^11*f^4 + 18624*a^
13*b^7*c^5*d^11*f^4 + 18624*a^7*b^13*c^11*d^5*f^4 - 17328*a^14*b^6*c^8*d^8*
f^4 - 17328*a^6*b^14*c^8*d^8*f^4 - 17232*a^14*b^6*c^6*d^10*f^4 - 17232*a^6*
b^14*c^10*d^6*f^4 - 13520*a^12*b^8*c^4*d^12*f^4 - 13520*a^8*b^12*c^12*d^4*f
^4 - 12464*a^10*b^10*c^12*d^4*f^4 - 12464*a^10*b^10*c^4*d^12*f^4 + 10880*a^
15*b^5*c^7*d^9*f^4 + 10880*a^5*b^15*c^9*d^7*f^4 - 9072*a^14*b^6*c^10*d^6*f^
4 - 9072*a^6*b^14*c^6*d^10*f^4 + 8928*a^13*b^7*c^11*d^5*f^4 + 8928*a^7*b^13
*c^5*d^11*f^4 - 8880*a^14*b^6*c^4*d^12*f^4 - 8880*a^6*b^14*c^12*d^4*f^4 + 8
480*a^15*b^5*c^5*d^11*f^4 + 8480*a^5*b^15*c^11*d^5*f^4 + 7200*a^15*b^5*c^9*
d^7*f^4 + 7200*a^5*b^15*c^7*d^9*f^4 - 6912*a^12*b^8*c^12*d^4*f^4 - 6912*a^8
*b^12*c^4*d^12*f^4 + 6400*a^11*b^9*c^3*d^13*f^4 + 6400*a^9*b^11*c^13*d^3*f^
4 + 5920*a^13*b^7*c^3*d^13*f^4 + 5920*a^7*b^13*c^13*d^3*f^4 - 5392*a^16*b^4
*c^6*d^10*f^4 - 5392*a^4*b^16*c^10*d^6*f^4 - 4428*a^16*b^4*c^8*d^8*f^4 - 44
28*a^4*b^16*c^8*d^8*f^4 + 4128*a^11*b^9*c^13*d^3*f^4 + 4128*a^9*b^11*c^3*d^
13*f^4 - 3328*a^16*b^4*c^4*d^12*f^4 - 3328*a^4*b^16*c^12*d^4*f^4 + 3264*a^1
5*b^5*c^3*d^13*f^4 + 3264*a^5*b^15*c^13*d^3*f^4 - 2480*a^12*b^8*c^2*d^14*f^
4 - 2480*a^8*b^12*c^14*d^2*f^4 + 2240*a^15*b^5*c^11*d^5*f^4 + 2240*a^5*b^15
*c^5*d^11*f^4 - 2128*a^14*b^6*c^12*d^4*f^4 - 2128*a^6*b^14*c^4*d^12*f^4 + 2
112*a^17*b^3*c^7*d^9*f^4 + 2112*a^3*b^17*c^9*d^7*f^4 + 2048*a^17*b^3*c^5*d^
11*f^4 + 2048*a^3*b^17*c^11*d^5*f^4 - 2000*a^14*b^6*c^2*d^14*f^4 - 2000*a^6
*b^14*c^14*d^2*f^4 - 1792*a^16*b^4*c^10*d^6*f^4 - 1792*a^4*b^16*c^6*d^10*f^
4 - 1776*a^10*b^10*c^14*d^2*f^4 - 1776*a^10*b^10*c^2*d^14*f^4 + 1472*a^13*b
^7*c^13*d^3*f^4 + 1472*a^7*b^13*c^3*d^13*f^4 + 1088*a^17*b^3*c^9*d^7*f^4 +
1088*a^3*b^17*c^7*d^9*f^4 + 992*a^17*b^3*c^3*d^13*f^4 + 992*a^3*b^17*c^13*d
^3*f^4 - 912*a^16*b^4*c^2*d^14*f^4 - 912*a^4*b^16*c^14*d^2*f^4 - 768*a^18*b
^2*c^6*d^10*f^4 - 768*a^2*b^18*c^10*d^6*f^4 - 688*a^12*b^8*c^14*d^2*f^4 - 6
88*a^8*b^12*c^2*d^14*f^4 - 592*a^18*b^2*c^4*d^12*f^4 - 592*a^2*b^18*c^12*d^
4*f^4 - 472*a^18*b^2*c^8*d^8*f^4 - 472*a^2*b^18*c^8*d^8*f^4 - 280*a^16*b^4*
c^12*d^4*f^4 - 280*a^4*b^16*c^4*d^12*f^4 + 224*a^17*b^3*c^11*d^5*f^4 + 224*
a^15*b^5*c^13*d^3*f^4 + 224*a^5*b^15*c^3*d^13*f^4 + 224*a^3*b^17*c^5*d^11*f
^4 - 208*a^18*b^2*c^2*d^14*f^4 - 208*a^2*b^18*c^14*d^2*f^4 - 112*a^18*b^2*c
^10*d^6*f^4 - 112*a^14*b^6*c^14*d^2*f^4 - 112*a^6*b^14*c^2*d^14*f^4 - 112*a
^2*b^18*c^6*d^10*f^4 - 24*b^20*c^12*d^4*f^4 - 16*b^20*c^14*d^2*f^4 - 16*b^2

0*c^10*d^6*f^4 - 4*b^20*c^8*d^8*f^4 - 24*a^20*c^4*d^12*f^4 - 16*a^20*c^6*d^
10*f^4 - 16*a^20*c^2*d^14*f^4 - 4*a^20*c^8*d^8*f^4 - 80*a^14*b^6*d^16*f^4 -
 60*a^16*b^4*d^16*f^4 - 60*a^12*b^8*d^16*f^4 - 24*a^18*b^2*d^16*f^4 - 24*a^
10*b^10*d^16*f^4 - 4*a^8*b^12*d^16*f^4 - 80*a^6*b^14*c^16*f^4 - 60*a^8*b^12
*c^16*f^4 - 60*a^4*b^16*c^16*f^4 - 24*a^10*b^10*c^16*f^4 - 24*a^2*b^18*c^16
*f^4 - 4*a^12*b^8*c^16*f^4 - 4*b^20*c^16*f^4 - 4*a^20*d^16*f^4 + 56*A*C*a^1
3*b*c*d^11*f^2 - 48*A*C*a*b^13*c^11*d*f^2 + 48*A*C*a*b^13*c*d^11*f^2 + 5904
*B*C*a^7*b^7*c^6*d^6*f^2 - 5016*B*C*a^8*b^6*c^5*d^7*f^2 - 4608*B*C*a^6*b^8*
c^7*d^5*f^2 - 4512*B*C*a^6*b^8*c^5*d^7*f^2 - 4384*B*C*a^8*b^6*c^7*d^5*f^2 +
 3056*B*C*a^7*b^7*c^8*d^4*f^2 + 2256*B*C*a^7*b^7*c^4*d^8*f^2 - 1824*B*C*a^8
*b^6*c^3*d^9*f^2 + 1632*B*C*a^4*b^10*c^9*d^3*f^2 - 1400*B*C*a^3*b^11*c^8*d^
4*f^2 - 1320*B*C*a^11*b^3*c^4*d^8*f^2 - 1248*B*C*a^6*b^8*c^3*d^9*f^2 + 1152
*B*C*a^10*b^4*c^3*d^9*f^2 - 1072*B*C*a^6*b^8*c^9*d^3*f^2 + 1068*B*C*a^9*b^5
*c^6*d^6*f^2 - 1004*B*C*a^5*b^9*c^4*d^8*f^2 - 968*B*C*a^3*b^11*c^6*d^6*f^2
- 864*B*C*a^5*b^9*c^8*d^4*f^2 - 828*B*C*a^9*b^5*c^4*d^8*f^2 - 792*B*C*a^11*
b^3*c^2*d^10*f^2 - 792*B*C*a^3*b^11*c^4*d^8*f^2 - 776*B*C*a^8*b^6*c^9*d^3*f
^2 + 688*B*C*a^4*b^10*c^7*d^5*f^2 - 672*B*C*a^3*b^11*c^10*d^2*f^2 - 592*B*C
*a^9*b^5*c^2*d^10*f^2 + 544*B*C*a^7*b^7*c^10*d^2*f^2 - 492*B*C*a^5*b^9*c^2*
d^10*f^2 + 480*B*C*a^10*b^4*c^5*d^7*f^2 - 392*B*C*a^5*b^9*c^10*d^2*f^2 + 33
2*B*C*a^9*b^5*c^8*d^4*f^2 - 328*B*C*a^11*b^3*c^6*d^6*f^2 + 320*B*C*a^2*b^12
*c^9*d^3*f^2 + 272*B*C*a^12*b^2*c^3*d^9*f^2 - 248*B*C*a^4*b^10*c^5*d^7*f^2
- 248*B*C*a^3*b^11*c^2*d^10*f^2 - 208*B*C*a^10*b^4*c^7*d^5*f^2 - 192*B*C*a^
2*b^12*c^5*d^7*f^2 + 144*B*C*a^7*b^7*c^2*d^10*f^2 - 96*B*C*a^4*b^10*c^3*d^9
*f^2 + 88*B*C*a^12*b^2*c^5*d^7*f^2 - 72*B*C*a^11*b^3*c^8*d^4*f^2 - 48*B*C*a
^12*b^2*c^7*d^5*f^2 + 48*B*C*a^10*b^4*c^9*d^3*f^2 - 48*B*C*a^2*b^12*c^7*d^5
*f^2 - 48*B*C*a^2*b^12*c^3*d^9*f^2 - 12*B*C*a^9*b^5*c^10*d^2*f^2 + 4*B*C*a^
5*b^9*c^6*d^6*f^2 + 5824*A*C*a^5*b^9*c^7*d^5*f^2 - 4378*A*C*a^6*b^8*c^8*d^4
*f^2 + 4296*A*C*a^5*b^9*c^5*d^7*f^2 - 3912*A*C*a^6*b^8*c^6*d^6*f^2 - 3672*A
*C*a^9*b^5*c^5*d^7*f^2 + 3594*A*C*a^8*b^6*c^4*d^8*f^2 + 3236*A*C*a^8*b^6*c^
6*d^6*f^2 + 2816*A*C*a^5*b^9*c^9*d^3*f^2 + 2624*A*C*a^5*b^9*c^3*d^9*f^2 + 2
432*A*C*a^7*b^7*c^7*d^5*f^2 - 2366*A*C*a^4*b^10*c^8*d^4*f^2 + 2298*A*C*a^10
*b^4*c^4*d^8*f^2 + 1872*A*C*a^7*b^7*c^3*d^9*f^2 + 1848*A*C*a^10*b^4*c^6*d^6
*f^2 - 1644*A*C*a^4*b^10*c^6*d^6*f^2 - 1488*A*C*a^9*b^5*c^7*d^5*f^2 - 1408*
A*C*a^9*b^5*c^3*d^9*f^2 - 1308*A*C*a^6*b^8*c^4*d^8*f^2 + 1248*A*C*a^7*b^7*c
^5*d^7*f^2 - 1012*A*C*a^6*b^8*c^10*d^2*f^2 + 1008*A*C*a^3*b^11*c^7*d^5*f^2
+ 992*A*C*a^3*b^11*c^5*d^7*f^2 + 928*A*C*a^3*b^11*c^3*d^9*f^2 + 848*A*C*a^7
*b^7*c^9*d^3*f^2 + 636*A*C*a^8*b^6*c^2*d^10*f^2 - 628*A*C*a^4*b^10*c^10*d^2
*f^2 - 600*A*C*a^6*b^8*c^2*d^10*f^2 - 576*A*C*a^11*b^3*c^5*d^7*f^2 + 572*A*
C*a^10*b^4*c^2*d^10*f^2 + 464*A*C*a^8*b^6*c^8*d^4*f^2 - 304*A*C*a^4*b^10*c^
4*d^8*f^2 + 304*A*C*a^2*b^12*c^6*d^6*f^2 + 296*A*C*a^2*b^12*c^4*d^8*f^2 + 2
60*A*C*a^10*b^4*c^8*d^4*f^2 - 232*A*C*a^12*b^2*c^2*d^10*f^2 - 232*A*C*a^9*b
^5*c^9*d^3*f^2 + 228*A*C*a^2*b^12*c^10*d^2*f^2 - 188*A*C*a^4*b^10*c^2*d^10*
f^2 + 144*A*C*a^11*b^3*c^3*d^9*f^2 + 116*A*C*a^12*b^2*c^6*d^6*f^2 - 112*A*C
*a^11*b^3*c^7*d^5*f^2 + 112*A*C*a^3*b^11*c^9*d^3*f^2 + 92*A*C*a^8*b^6*c^10*
d^2*f^2 + 74*A*C*a^12*b^2*c^4*d^8*f^2 + 62*A*C*a^2*b^12*c^8*d^4*f^2 + 40*A*

C*a^2*b^12*c^2*d^10*f^2 - 7008*A*B*a^7*b^7*c^6*d^6*f^2 - 4032*A*B*a^7*b^7*c
^4*d^8*f^2 + 3952*A*B*a^8*b^6*c^7*d^5*f^2 + 3648*A*B*a^8*b^6*c^5*d^7*f^2 -
3392*A*B*a^7*b^7*c^8*d^4*f^2 + 3264*A*B*a^6*b^8*c^7*d^5*f^2 - 2992*A*B*a^4*
b^10*c^5*d^7*f^2 - 2368*A*B*a^4*b^10*c^7*d^5*f^2 - 2304*A*B*a^4*b^10*c^3*d^
9*f^2 - 1968*A*B*a^9*b^5*c^6*d^6*f^2 - 1872*A*B*a^4*b^10*c^9*d^3*f^2 - 1728
*A*B*a^7*b^7*c^2*d^10*f^2 + 1712*A*B*a^3*b^11*c^8*d^4*f^2 - 1536*A*B*a^10*b
^4*c^3*d^9*f^2 + 1536*A*B*a^6*b^8*c^5*d^7*f^2 - 1392*A*B*a^2*b^12*c^5*d^7*f
^2 + 1328*A*B*a^3*b^11*c^6*d^6*f^2 - 1104*A*B*a^2*b^12*c^3*d^9*f^2 - 1056*A
*B*a^6*b^8*c^3*d^9*f^2 + 976*A*B*a^6*b^8*c^9*d^3*f^2 + 960*A*B*a^11*b^3*c^4
*d^8*f^2 + 936*A*B*a^5*b^9*c^8*d^4*f^2 - 912*A*B*a^10*b^4*c^5*d^7*f^2 + 848
*A*B*a^8*b^6*c^9*d^3*f^2 + 816*A*B*a^3*b^11*c^4*d^8*f^2 - 816*A*B*a^2*b^12*
c^7*d^5*f^2 + 768*A*B*a^3*b^11*c^10*d^2*f^2 + 672*A*B*a^8*b^6*c^3*d^9*f^2 -
632*A*B*a^9*b^5*c^8*d^4*f^2 - 608*A*B*a^9*b^5*c^2*d^10*f^2 - 552*A*B*a^9*b
^5*c^4*d^8*f^2 - 544*A*B*a^7*b^7*c^10*d^2*f^2 - 480*A*B*a^5*b^9*c^2*d^10*f^
2 + 464*A*B*a^5*b^9*c^10*d^2*f^2 - 464*A*B*a^2*b^12*c^9*d^3*f^2 + 432*A*B*a
^11*b^3*c^2*d^10*f^2 - 368*A*B*a^12*b^2*c^3*d^9*f^2 - 256*A*B*a^5*b^9*c^6*d
^6*f^2 - 208*A*B*a^12*b^2*c^5*d^7*f^2 + 176*A*B*a^5*b^9*c^4*d^8*f^2 + 112*A
*B*a^11*b^3*c^6*d^6*f^2 + 112*A*B*a^10*b^4*c^7*d^5*f^2 - 16*A*B*a^3*b^11*c^
2*d^10*f^2 - 576*B*C*a^8*b^6*c*d^11*f^2 + 400*B*C*a^4*b^10*c^11*d*f^2 - 288
*B*C*a^6*b^8*c*d^11*f^2 - 176*B*C*a^6*b^8*c^11*d*f^2 + 128*B*C*a^10*b^4*c*d
^11*f^2 - 108*B*C*a*b^13*c^4*d^8*f^2 - 104*B*C*a^4*b^10*c*d^11*f^2 - 92*B*C
*a^13*b*c^4*d^8*f^2 - 60*B*C*a*b^13*c^8*d^4*f^2 - 60*B*C*a*b^13*c^6*d^6*f^2
+ 48*B*C*a^2*b^12*c^11*d*f^2 - 40*B*C*a*b^13*c^2*d^10*f^2 - 28*B*C*a^13*b*
c^2*d^10*f^2 - 24*B*C*a^12*b^2*c*d^11*f^2 + 20*B*C*a*b^13*c^10*d^2*f^2 - 16
*B*C*a^2*b^12*c*d^11*f^2 + 12*B*C*a^13*b*c^6*d^6*f^2 + 912*A*C*a^7*b^7*c*d^
11*f^2 + 808*A*C*a^5*b^9*c*d^11*f^2 + 432*A*C*a^5*b^9*c^11*d*f^2 + 336*A*C*
a^3*b^11*c*d^11*f^2 + 224*A*C*a^11*b^3*c*d^11*f^2 - 112*A*C*a^3*b^11*c^11*d
*f^2 + 112*A*C*a*b^13*c^3*d^9*f^2 - 88*A*C*a*b^13*c^9*d^3*f^2 + 80*A*C*a^13
*b*c^3*d^9*f^2 + 56*A*C*a*b^13*c^5*d^7*f^2 + 48*A*C*a^9*b^5*c*d^11*f^2 - 40
*A*C*a^13*b*c^5*d^7*f^2 - 16*A*C*a^7*b^7*c^11*d*f^2 + 16*A*C*a*b^13*c^7*d^5
*f^2 - 496*A*B*a^4*b^10*c*d^11*f^2 - 400*A*B*a^4*b^10*c^11*d*f^2 + 288*A*B*
a^8*b^6*c*d^11*f^2 - 288*A*B*a^6*b^8*c*d^11*f^2 - 272*A*B*a^2*b^12*c*d^11*f
^2 + 240*A*B*a*b^13*c^6*d^6*f^2 - 224*A*B*a^10*b^4*c*d^11*f^2 + 192*A*B*a*b
^13*c^8*d^4*f^2 + 192*A*B*a*b^13*c^4*d^8*f^2 + 176*A*B*a^6*b^8*c^11*d*f^2 +
104*A*B*a^13*b*c^4*d^8*f^2 - 48*A*B*a^2*b^12*c^11*d*f^2 + 16*A*B*a^13*b*c^
2*d^10*f^2 + 16*A*B*a*b^13*c^10*d^2*f^2 + 16*A*B*a*b^13*c^2*d^10*f^2 - 96*B
*C*b^14*c^7*d^5*f^2 - 72*B*C*b^14*c^5*d^7*f^2 - 24*B*C*b^14*c^9*d^3*f^2 - 1
6*B*C*b^14*c^3*d^9*f^2 + 116*A*C*b^14*c^6*d^6*f^2 + 100*A*C*b^14*c^4*d^8*f^
2 + 24*A*C*b^14*c^2*d^10*f^2 + 22*A*C*b^14*c^8*d^4*f^2 + 16*B*C*a^14*c^3*d^
9*f^2 + 8*A*C*b^14*c^10*d^2*f^2 - 192*A*B*b^14*c^5*d^7*f^2 - 176*A*B*b^14*c
^3*d^9*f^2 - 112*B*C*a^11*b^3*d^12*f^2 - 48*A*B*b^14*c^7*d^5*f^2 - 28*A*C*a
^14*c^2*d^10*f^2 + 4*B*C*a^5*b^9*d^12*f^2 + 2*A*C*a^14*c^4*d^8*f^2 + 150*A*
C*a^10*b^4*d^12*f^2 - 80*B*C*a^3*b^11*c^12*f^2 + 66*A*C*a^8*b^6*d^12*f^2 -
30*A*C*a^12*b^2*d^12*f^2 + 24*B*C*a^5*b^9*c^12*f^2 - 16*A*B*a^14*c^3*d^9*f^
2 - 12*A*C*a^4*b^10*d^12*f^2 - 576*A*B*a^7*b^7*d^12*f^2 - 432*A*B*a^9*b^5*d

^12*f^2 - 400*A*B*a^5*b^9*d^12*f^2 - 144*A*B*a^3*b^11*d^12*f^2 - 66*A*C*a^4
*b^10*c^12*f^2 + 54*A*C*a^2*b^12*c^12*f^2 - 32*A*B*a^11*b^3*d^12*f^2 + 2*A*
C*a^6*b^8*c^12*f^2 + 80*A*B*a^3*b^11*c^12*f^2 - 24*A*B*a^5*b^9*c^12*f^2 + 2
508*C^2*a^6*b^8*c^6*d^6*f^2 + 2376*C^2*a^9*b^5*c^5*d^7*f^2 + 2357*C^2*a^6*b
^8*c^8*d^4*f^2 - 2048*C^2*a^5*b^9*c^7*d^5*f^2 + 1304*C^2*a^9*b^5*c^3*d^9*f^
2 + 1303*C^2*a^4*b^10*c^8*d^4*f^2 + 1212*C^2*a^4*b^10*c^6*d^6*f^2 - 1203*C^
2*a^8*b^6*c^4*d^8*f^2 - 1192*C^2*a^5*b^9*c^9*d^3*f^2 + 1062*C^2*a^6*b^8*c^4
*d^8*f^2 + 984*C^2*a^9*b^5*c^7*d^5*f^2 - 952*C^2*a^8*b^6*c^6*d^6*f^2 + 768*
C^2*a^7*b^7*c^5*d^7*f^2 - 681*C^2*a^10*b^4*c^4*d^8*f^2 - 672*C^2*a^5*b^9*c^
5*d^7*f^2 - 480*C^2*a^10*b^4*c^6*d^6*f^2 + 458*C^2*a^6*b^8*c^10*d^2*f^2 - 4
48*C^2*a^7*b^7*c^7*d^5*f^2 + 422*C^2*a^4*b^10*c^4*d^8*f^2 + 372*C^2*a^6*b^8
*c^2*d^10*f^2 + 360*C^2*a^11*b^3*c^5*d^7*f^2 + 312*C^2*a^7*b^7*c^3*d^9*f^2
+ 278*C^2*a^4*b^10*c^10*d^2*f^2 - 232*C^2*a^7*b^7*c^9*d^3*f^2 + 194*C^2*a^1
2*b^2*c^2*d^10*f^2 + 176*C^2*a^9*b^5*c^9*d^3*f^2 + 152*C^2*a^3*b^11*c^5*d^7
*f^2 + 124*C^2*a^4*b^10*c^2*d^10*f^2 - 120*C^2*a^3*b^11*c^7*d^5*f^2 - 114*C
^2*a^2*b^12*c^10*d^2*f^2 - 102*C^2*a^8*b^6*c^2*d^10*f^2 + 101*C^2*a^12*b^2*
c^4*d^8*f^2 + 100*C^2*a^2*b^12*c^6*d^6*f^2 - 88*C^2*a^5*b^9*c^3*d^9*f^2 + 7
7*C^2*a^2*b^12*c^8*d^4*f^2 + 72*C^2*a^11*b^3*c^3*d^9*f^2 - 64*C^2*a^8*b^6*c
^10*d^2*f^2 + 64*C^2*a^3*b^11*c^3*d^9*f^2 - 58*C^2*a^10*b^4*c^2*d^10*f^2 +
56*C^2*a^12*b^2*c^6*d^6*f^2 + 56*C^2*a^11*b^3*c^7*d^5*f^2 + 40*C^2*a^3*b^11
*c^9*d^3*f^2 + 36*C^2*a^12*b^2*c^8*d^4*f^2 + 32*C^2*a^2*b^12*c^4*d^8*f^2 +
26*C^2*a^10*b^4*c^8*d^4*f^2 + 16*C^2*a^2*b^12*c^2*d^10*f^2 + 2*C^2*a^8*b^6*
c^8*d^4*f^2 + 2277*B^2*a^8*b^6*c^4*d^8*f^2 + 2144*B^2*a^5*b^9*c^7*d^5*f^2 -
2112*B^2*a^9*b^5*c^5*d^7*f^2 + 2028*B^2*a^8*b^6*c^6*d^6*f^2 - 1671*B^2*a^6
*b^8*c^8*d^4*f^2 + 1275*B^2*a^10*b^4*c^4*d^8*f^2 + 1176*B^2*a^5*b^9*c^5*d^7
*f^2 + 1096*B^2*a^5*b^9*c^9*d^3*f^2 - 1044*B^2*a^6*b^8*c^6*d^6*f^2 + 984*B^
2*a^10*b^4*c^6*d^6*f^2 - 968*B^2*a^9*b^5*c^3*d^9*f^2 - 888*B^2*a^9*b^5*c^7*
d^5*f^2 + 672*B^2*a^7*b^7*c^7*d^5*f^2 + 664*B^2*a^5*b^9*c^3*d^9*f^2 - 649*B
^2*a^4*b^10*c^8*d^4*f^2 + 618*B^2*a^8*b^6*c^2*d^10*f^2 + 514*B^2*a^4*b^10*c
^4*d^8*f^2 + 460*B^2*a^2*b^12*c^6*d^6*f^2 + 422*B^2*a^8*b^6*c^8*d^4*f^2 + 4
06*B^2*a^10*b^4*c^2*d^10*f^2 - 382*B^2*a^6*b^8*c^10*d^2*f^2 + 368*B^2*a^2*b
^12*c^4*d^8*f^2 - 312*B^2*a^11*b^3*c^5*d^7*f^2 + 312*B^2*a^7*b^7*c^3*d^9*f^
2 + 248*B^2*a^7*b^7*c^9*d^3*f^2 + 245*B^2*a^2*b^12*c^8*d^4*f^2 - 192*B^2*a^
7*b^7*c^5*d^7*f^2 - 184*B^2*a^3*b^11*c^9*d^3*f^2 + 182*B^2*a^2*b^12*c^10*d^
2*f^2 + 176*B^2*a^3*b^11*c^3*d^9*f^2 + 174*B^2*a^6*b^8*c^4*d^8*f^2 - 170*B^
2*a^4*b^10*c^10*d^2*f^2 - 152*B^2*a^9*b^5*c^9*d^3*f^2 + 152*B^2*a^4*b^10*c^
2*d^10*f^2 + 142*B^2*a^10*b^4*c^8*d^4*f^2 - 90*B^2*a^12*b^2*c^2*d^10*f^2 +
88*B^2*a^2*b^12*c^2*d^10*f^2 + 84*B^2*a^8*b^6*c^10*d^2*f^2 + 84*B^2*a^6*b^8
*c^2*d^10*f^2 + 60*B^2*a^12*b^2*c^6*d^6*f^2 - 56*B^2*a^11*b^3*c^7*d^5*f^2 +
53*B^2*a^12*b^2*c^4*d^8*f^2 + 24*B^2*a^11*b^3*c^3*d^9*f^2 + 24*B^2*a^4*b^1
0*c^6*d^6*f^2 + 24*B^2*a^3*b^11*c^7*d^5*f^2 - 8*B^2*a^3*b^11*c^5*d^7*f^2 +
4566*A^2*a^6*b^8*c^4*d^8*f^2 + 4284*A^2*a^6*b^8*c^6*d^6*f^2 - 3776*A^2*a^5*
b^9*c^7*d^5*f^2 - 3624*A^2*a^5*b^9*c^5*d^7*f^2 + 3122*A^2*a^4*b^10*c^4*d^8*
f^2 + 3108*A^2*a^6*b^8*c^2*d^10*f^2 + 2741*A^2*a^6*b^8*c^8*d^4*f^2 + 2592*A
^2*a^4*b^10*c^6*d^6*f^2 - 2536*A^2*a^5*b^9*c^3*d^9*f^2 + 2224*A^2*a^4*b^10*

c^2*d^10*f^2 - 2184*A^2*a^7*b^7*c^3*d^9*f^2 - 2016*A^2*a^7*b^7*c^5*d^7*f^2 - 1984*A^2*a^7*b^7*c^7*d^5*f^2 + 1626*A^2*a^8*b^6*c^2*d^10*f^2 - 1624*A^2*a^5*b^9*c^9*d^3*f^2 + 1603*A^2*a^4*b^10*c^8*d^4*f^2 + 1296*A^2*a^9*b^5*c^5*d^7*f^2 - 1144*A^2*a^3*b^11*c^5*d^7*f^2 - 992*A^2*a^3*b^11*c^3*d^9*f^2 + 968*A^2*a^2*b^12*c^4*d^8*f^2 - 888*A^2*a^3*b^11*c^7*d^5*f^2 + 849*A^2*a^8*b^6*c^4*d^8*f^2 + 808*A^2*a^2*b^12*c^2*d^10*f^2 - 616*A^2*a^7*b^7*c^9*d^3*f^2 + 554*A^2*a^6*b^8*c^10*d^2*f^2 - 504*A^2*a^10*b^4*c^6*d^6*f^2 + 504*A^2*a^9*b^5*c^7*d^5*f^2 + 460*A^2*a^2*b^12*c^6*d^6*f^2 + 350*A^2*a^10*b^4*c^2*d^10*f^2 + 350*A^2*a^4*b^10*c^10*d^2*f^2 - 321*A^2*a^10*b^4*c^4*d^8*f^2 + 216*A^2*a^11*b^3*c^5*d^7*f^2 - 216*A^2*a^11*b^3*c^3*d^9*f^2 + 182*A^2*a^12*b^2*c^2*d^10*f^2 - 152*A^2*a^3*b^11*c^9*d^3*f^2 - 124*A^2*a^8*b^6*c^6*d^6*f^2 - 114*A^2*a^2*b^12*c^10*d^2*f^2 + 104*A^2*a^9*b^5*c^3*d^9*f^2 + 77*A^2*a^2*b^12*c^8*d^4*f^2 + 74*A^2*a^8*b^6*c^8*d^4*f^2 - 70*A^2*a^10*b^4*c^8*d^4*f^2 + 56*A^2*a^11*b^3*c^7*d^5*f^2 + 56*A^2*a^9*b^5*c^9*d^3*f^2 + 41*A^2*a^12*b^2*c^4*d^8*f^2 - 28*A^2*a^12*b^2*c^6*d^6*f^2 - 28*A^2*a^8*b^6*c^10*d^2*f^2 - 16*B*C*b^14*c^11*d*f^2 - 16*B*C*a^14*c*d^11*f^2 - 48*A*B*b^14*c*d^11*f^2 + 16*A*B*b^14*c^11*d*f^2 + 12*B*C*a^13*b*d^12*f^2 + 24*B*C*a*b^13*c^12*f^2 + 16*A*B*a^14*c*d^11*f^2 - 24*A*B*a^13*b*d^12*f^2 - 24*A*B*a*b^13*d^12*f^2 - 24*A*B*a*b^13*c^12*f^2 + 216*C^2*a^9*b^5*c*d^11*f^2 - 216*C^2*a^5*b^9*c^11*d*f^2 + 56*C^2*a^3*b^11*c^11*d*f^2 + 56*C^2*a*b^13*c^9*d^3*f^2 + 56*C^2*a*b^13*c^5*d^7*f^2 - 40*C^2*a^11*b^3*c*d^11*f^2 + 40*C^2*a*b^13*c^7*d^5*f^2 + 32*C^2*a^13*b*c^5*d^7*f^2 - 24*C^2*a^7*b^7*c*d^11*f^2 - 16*C^2*a^13*b*c^3*d^9*f^2 + 16*C^2*a*b^13*c^3*d^9*f^2 + 8*C^2*a^7*b^7*c^11*d*f^2 - 8*C^2*a^5*b^9*c*d^11*f^2 + 264*B^2*a^7*b^7*c*d^11*f^2 + 224*B^2*a^5*b^9*c*d^11*f^2 + 168*B^2*a^5*b^9*c^11*d*f^2 - 112*B^2*a*b^13*c^9*d^3*f^2 - 104*B^2*a^3*b^11*c^11*d*f^2 - 104*B^2*a*b^13*c^7*d^5*f^2 + 96*B^2*a^3*b^11*c*d^11*f^2 + 88*B^2*a^11*b^3*c*d^11*f^2 - 72*B^2*a^9*b^5*c*d^11*f^2 - 64*B^2*a*b^13*c^5*d^7*f^2 + 32*B^2*a^13*b*c^3*d^9*f^2 - 24*B^2*a^13*b*c^5*d^7*f^2 - 24*B^2*a^7*b^7*c^11*d*f^2 + 16*B^2*a*b^13*c^3*d^9*f^2 - 888*A^2*a^7*b^7*c*d^11*f^2 - 800*A^2*a^5*b^9*c*d^11*f^2 - 336*A^2*a^3*b^11*c*d^11*f^2 - 264*A^2*a^9*b^5*c*d^11*f^2 - 216*A^2*a^5*b^9*c^11*d*f^2 - 184*A^2*a^11*b^3*c*d^11*f^2 - 128*A^2*a*b^13*c^3*d^9*f^2 - 112*A^2*a*b^13*c^5*d^7*f^2 - 64*A^2*a^13*b*c^3*d^9*f^2 + 56*A^2*a^3*b^11*c^11*d*f^2 - 56*A^2*a*b^13*c^7*d^5*f^2 + 32*A^2*a*b^13*c^9*d^3*f^2 + 8*A^2*a^13*b*c^5*d^7*f^2 + 8*A^2*a^7*b^7*c^11*d*f^2 + 24*C^2*a*b^13*c^11*d*f^2 - 16*C^2*a^13*b*c*d^11*f^2 - 40*B^2*a*b^13*c^11*d*f^2 + 24*B^2*a^13*b*c*d^11*f^2 + 16*B^2*a*b^13*c*d^11*f^2 - 48*A^2*a*b^13*c*d^11*f^2 - 40*A^2*a^13*b*c*d^11*f^2 + 24*A^2*a*b^13*c^11*d*f^2 - 6*A*C*b^14*c^12*f^2 + 2*A*C*a^14*d^12*f^2 + 31*C^2*b^14*c^8*d^4*f^2 + 20*C^2*b^14*c^6*d^6*f^2 + 4*C^2*b^14*c^4*d^8*f^2 + 2*C^2*b^14*c^10*d^2*f^2 + 80*B^2*b^14*c^6*d^6*f^2 + 64*B^2*b^14*c^4*d^8*f^2 + 31*B^2*b^14*c^8*d^4*f^2 + 16*B^2*b^14*c^2*d^10*f^2 + 14*C^2*a^14*c^2*d^10*f^2 + 14*B^2*b^14*c^10*d^2*f^2 - C^2*a^14*c^4*d^8*f^2 + 120*A^2*b^14*c^2*d^10*f^2 + 112*A^2*b^14*c^4*d^8*f^2 + 33*C^2*a^12*b^2*d^12*f^2 - 27*C^2*a^10*b^4*d^12*f^2 - 17*A^2*b^14*c^8*d^4*f^2 - 10*B^2*a^14*c^2*d^10*f^2 - 10*A^2*b^14*c^10*d^2*f^2 + 8*A^2*b^14*c^6*d^6*f^2 + 3*C^2*a^8*b^6*d^12*f^2 + 3*B^2*a^14*c^4*d^8*f^2 + 117*B^2*a^10*b^4*d^12*f^

2 + 111*B^2*a^8*b^6*d^12*f^2 + 72*B^2*a^6*b^8*d^12*f^2 + 33*C^2*a^4*b^10*c^
12*f^2 - 27*C^2*a^2*b^12*c^12*f^2 + 24*B^2*a^4*b^10*d^12*f^2 + 14*A^2*a^14*
c^2*d^10*f^2 + 4*B^2*a^2*b^12*d^12*f^2 - 3*B^2*a^12*b^2*d^12*f^2 - C^2*a^6*
b^8*c^12*f^2 - A^2*a^14*c^4*d^8*f^2 + 720*A^2*a^6*b^8*d^12*f^2 + 552*A^2*a^
4*b^10*d^12*f^2 + 471*A^2*a^8*b^6*d^12*f^2 + 216*A^2*a^2*b^12*d^12*f^2 + 93
*A^2*a^10*b^4*d^12*f^2 + 33*B^2*a^2*b^12*c^12*f^2 + 33*A^2*a^12*b^2*d^12*f^
2 - 27*B^2*a^4*b^10*c^12*f^2 + 3*B^2*a^6*b^8*c^12*f^2 + 33*A^2*a^4*b^10*c^1
2*f^2 - 27*A^2*a^2*b^12*c^12*f^2 - A^2*a^6*b^8*c^12*f^2 + 3*C^2*b^14*c^12*f
^2 - C^2*a^14*d^12*f^2 + 36*A^2*b^14*d^12*f^2 + 3*B^2*a^14*d^12*f^2 - B^2*b
^14*c^12*f^2 + 3*A^2*b^14*c^12*f^2 - A^2*a^14*d^12*f^2 - 44*A*B*C*a^10*b*c*
d^9*f + 3816*A*B*C*a^4*b^7*c^5*d^5*f + 2920*A*B*C*a^5*b^6*c^2*d^8*f - 2736*
A*B*C*a^6*b^5*c^3*d^7*f - 2672*A*B*C*a^3*b^8*c^4*d^6*f + 1996*A*B*C*a^7*b^4
*c^4*d^6*f - 1412*A*B*C*a^5*b^6*c^6*d^4*f + 1120*A*B*C*a^2*b^9*c^3*d^7*f +
1080*A*B*C*a^7*b^4*c^2*d^8*f + 1040*A*B*C*a^2*b^9*c^5*d^5*f + 684*A*B*C*a^5
*b^6*c^4*d^6*f + 592*A*B*C*a^4*b^7*c^3*d^7*f - 560*A*B*C*a^2*b^9*c^7*d^3*f
- 448*A*B*C*a^3*b^8*c^2*d^8*f - 400*A*B*C*a^8*b^3*c^5*d^5*f - 398*A*B*C*a^9
*b^2*c^2*d^8*f - 312*A*B*C*a^3*b^8*c^6*d^4*f + 166*A*B*C*a^3*b^8*c^8*d^2*f
+ 136*A*B*C*a^6*b^5*c^5*d^5*f + 128*A*B*C*a^6*b^5*c^7*d^3*f - 100*A*B*C*a^7
*b^4*c^6*d^4*f - 64*A*B*C*a^9*b^2*c^4*d^6*f + 64*A*B*C*a^4*b^7*c^7*d^3*f -
32*A*B*C*a^8*b^3*c^3*d^7*f - 16*A*B*C*a^5*b^6*c^8*d^2*f - 1312*A*B*C*a^4*b^
7*c*d^9*f + 996*A*B*C*a^8*b^3*c*d^9*f + 728*A*B*C*a*b^10*c^6*d^4*f - 624*A*
B*C*a^6*b^5*c*d^9*f - 584*A*B*C*a*b^10*c^2*d^8*f - 512*A*B*C*a*b^10*c^4*d^6
*f - 320*A*B*C*a^2*b^9*c*d^9*f - 98*A*B*C*a*b^10*c^8*d^2*f + 36*A*B*C*a^2*b
^9*c^9*d*f + 32*A*B*C*a^10*b*c^3*d^7*f - 16*A*B*C*a^4*b^7*c^9*d*f + 46*B*C^
2*a^10*b*c*d^9*f - 16*B^2*C*a*b^10*c*d^9*f - 2*B^2*C*a*b^10*c^9*d*f + 312*A
^2*C*a*b^10*c*d^9*f - 48*A*C^2*a*b^10*c*d^9*f - 6*A^2*C*a*b^10*c^9*d*f + 6*
A*C^2*a*b^10*c^9*d*f + 208*A*B^2*a*b^10*c*d^9*f - 2*A^2*B*a^10*b*c*d^9*f +
2*A*B^2*a*b^10*c^9*d*f - 224*A*B*C*b^11*c^5*d^5*f + 80*A*B*C*b^11*c^7*d^3*f
 - 32*A*B*C*b^11*c^3*d^7*f + 2*A*B*C*a^11*c^2*d^8*f - 480*A*B*C*a^7*b^4*d^1
0*f + 78*A*B*C*a^9*b^2*d^10*f - 64*A*B*C*a^5*b^6*d^10*f + 2*A*B*C*a^3*b^8*c
^10*f - 1692*B*C^2*a^4*b^7*c^5*d^5*f - 1500*B^2*C*a^5*b^6*c^5*d^5*f - 1464*
B^2*C*a^5*b^6*c^3*d^7*f + 1426*B*C^2*a^5*b^6*c^6*d^4*f - 1158*B^2*C*a^4*b^7
*c^6*d^4*f + 1152*B*C^2*a^6*b^5*c^3*d^7*f + 1026*B^2*C*a^6*b^5*c^4*d^6*f -
974*B*C^2*a^7*b^4*c^4*d^6*f + 960*B^2*C*a^3*b^8*c^5*d^5*f - 884*B*C^2*a^5*b
^6*c^2*d^8*f - 764*B^2*C*a^7*b^4*c^5*d^5*f + 752*B^2*C*a^4*b^7*c^2*d^8*f -
752*B*C^2*a^4*b^7*c^3*d^7*f + 738*B^2*C*a^4*b^7*c^4*d^6*f - 688*B^2*C*a^2*b
^9*c^6*d^4*f - 675*B^2*C*a^8*b^3*c^2*d^8*f + 560*B*C^2*a^8*b^3*c^5*d^5*f +
496*B*C^2*a^3*b^8*c^4*d^6*f + 496*B*C^2*a^2*b^9*c^7*d^3*f - 468*B*C^2*a^7*b
^4*c^2*d^8*f + 456*B^2*C*a^3*b^8*c^7*d^3*f - 452*B^2*C*a^8*b^3*c^4*d^6*f -
416*B*C^2*a^2*b^9*c^3*d^7*f + 378*B*C^2*a^5*b^6*c^4*d^6*f + 376*B*C^2*a^8*b
^3*c^3*d^7*f - 360*B^2*C*a^6*b^5*c^2*d^8*f + 355*B*C^2*a^9*b^2*c^2*d^8*f +
346*B^2*C*a^6*b^5*c^6*d^4*f - 320*B^2*C*a^2*b^9*c^4*d^6*f + 268*B^2*C*a^2*b
^9*c^2*d^8*f + 216*B^2*C*a^7*b^4*c^3*d^7*f - 203*B*C^2*a^3*b^8*c^8*d^2*f -
184*B*C^2*a^6*b^5*c^7*d^3*f + 170*B*C^2*a^7*b^4*c^6*d^4*f + 160*B^2*C*a^5*b
^6*c^7*d^3*f - 160*B*C^2*a^2*b^9*c^5*d^5*f - 140*B^2*C*a^4*b^7*c^8*d^2*f -

136*B*C^2*a^3*b^8*c^2*d^8*f + 112*B^2*C*a^9*b^2*c^3*d^7*f + 91*B^2*C*a^2*b^
9*c^8*d^2*f + 88*B*C^2*a^4*b^7*c^7*d^3*f + 72*B^2*C*a^8*b^3*c^6*d^4*f - 64*
B^2*C*a^3*b^8*c^3*d^7*f - 60*B*C^2*a^3*b^8*c^6*d^4*f + 56*B*C^2*a^9*b^2*c^4
*d^6*f + 52*B*C^2*a^6*b^5*c^5*d^5*f + 48*B^2*C*a^9*b^2*c^5*d^5*f - 48*B^2*C
*a^7*b^4*c^7*d^3*f + 44*B*C^2*a^5*b^6*c^8*d^2*f - 36*B*C^2*a^9*b^2*c^6*d^4*
f + 12*B^2*C*a^6*b^5*c^8*d^2*f - 2958*A^2*C*a^4*b^7*c^4*d^6*f - 1932*A^2*C*
a^4*b^7*c^2*d^8*f + 1848*A^2*C*a^5*b^6*c^3*d^7*f + 1728*A^2*C*a^3*b^8*c^3*d
^7*f + 1524*A^2*C*a^5*b^6*c^5*d^5*f + 1374*A*C^2*a^4*b^7*c^4*d^6*f - 1272*A
*C^2*a^5*b^6*c^3*d^7*f - 1236*A*C^2*a^5*b^6*c^5*d^5*f + 1116*A*C^2*a^4*b^7*
c^2*d^8*f - 1110*A^2*C*a^6*b^5*c^4*d^6*f + 1038*A*C^2*a^6*b^5*c^4*d^6*f - 7
68*A^2*C*a^2*b^9*c^2*d^8*f - 696*A^2*C*a^7*b^4*c^3*d^7*f - 666*A*C^2*a^4*b^
7*c^6*d^4*f + 564*A^2*C*a^6*b^5*c^2*d^8*f - 564*A*C^2*a^7*b^4*c^5*d^5*f - 5
55*A*C^2*a^8*b^3*c^2*d^8*f + 519*A^2*C*a^8*b^3*c^2*d^8*f - 480*A*C^2*a^3*b^
8*c^3*d^7*f + 456*A*C^2*a^3*b^8*c^5*d^5*f - 420*A*C^2*a^2*b^9*c^6*d^4*f + 4
08*A*C^2*a^7*b^4*c^3*d^7*f + 408*A*C^2*a^2*b^9*c^2*d^8*f + 348*A^2*C*a^2*b^
9*c^6*d^4*f - 348*A*C^2*a^6*b^5*c^2*d^8*f + 342*A*C^2*a^6*b^5*c^6*d^4*f - 3
36*A*C^2*a^8*b^3*c^4*d^6*f + 324*A^2*C*a^7*b^4*c^5*d^5*f - 312*A^2*C*a^2*b^
9*c^4*d^6*f + 264*A^2*C*a^8*b^3*c^4*d^6*f + 240*A*C^2*a^5*b^6*c^7*d^3*f + 1
95*A*C^2*a^2*b^9*c^8*d^2*f - 174*A^2*C*a^6*b^5*c^6*d^4*f + 144*A*C^2*a^9*b^
2*c^3*d^7*f - 123*A^2*C*a^2*b^9*c^8*d^2*f + 120*A*C^2*a^3*b^8*c^7*d^3*f + 1
08*A*C^2*a^8*b^3*c^6*d^4*f - 102*A^2*C*a^4*b^7*c^6*d^4*f - 96*A^2*C*a^4*b^7
*c^8*d^2*f + 72*A^2*C*a^3*b^8*c^7*d^3*f + 72*A*C^2*a^9*b^2*c^5*d^5*f - 48*A
^2*C*a^9*b^2*c^3*d^7*f + 48*A^2*C*a^5*b^6*c^7*d^3*f - 48*A*C^2*a^2*b^9*c^4*
d^6*f - 24*A^2*C*a^3*b^8*c^5*d^5*f - 12*A*C^2*a^4*b^7*c^8*d^2*f + 2736*A^2*
B*a^6*b^5*c^3*d^7*f + 2464*A^2*B*a^3*b^8*c^4*d^6*f - 2298*A*B^2*a^4*b^7*c^4
*d^6*f - 2252*A^2*B*a^5*b^6*c^2*d^8*f - 1692*A^2*B*a^4*b^7*c^5*d^5*f - 1592
*A*B^2*a^4*b^7*c^2*d^8*f - 1338*A*B^2*a^6*b^5*c^4*d^6*f + 1320*A*B^2*a^5*b^
6*c^3*d^7*f + 1212*A*B^2*a^5*b^6*c^5*d^5*f - 1056*A*B^2*a^3*b^8*c^5*d^5*f +
1024*A^2*B*a^4*b^7*c^3*d^7*f - 1022*A^2*B*a^7*b^4*c^4*d^6*f - 880*A^2*B*a^
2*b^9*c^5*d^5*f - 846*A^2*B*a^5*b^6*c^4*d^6*f - 840*A*B^2*a^7*b^4*c^3*d^7*f
+ 760*A*B^2*a^2*b^9*c^6*d^4*f - 704*A^2*B*a^2*b^9*c^3*d^7*f + 688*A*B^2*a^
3*b^8*c^3*d^7*f + 660*A^2*B*a^3*b^8*c^6*d^4*f - 612*A^2*B*a^7*b^4*c^2*d^8*f
+ 462*A*B^2*a^4*b^7*c^6*d^4*f + 459*A*B^2*a^8*b^3*c^2*d^8*f - 412*A*B^2*a^
2*b^9*c^2*d^8*f - 408*A*B^2*a^3*b^8*c^7*d^3*f + 388*A^2*B*a^6*b^5*c^5*d^5*f
+ 296*A^2*B*a^3*b^8*c^2*d^8*f + 288*A*B^2*a^6*b^5*c^2*d^8*f + 284*A*B^2*a^
7*b^4*c^5*d^5*f + 236*A*B^2*a^8*b^3*c^4*d^6*f - 226*A*B^2*a^6*b^5*c^6*d^4*f
+ 212*A*B^2*a^2*b^9*c^4*d^6*f + 202*A^2*B*a^5*b^6*c^6*d^4*f - 152*A^2*B*a^
4*b^7*c^7*d^3*f + 88*A^2*B*a^8*b^3*c^3*d^7*f + 79*A^2*B*a^9*b^2*c^2*d^8*f -
70*A^2*B*a^7*b^4*c^6*d^4*f + 68*A*B^2*a^4*b^7*c^8*d^2*f + 64*A^2*B*a^2*b^9
*c^7*d^3*f - 64*A*B^2*a^9*b^2*c^3*d^7*f + 56*A^2*B*a^8*b^3*c^5*d^5*f + 56*A
^2*B*a^6*b^5*c^7*d^3*f + 37*A^2*B*a^3*b^8*c^8*d^2*f - 28*A^2*B*a^9*b^2*c^4*
d^6*f - 28*A^2*B*a^5*b^6*c^8*d^2*f + 17*A*B^2*a^2*b^9*c^8*d^2*f - 16*A*B^2*
a^5*b^6*c^7*d^3*f + 48*A*B*C*b^11*c*d^9*f + 4*A*B*C*b^11*c^9*d*f + 24*A*B*C
*a*b^10*d^10*f - 6*A*B*C*a*b^10*c^10*f + 432*B^2*C*a^7*b^4*c*d^9*f - 376*B*
C^2*a*b^10*c^6*d^4*f - 354*B*C^2*a^8*b^3*c*d^9*f + 352*B^2*C*a*b^10*c^5*d^5

*f + 320*B^2*C*a^5*b^6*c*d^9*f + 256*B^2*C*a*b^10*c^3*d^7*f - 232*B^2*C*a*b
^10*c^7*d^3*f - 210*B^2*C*a^9*b^2*c*d^9*f - 152*B*C^2*a*b^10*c^4*d^6*f + 85
*B*C^2*a*b^10*c^8*d^2*f + 72*B^2*C*a^3*b^8*c*d^9*f - 48*B*C^2*a^6*b^5*c*d^9
*f - 40*B*C^2*a^10*b*c^3*d^7*f + 40*B*C^2*a*b^10*c^2*d^8*f + 37*B^2*C*a^10*
b*c^2*d^8*f + 22*B^2*C*a^3*b^8*c^9*d*f - 18*B*C^2*a^2*b^9*c^9*d*f + 16*B*C^
2*a^2*b^9*c*d^9*f - 12*B^2*C*a^10*b*c^4*d^6*f + 8*B*C^2*a^4*b^7*c^9*d*f + 8
*B*C^2*a^4*b^7*c*d^9*f - 984*A^2*C*a^7*b^4*c*d^9*f + 672*A^2*C*a^3*b^8*c*d^
9*f + 552*A*C^2*a^7*b^4*c*d^9*f - 504*A^2*C*a*b^10*c^5*d^5*f - 408*A^2*C*a^
5*b^6*c*d^9*f + 408*A*C^2*a^5*b^6*c*d^9*f + 336*A*C^2*a*b^10*c^5*d^5*f - 21
6*A*C^2*a*b^10*c^7*d^3*f + 192*A*C^2*a*b^10*c^3*d^7*f - 162*A*C^2*a^9*b^2*c
*d^9*f + 120*A^2*C*a*b^10*c^7*d^3*f + 96*A^2*C*a*b^10*c^3*d^7*f + 90*A^2*C*
a^9*b^2*c*d^9*f + 66*A^2*C*a^3*b^8*c^9*d*f - 66*A*C^2*a^3*b^8*c^9*d*f + 57*
A*C^2*a^10*b*c^2*d^8*f - 48*A*C^2*a^3*b^8*c*d^9*f - 9*A^2*C*a^10*b*c^2*d^8*
f + 1736*A^2*B*a^4*b^7*c*d^9*f + 1248*A^2*B*a^6*b^5*c*d^9*f - 1008*A*B^2*a^
7*b^4*c*d^9*f + 772*A^2*B*a*b^10*c^4*d^6*f - 688*A*B^2*a*b^10*c^5*d^5*f - 6
08*A*B^2*a^5*b^6*c*d^9*f + 436*A^2*B*a*b^10*c^2*d^8*f - 426*A^2*B*a^8*b^3*c
*d^9*f + 312*A*B^2*a^3*b^8*c*d^9*f + 304*A^2*B*a^2*b^9*c*d^9*f - 244*A^2*B*
a*b^10*c^6*d^4*f - 160*A*B^2*a*b^10*c^3*d^7*f + 114*A*B^2*a^9*b^2*c*d^9*f +
 88*A*B^2*a*b^10*c^7*d^3*f - 22*A*B^2*a^3*b^8*c^9*d*f - 18*A^2*B*a^2*b^9*c^
9*d*f + 13*A^2*B*a*b^10*c^8*d^2*f - 13*A*B^2*a^10*b*c^2*d^8*f + 8*A^2*B*a^1
0*b*c^3*d^7*f + 8*A^2*B*a^4*b^7*c^9*d*f + 112*B^2*C*b^11*c^6*d^4*f - 64*B*C
^2*b^11*c^7*d^3*f + 16*B^2*C*b^11*c^4*d^6*f - 16*B^2*C*b^11*c^2*d^8*f + 16*
B*C^2*b^11*c^5*d^5*f + 16*B*C^2*b^11*c^3*d^7*f - B^2*C*b^11*c^8*d^2*f + 96*
A^2*C*b^11*c^4*d^6*f - 84*A^2*C*b^11*c^6*d^4*f + 72*A*C^2*b^11*c^6*d^4*f -
24*A*C^2*b^11*c^4*d^6*f - 24*A*C^2*b^11*c^2*d^8*f - 21*A*C^2*b^11*c^8*d^2*f
 + 12*A^2*C*b^11*c^2*d^8*f + 9*A^2*C*b^11*c^8*d^2*f - B*C^2*a^11*c^2*d^8*f
+ 176*A*B^2*b^11*c^4*d^6*f + 136*A^2*B*b^11*c^5*d^5*f - 128*A^2*B*b^11*c^3*
d^7*f + 112*A*B^2*b^11*c^2*d^8*f + 111*B^2*C*a^8*b^3*d^10*f - 64*A*B^2*b^11
*c^6*d^4*f - 39*B*C^2*a^9*b^2*d^10*f + 24*B*C^2*a^7*b^4*d^10*f - 16*A^2*B*b
^11*c^7*d^3*f - 4*B^2*C*a^2*b^9*d^10*f - 4*B*C^2*a^5*b^6*d^10*f + 432*A^2*C
*a^6*b^5*d^10*f + 192*A^2*C*a^4*b^7*d^10*f - 111*A^2*C*a^8*b^3*d^10*f + 111
*A*C^2*a^8*b^3*d^10*f - 72*A*C^2*a^6*b^5*d^10*f + 12*A*C^2*a^4*b^7*d^10*f -
 3*B^2*C*a^2*b^9*c^10*f - A^2*B*a^11*c^2*d^8*f - B*C^2*a^3*b^8*c^10*f + 456
*A^2*B*a^7*b^4*d^10*f - 288*A^2*B*a^3*b^8*d^10*f + 252*A*B^2*a^6*b^5*d^10*f
 + 192*A*B^2*a^4*b^7*d^10*f - 183*A*B^2*a^8*b^3*d^10*f - 148*A^2*B*a^5*b^6*
d^10*f + 76*A*B^2*a^2*b^9*d^10*f - 9*A^2*C*a^2*b^9*c^10*f + 9*A*C^2*a^2*b^9
*c^10*f - 3*A^2*B*a^9*b^2*d^10*f + 3*A*B^2*a^2*b^9*c^10*f - A^2*B*a^3*b^8*c
^10*f - 2*C^3*a*b^10*c^9*d*f - 2*B^3*a^10*b*c*d^9*f - 264*A^3*a*b^10*c*d^9*
f + 2*A^3*a*b^10*c^9*d*f - 2*B*C^2*b^11*c^9*d*f - 2*B^2*C*a^11*c*d^9*f - 12
0*A^2*B*b^11*c*d^9*f - 9*B^2*C*a^10*b*d^10*f - 6*A^2*C*a^11*c*d^9*f + 6*A*C
^2*a^11*c*d^9*f - 2*A^2*B*b^11*c^9*d*f + 9*A^2*C*a^10*b*d^10*f - 9*A*C^2*a^
10*b*d^10*f + 3*B*C^2*a*b^10*c^10*f + 2*A*B^2*a^11*c*d^9*f - 132*A^2*B*a*b^
10*d^10*f - 3*A*B^2*a^10*b*d^10*f + 3*A^2*B*a*b^10*c^10*f + 520*C^3*a^5*b^6
*c^3*d^7*f + 460*C^3*a^5*b^6*c^5*d^5*f - 418*C^3*a^6*b^5*c^4*d^6*f + 406*C^
3*a^4*b^7*c^6*d^4*f + 268*C^3*a^7*b^4*c^5*d^5*f - 266*C^3*a^6*b^5*c^6*d^4*f

+ 233*C^3*a^8*b^3*c^2*d^8*f - 176*C^3*a^5*b^6*c^7*d^3*f + 164*C^3*a^2*b^9*
c^6*d^4*f + 140*C^3*a^6*b^5*c^2*d^8*f + 136*C^3*a^2*b^9*c^4*d^6*f - 128*C^3
*a^9*b^2*c^3*d^7*f + 128*C^3*a^3*b^8*c^3*d^7*f - 108*C^3*a^8*b^3*c^6*d^4*f
- 104*C^3*a^3*b^8*c^7*d^3*f - 104*C^3*a^3*b^8*c^5*d^5*f + 100*C^3*a^8*b^3*c
^4*d^6*f - 89*C^3*a^2*b^9*c^8*d^2*f - 72*C^3*a^9*b^2*c^5*d^5*f - 40*C^3*a^7
*b^4*c^3*d^7*f + 40*C^3*a^4*b^7*c^8*d^2*f - 28*C^3*a^4*b^7*c^2*d^8*f - 16*C
^3*a^2*b^9*c^2*d^8*f - 2*C^3*a^4*b^7*c^4*d^6*f + 828*B^3*a^4*b^7*c^5*d^5*f
+ 408*B^3*a^5*b^6*c^2*d^8*f + 390*B^3*a^7*b^4*c^4*d^6*f - 372*B^3*a^3*b^8*c
^4*d^6*f - 336*B^3*a^6*b^5*c^3*d^7*f - 314*B^3*a^5*b^6*c^6*d^4*f + 288*B^3*
a^4*b^7*c^3*d^7*f + 216*B^3*a^7*b^4*c^2*d^8*f - 176*B^3*a^2*b^9*c^7*d^3*f +
128*B^3*a^2*b^9*c^3*d^7*f + 108*B^3*a^6*b^5*c^5*d^5*f + 88*B^3*a^4*b^7*c^7
*d^3*f + 72*B^3*a^2*b^9*c^5*d^5*f - 68*B^3*a^3*b^8*c^2*d^8*f - 65*B^3*a^9*b
^2*c^2*d^8*f - 56*B^3*a^8*b^3*c^5*d^5*f + 40*B^3*a^6*b^5*c^7*d^3*f + 37*B^3
*a^3*b^8*c^8*d^2*f + 30*B^3*a^5*b^6*c^4*d^6*f - 28*B^3*a^5*b^6*c^8*d^2*f +
24*B^3*a^8*b^3*c^3*d^7*f - 4*B^3*a^9*b^2*c^4*d^6*f - 2*B^3*a^7*b^4*c^6*d^4*
f + 1586*A^3*a^4*b^7*c^4*d^6*f - 1376*A^3*a^3*b^8*c^3*d^7*f - 1096*A^3*a^5*
b^6*c^3*d^7*f + 844*A^3*a^4*b^7*c^2*d^8*f - 748*A^3*a^5*b^6*c^5*d^5*f + 490
*A^3*a^6*b^5*c^4*d^6*f + 376*A^3*a^2*b^9*c^2*d^8*f + 362*A^3*a^4*b^7*c^6*d^
4*f - 356*A^3*a^6*b^5*c^2*d^8*f + 328*A^3*a^7*b^4*c^3*d^7*f - 328*A^3*a^3*b
^8*c^5*d^5*f + 224*A^3*a^2*b^9*c^4*d^6*f - 197*A^3*a^8*b^3*c^2*d^8*f - 112*
A^3*a^5*b^6*c^7*d^3*f + 98*A^3*a^6*b^5*c^6*d^4*f - 92*A^3*a^2*b^9*c^6*d^4*f
- 88*A^3*a^3*b^8*c^7*d^3*f + 68*A^3*a^4*b^7*c^8*d^2*f + 32*A^3*a^9*b^2*c^3
*d^7*f - 28*A^3*a^8*b^3*c^4*d^6*f - 28*A^3*a^7*b^4*c^5*d^5*f + 17*A^3*a^2*b
^9*c^8*d^2*f + 104*C^3*a*b^10*c^7*d^3*f + 54*C^3*a^9*b^2*c*d^9*f - 40*C^3*a
^7*b^4*c*d^9*f - 35*C^3*a^10*b*c^2*d^8*f + 22*C^3*a^3*b^8*c^9*d*f + 16*C^3*
a*b^10*c^5*d^5*f - 16*C^3*a*b^10*c^3*d^7*f + 8*C^3*a^5*b^6*c*d^9*f - 2*A*B*
C*a^11*d^10*f + 198*B^3*a^8*b^3*c*d^9*f + 192*B^3*a*b^10*c^6*d^4*f - 128*B^
3*a^4*b^7*c*d^9*f - 80*B^3*a*b^10*c^2*d^8*f - 56*B^3*a^2*b^9*c*d^9*f - 24*B
^3*a^6*b^5*c*d^9*f - 18*B^3*a^2*b^9*c^9*d*f - 16*B^3*a*b^10*c^4*d^6*f + 13*
B^3*a*b^10*c^8*d^2*f + 8*B^3*a^10*b*c^3*d^7*f + 8*B^3*a^4*b^7*c^9*d*f - 624
*A^3*a^3*b^8*c*d^9*f + 472*A^3*a^7*b^4*c*d^9*f - 272*A^3*a*b^10*c^3*d^7*f +
152*A^3*a*b^10*c^5*d^5*f - 22*A^3*a^3*b^8*c^9*d*f + 18*A^3*a^9*b^2*c*d^9*f
- 13*A^3*a^10*b*c^2*d^8*f - 8*A^3*a^5*b^6*c*d^9*f - 8*A^3*a*b^10*c^7*d^3*f
+ A*B^2*b^11*c^8*d^2*f + 11*C^3*b^11*c^8*d^2*f - 8*C^3*b^11*c^6*d^4*f - 4*
C^3*b^11*c^4*d^6*f - 64*B^3*b^11*c^5*d^5*f - 32*B^3*b^11*c^3*d^7*f - 68*A^3
*b^11*c^4*d^6*f + 20*A^3*b^11*c^6*d^4*f + 12*A^3*b^11*c^2*d^8*f - C^3*a^8*b
^3*d^10*f - B^3*a^11*c^2*d^8*f - 60*B^3*a^7*b^4*d^10*f - 32*B^3*a^5*b^6*d^1
0*f + 21*B^3*a^9*b^2*d^10*f - 12*B^3*a^3*b^8*d^10*f - 3*C^3*a^2*b^9*c^10*f
- 360*A^3*a^6*b^5*d^10*f - 204*A^3*a^4*b^7*d^10*f - B^3*a^3*b^8*c^10*f + 3*
A^3*a^2*b^9*c^10*f - 2*C^3*a^11*c*d^9*f - 2*B^3*b^11*c^9*d*f + 3*C^3*a^10*b
*d^10*f + 2*A^3*a^11*c*d^9*f + 3*B^3*a*b^10*c^10*f - 3*A^3*a^10*b*d^10*f -
36*A^2*C*b^11*d^10*f + 3*A^2*C*b^11*c^10*f - 3*A*C^2*b^11*c^10*f - A*B^2*b^
11*c^10*f + 36*A^3*b^11*d^10*f - A^3*b^11*c^10*f + A^3*b^11*c^8*d^2*f + A^3
*a^8*b^3*d^10*f + B^2*C*b^11*c^10*f + B*C^2*a^11*d^10*f + A^2*B*a^11*d^10*f
+ C^3*b^11*c^10*f + B^3*a^11*d^10*f - 6*A*B^2*C*a^7*b*c*d^7 + 4*A*B^2*C*a*

b^7*c*d^7 + 168*A^2*B*C*a^2*b^6*c^3*d^5 + 144*A*B*C^2*a^3*b^5*c^4*d^4 - 129
*A^2*B*C*a^3*b^5*c^4*d^4 - 96*A*B*C^2*a^2*b^6*c^3*d^5 + 84*A*B*C^2*a^3*b^5*
c^2*d^6 + 72*A^2*B*C*a^4*b^4*c^3*d^5 - 72*A^2*B*C*a^3*b^5*c^2*d^6 + 64*A*B^
2*C*a^4*b^4*c^4*d^4 - 60*A*B*C^2*a^4*b^4*c^3*d^5 + 57*A^2*B*C*a^5*b^3*c^2*d
^6 - 56*A*B^2*C*a^5*b^3*c^3*d^5 - 39*A*B^2*C*a^2*b^6*c^4*d^4 - 38*A*B^2*C*a
^3*b^5*c^5*d^3 + 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^5*b^3*c^4*d^4 -
30*A*B*C^2*a^5*b^3*c^2*d^6 + 27*A*B^2*C*a^6*b^2*c^2*d^6 - 24*A*B^2*C*a^2*b^
6*c^2*d^6 + 24*A*B*C^2*a^6*b^2*c^3*d^5 - 24*A*B*C^2*a^4*b^4*c^5*d^3 - 18*A^
2*B*C*a^5*b^3*c^4*d^4 + 18*A^2*B*C*a^2*b^6*c^5*d^3 - 15*A*B^2*C*a^4*b^4*c^2
*d^6 - 12*A^2*B*C*a^6*b^2*c^3*d^5 + 12*A^2*B*C*a^4*b^4*c^5*d^3 + 9*A*B^2*C*
a^2*b^6*c^6*d^2 + 6*A*B*C^2*a^3*b^5*c^6*d^2 - 3*A^2*B*C*a^3*b^5*c^6*d^2 + 6
0*A^2*B*C*a^2*b^6*c*d^7 - 51*A^2*B*C*a*b^7*c^4*d^4 + 48*A*B*C^2*a^6*b^2*c*d
^7 - 42*A^2*B*C*a^6*b^2*c*d^7 - 42*A^2*B*C*a*b^7*c^2*d^6 + 36*A*B*C^2*a^4*b
^4*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4 + 36*A*B*C^2*a*b^7*c^2*d^6 - 30*A^2*B*C
*a^4*b^4*c*d^7 + 24*A*B^2*C*a^3*b^5*c*d^7 - 24*A*B*C^2*a^2*b^6*c*d^7 + 18*A
*B^2*C*a*b^7*c^5*d^3 - 18*A*B*C^2*a*b^7*c^6*d^2 + 12*A*B^2*C*a*b^7*c^3*d^5
+ 9*A^2*B*C*a*b^7*c^6*d^2 + 6*A*B^2*C*a^5*b^3*c*d^7 - 6*A*B*C^2*a^7*b*c^2*d
^6 + 3*A^2*B*C*a^7*b*c^2*d^6 - 18*B^3*C*a^6*b^2*c*d^7 - 18*B*C^3*a^6*b^2*c*
d^7 - 14*B^3*C*a^4*b^4*c*d^7 - 14*B*C^3*a^4*b^4*c*d^7 - 10*B^3*C*a*b^7*c^2*
d^6 - 10*B*C^3*a*b^7*c^2*d^6 + 9*B^3*C*a*b^7*c^6*d^2 + 9*B*C^3*a*b^7*c^6*d^
2 - 7*B^3*C*a*b^7*c^4*d^4 - 7*B*C^3*a*b^7*c^4*d^4 + 6*B^2*C^2*a^7*b*c*d^7 -
 4*B^3*C*a^2*b^6*c*d^7 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a^2*b^6*c*d^7 + 3*
B^3*C*a^7*b*c^2*d^6 + 3*B*C^3*a^7*b*c^2*d^6 + 144*A^3*C*a^3*b^5*c*d^7 + 62*
A^3*C*a^5*b^3*c*d^7 + 48*A*C^3*a^3*b^5*c*d^7 - 36*A^2*C^2*a*b^7*c*d^7 + 26*
A*C^3*a^5*b^3*c*d^7 + 20*A^3*C*a*b^7*c^3*d^5 + 18*A^2*C^2*a^7*b*c*d^7 - 18*
A*C^3*a*b^7*c^5*d^3 - 6*A^3*C*a*b^7*c^5*d^3 - 4*A*C^3*a*b^7*c^3*d^5 - 32*A^
3*B*a^2*b^6*c*d^7 - 32*A*B^3*a^2*b^6*c*d^7 + 22*A^3*B*a*b^7*c^4*d^4 + 22*A*
B^3*a*b^7*c^4*d^4 + 16*A^3*B*a*b^7*c^2*d^6 + 16*A*B^3*a*b^7*c^2*d^6 + 12*A^
3*B*a^6*b^2*c*d^7 + 12*A*B^3*a^6*b^2*c*d^7 + 8*A^3*B*a^4*b^4*c*d^7 - 8*A^2*
B^2*a*b^7*c*d^7 + 8*A*B^3*a^4*b^4*c*d^7 + 36*A^2*B*C*b^8*c^3*d^5 + 24*A*B*C
^2*b^8*c^5*d^3 - 18*A^2*B*C*b^8*c^5*d^3 - 12*A*B*C^2*b^8*c^3*d^5 - 3*A*B^2*
C*b^8*c^6*d^2 - 3*A*B^2*C*b^8*c^4*d^4 - 2*A*B^2*C*b^8*c^2*d^6 + 57*A^2*B*C*
a^5*b^3*d^8 + 36*A^2*B*C*a^3*b^5*d^8 - 30*A*B*C^2*a^5*b^3*d^8 - 18*A*B*C^2*
a^3*b^5*d^8 - 9*A*B^2*C*a^4*b^4*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 2*A*B^2*C*a^2
*b^6*d^8 + 34*B^2*C^2*a^3*b^5*c^5*d^3 + 28*B^2*C^2*a^5*b^3*c^3*d^5 + 24*B^2
*C^2*a^2*b^6*c^4*d^4 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*
d^5 + 12*B^2*C^2*a^2*b^6*c^2*d^6 + 9*B^2*C^2*a^6*b^2*c^4*d^4 + 9*B^2*C^2*a^
4*b^4*c^2*d^6 - 9*B^2*C^2*a^2*b^6*c^6*d^2 - 3*B^2*C^2*a^6*b^2*c^2*d^6 + 159
*A^2*C^2*a^4*b^4*c^2*d^6 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^3*b^5
*c^5*d^3 + 78*A^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2
*C^2*a^6*b^2*c^2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^2*b^6*c^4*
d^4 + 9*A^2*C^2*a^6*b^2*c^4*d^4 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a
^4*b^4*c^2*d^6 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^2*b^6*c^4*d^4 +
28*A^2*B^2*a^5*b^3*c^3*d^5 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^6*b^2
*c^2*d^6 + 4*A^2*B^2*a^3*b^5*c^5*d^3 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a^7*

b*c*d^7 + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a^7*b*c*d^7 + 24*A^2*B*C*b^8*c*d^7
 - 12*A*B*C^2*b^8*c*d^7 + 12*A^2*B*C*a*b^7*d^8 + 6*A*B*C^2*a^7*b*d^8 - 6*A*
B*C^2*a*b^7*d^8 - 3*A^2*B*C*a^7*b*d^8 - 53*B^3*C*a^3*b^5*c^4*d^4 - 53*B*C^3
*a^3*b^5*c^4*d^4 - 32*B^3*C*a^3*b^5*c^2*d^6 - 32*B*C^3*a^3*b^5*c^2*d^6 - 18
*B^3*C*a^5*b^3*c^4*d^4 - 18*B*C^3*a^5*b^3*c^4*d^4 + 16*B^3*C*a^4*b^4*c^3*d^
5 + 16*B*C^3*a^4*b^4*c^3*d^5 - 12*B^3*C*a^6*b^2*c^3*d^5 + 12*B^3*C*a^4*b^4*
c^5*d^3 + 12*B^2*C^2*a^3*b^5*c*d^7 - 12*B*C^3*a^6*b^2*c^3*d^5 + 12*B*C^3*a^
4*b^4*c^5*d^3 + 8*B^3*C*a^2*b^6*c^3*d^5 + 8*B*C^3*a^2*b^6*c^3*d^5 - 6*B^3*C
*a^2*b^6*c^5*d^3 + 6*B^2*C^2*a^5*b^3*c*d^7 - 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*
C^3*a^2*b^6*c^5*d^3 - 3*B^3*C*a^3*b^5*c^6*d^2 - 3*B*C^3*a^3*b^5*c^6*d^2 - 1
75*A^3*C*a^4*b^4*c^2*d^6 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a^3*b^5*
c*d^7 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^3*b^5*c^5*d^3 - 73*A*C^3*a^4
*b^4*c^2*d^6 - 66*A^2*C^2*a^5*b^3*c*d^7 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C
^3*a^4*b^4*c^4*d^4 + 30*A^3*C*a^4*b^4*c^4*d^4 - 30*A^3*C*a^3*b^5*c^5*d^3 +
27*A*C^3*a^2*b^6*c^6*d^2 + 21*A*C^3*a^2*b^6*c^4*d^4 + 18*A^2*C^2*a*b^7*c^5*
d^3 - 18*A*C^3*a^6*b^2*c^4*d^4 - 16*A*C^3*a^2*b^6*c^2*d^6 + 15*A^3*C*a^6*b^
2*c^2*d^6 - 15*A^3*C*a^2*b^6*c^4*d^4 - 12*A^2*C^2*a*b^7*c^3*d^5 + 9*A^3*C*a
^2*b^6*c^6*d^2 + 9*A*C^3*a^6*b^2*c^2*d^6 - 80*A^3*B*a^2*b^6*c^3*d^5 - 80*A*
B^3*a^2*b^6*c^3*d^5 + 38*A^3*B*a^3*b^5*c^4*d^4 + 38*A*B^3*a^3*b^5*c^4*d^4 -
 36*A^2*B^2*a^3*b^5*c*d^7 - 28*A^3*B*a^5*b^3*c^2*d^6 - 28*A^3*B*a^4*b^4*c^3
*d^5 - 28*A*B^3*a^5*b^3*c^2*d^6 - 28*A*B^3*a^4*b^4*c^3*d^5 + 20*A^3*B*a^3*b
^5*c^2*d^6 + 20*A*B^3*a^3*b^5*c^2*d^6 - 12*A^3*B*a^2*b^6*c^5*d^3 - 12*A^2*B
^2*a^5*b^3*c*d^7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A^2*B^2*a*b^7*c^3*d^5 - 12
*A*B^3*a^2*b^6*c^5*d^3 + 9*B^2*C^2*b^8*c^4*d^4 + 4*B^2*C^2*b^8*c^2*d^6 + 3*
B^2*C^2*b^8*c^6*d^2 - 30*A^2*C^2*b^8*c^4*d^4 + 9*A^2*C^2*b^8*c^6*d^2 + 16*A
^2*B^2*b^8*c^2*d^6 + 6*B^2*C^2*a^6*b^2*d^8 + 3*B^2*C^2*a^4*b^4*d^8 + 3*A^2*
B^2*b^8*c^4*d^4 + 36*A^2*C^2*a^4*b^4*d^8 + 27*A^2*C^2*a^2*b^6*d^8 - 18*A^2*
C^2*a^6*b^2*d^8 + 33*A^2*B^2*a^4*b^4*d^8 + 28*A^2*B^2*a^2*b^6*d^8 + 6*A^2*B
^2*a^6*b^2*d^8 + 6*C^4*a*b^7*c^5*d^3 + 4*C^4*a*b^7*c^3*d^5 - 2*C^4*a^5*b^3*
c*d^7 + 12*B^4*a^3*b^5*c*d^7 - 12*B^4*a*b^7*c^5*d^3 + 8*B^4*a^5*b^3*c*d^7 -
 4*B^4*a*b^7*c^3*d^5 - 48*A^4*a^3*b^5*c*d^7 - 20*A^4*a^5*b^3*c*d^7 - 8*A^4*
a*b^7*c^3*d^5 - 10*B^3*C*b^8*c^5*d^3 - 10*B*C^3*b^8*c^5*d^3 - 4*B^3*C*b^8*c
^3*d^5 - 4*B*C^3*b^8*c^3*d^5 + 23*A^3*C*b^8*c^4*d^4 - 18*A^3*C*b^8*c^2*d^6
+ 11*A*C^3*b^8*c^4*d^4 - 9*A*C^3*b^8*c^6*d^2 + 6*A*C^3*b^8*c^2*d^6 - 3*A^3*
C*b^8*c^6*d^2 - 20*A^3*B*b^8*c^3*d^5 - 20*A*B^3*b^8*c^3*d^5 + 4*A^3*B*b^8*c
^5*d^3 + 4*A*B^3*b^8*c^5*d^3 - 63*A^3*C*a^4*b^4*d^8 - 54*A^3*C*a^2*b^6*d^8
+ 9*A^3*C*a^6*b^2*d^8 + 9*A*C^3*a^6*b^2*d^8 - 3*A*C^3*a^4*b^4*d^8 - 28*A^3*
B*a^5*b^3*d^8 - 28*A*B^3*a^5*b^3*d^8 - 18*A^3*B*a^3*b^5*d^8 - 18*A*B^3*a^3*
b^5*d^8 + B^3*C*a^5*b^3*c^2*d^6 + B*C^3*a^5*b^3*c^2*d^6 + 6*C^4*a^7*b*c*d^7
 + 4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 12*A^3*B*b^8*c*d^7 - 12*A*B^3*b
^8*c*d^7 - 3*B^3*C*a^7*b*d^8 - 3*B*C^3*a^7*b*d^8 - 6*A^3*B*a*b^7*d^8 - 6*A*
B^3*a*b^7*d^8 + 30*C^4*a^3*b^5*c^5*d^3 + 19*C^4*a^4*b^4*c^2*d^6 + 9*C^4*a^6
*b^2*c^4*d^4 - 9*C^4*a^2*b^6*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*b^
6*c^2*d^6 + 3*C^4*a^6*b^2*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^2*b^6*c
^4*d^4 + 28*B^4*a^5*b^3*c^3*d^5 + 27*B^4*a^2*b^6*c^4*d^4 - 17*B^4*a^4*b^4*c

^4*d^4 - 10*B^4*a^4*b^4*c^2*d^6 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c^2
*d^6 - 6*B^4*a^6*b^2*c^2*d^6 + 4*B^4*a^3*b^5*c^5*d^3 + 70*A^4*a^4*b^4*c^2*d
^6 + 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^2*b^6*c^4*d
^4 + B^2*C^2*a^2*b^6*d^8 - 18*A^3*C*b^8*d^8 + B^3*C*a^5*b^3*d^8 + B*C^3*a^5
*b^3*d^8 + 3*C^4*b^8*c^6*d^2 + 8*B^4*b^8*c^4*d^4 + 4*B^4*b^8*c^2*d^6 + 12*A
^4*b^8*c^2*d^6 - 5*A^4*b^8*c^4*d^4 + 6*B^4*a^6*b^2*d^8 + 3*B^4*a^4*b^4*d^8
+ 30*A^4*a^4*b^4*d^8 + 27*A^4*a^2*b^6*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2*b
^8*d^8 + 9*A^4*b^8*d^8 + C^4*b^8*c^4*d^4 + B^4*a^2*b^6*d^8, f, k)*(root(640
*a^13*b^7*c*d^15*f^4 + 640*a^7*b^13*c^15*d*f^4 + 480*a^15*b^5*c*d^15*f^4 +
480*a^11*b^9*c*d^15*f^4 + 480*a^9*b^11*c^15*d*f^4 + 480*a^5*b^15*c^15*d*f^4
+ 192*a^19*b*c^5*d^11*f^4 + 192*a^17*b^3*c*d^15*f^4 + 192*a^11*b^9*c^15*d*
f^4 + 192*a^9*b^11*c*d^15*f^4 + 192*a^3*b^17*c^15*d*f^4 + 192*a*b^19*c^11*d
^5*f^4 + 128*a^19*b*c^7*d^9*f^4 + 128*a^19*b*c^3*d^13*f^4 + 128*a*b^19*c^13
*d^3*f^4 + 128*a*b^19*c^9*d^7*f^4 + 32*a^19*b*c^9*d^7*f^4 + 32*a^13*b^7*c^1
5*d*f^4 + 32*a^7*b^13*c*d^15*f^4 + 32*a*b^19*c^7*d^9*f^4 + 32*a^19*b*c*d^15
*f^4 + 32*a*b^19*c^15*d*f^4 - 47088*a^10*b^10*c^8*d^8*f^4 + 42432*a^11*b^9*
c^7*d^9*f^4 + 42432*a^9*b^11*c^9*d^7*f^4 + 39328*a^11*b^9*c^9*d^7*f^4 + 393
28*a^9*b^11*c^7*d^9*f^4 - 36912*a^12*b^8*c^8*d^8*f^4 - 36912*a^8*b^12*c^8*d
^8*f^4 - 34256*a^10*b^10*c^10*d^6*f^4 - 34256*a^10*b^10*c^6*d^10*f^4 - 3115
2*a^12*b^8*c^6*d^10*f^4 - 31152*a^8*b^12*c^10*d^6*f^4 + 28128*a^13*b^7*c^7*
d^9*f^4 + 28128*a^7*b^13*c^9*d^7*f^4 + 24160*a^11*b^9*c^5*d^11*f^4 + 24160*
a^9*b^11*c^11*d^5*f^4 - 23088*a^12*b^8*c^10*d^6*f^4 - 23088*a^8*b^12*c^6*d^
10*f^4 + 22272*a^13*b^7*c^9*d^7*f^4 + 22272*a^7*b^13*c^7*d^9*f^4 + 19072*a^
11*b^9*c^11*d^5*f^4 + 19072*a^9*b^11*c^5*d^11*f^4 + 18624*a^13*b^7*c^5*d^11
*f^4 + 18624*a^7*b^13*c^11*d^5*f^4 - 17328*a^14*b^6*c^8*d^8*f^4 - 17328*a^6
*b^14*c^8*d^8*f^4 - 17232*a^14*b^6*c^6*d^10*f^4 - 17232*a^6*b^14*c^10*d^6*f
^4 - 13520*a^12*b^8*c^4*d^12*f^4 - 13520*a^8*b^12*c^12*d^4*f^4 - 12464*a^10
*b^10*c^12*d^4*f^4 - 12464*a^10*b^10*c^4*d^12*f^4 + 10880*a^15*b^5*c^7*d^9*
f^4 + 10880*a^5*b^15*c^9*d^7*f^4 - 9072*a^14*b^6*c^10*d^6*f^4 - 9072*a^6*b^
14*c^6*d^10*f^4 + 8928*a^13*b^7*c^11*d^5*f^4 + 8928*a^7*b^13*c^5*d^11*f^4 -
8880*a^14*b^6*c^4*d^12*f^4 - 8880*a^6*b^14*c^12*d^4*f^4 + 8480*a^15*b^5*c^
5*d^11*f^4 + 8480*a^5*b^15*c^11*d^5*f^4 + 7200*a^15*b^5*c^9*d^7*f^4 + 7200*
a^5*b^15*c^7*d^9*f^4 - 6912*a^12*b^8*c^12*d^4*f^4 - 6912*a^8*b^12*c^4*d^12*
f^4 + 6400*a^11*b^9*c^3*d^13*f^4 + 6400*a^9*b^11*c^13*d^3*f^4 + 5920*a^13*b
^7*c^3*d^13*f^4 + 5920*a^7*b^13*c^13*d^3*f^4 - 5392*a^16*b^4*c^6*d^10*f^4 -
5392*a^4*b^16*c^10*d^6*f^4 - 4428*a^16*b^4*c^8*d^8*f^4 - 4428*a^4*b^16*c^8
*d^8*f^4 + 4128*a^11*b^9*c^13*d^3*f^4 + 4128*a^9*b^11*c^3*d^13*f^4 - 3328*a
^16*b^4*c^4*d^12*f^4 - 3328*a^4*b^16*c^12*d^4*f^4 + 3264*a^15*b^5*c^3*d^13*
f^4 + 3264*a^5*b^15*c^13*d^3*f^4 - 2480*a^12*b^8*c^2*d^14*f^4 - 2480*a^8*b^
12*c^14*d^2*f^4 + 2240*a^15*b^5*c^11*d^5*f^4 + 2240*a^5*b^15*c^5*d^11*f^4 -
2128*a^14*b^6*c^12*d^4*f^4 - 2128*a^6*b^14*c^4*d^12*f^4 + 2112*a^17*b^3*c^
7*d^9*f^4 + 2112*a^3*b^17*c^9*d^7*f^4 + 2048*a^17*b^3*c^5*d^11*f^4 + 2048*a
^3*b^17*c^11*d^5*f^4 - 2000*a^14*b^6*c^2*d^14*f^4 - 2000*a^6*b^14*c^14*d^2*
f^4 - 1792*a^16*b^4*c^10*d^6*f^4 - 1792*a^4*b^16*c^6*d^10*f^4 - 1776*a^10*b
^10*c^14*d^2*f^4 - 1776*a^10*b^10*c^2*d^14*f^4 + 1472*a^13*b^7*c^13*d^3*f^4

+ 1472*a^7*b^13*c^3*d^13*f^4 + 1088*a^17*b^3*c^9*d^7*f^4 + 1088*a^3*b^17*c
^7*d^9*f^4 + 992*a^17*b^3*c^3*d^13*f^4 + 992*a^3*b^17*c^13*d^3*f^4 - 912*a^
16*b^4*c^2*d^14*f^4 - 912*a^4*b^16*c^14*d^2*f^4 - 768*a^18*b^2*c^6*d^10*f^4
 - 768*a^2*b^18*c^10*d^6*f^4 - 688*a^12*b^8*c^14*d^2*f^4 - 688*a^8*b^12*c^2
*d^14*f^4 - 592*a^18*b^2*c^4*d^12*f^4 - 592*a^2*b^18*c^12*d^4*f^4 - 472*a^1
8*b^2*c^8*d^8*f^4 - 472*a^2*b^18*c^8*d^8*f^4 - 280*a^16*b^4*c^12*d^4*f^4 -
280*a^4*b^16*c^4*d^12*f^4 + 224*a^17*b^3*c^11*d^5*f^4 + 224*a^15*b^5*c^13*d
^3*f^4 + 224*a^5*b^15*c^3*d^13*f^4 + 224*a^3*b^17*c^5*d^11*f^4 - 208*a^18*b
^2*c^2*d^14*f^4 - 208*a^2*b^18*c^14*d^2*f^4 - 112*a^18*b^2*c^10*d^6*f^4 - 1
12*a^14*b^6*c^14*d^2*f^4 - 112*a^6*b^14*c^2*d^14*f^4 - 112*a^2*b^18*c^6*d^1
0*f^4 - 24*b^20*c^12*d^4*f^4 - 16*b^20*c^14*d^2*f^4 - 16*b^20*c^10*d^6*f^4
 - 4*b^20*c^8*d^8*f^4 - 24*a^20*c^4*d^12*f^4 - 16*a^20*c^6*d^10*f^4 - 16*a^2
0*c^2*d^14*f^4 - 4*a^20*c^8*d^8*f^4 - 80*a^14*b^6*d^16*f^4 - 60*a^16*b^4*d^
16*f^4 - 60*a^12*b^8*d^16*f^4 - 24*a^18*b^2*d^16*f^4 - 24*a^10*b^10*d^16*f^
4 - 4*a^8*b^12*d^16*f^4 - 80*a^6*b^14*c^16*f^4 - 60*a^8*b^12*c^16*f^4 - 60*
a^4*b^16*c^16*f^4 - 24*a^10*b^10*c^16*f^4 - 24*a^2*b^18*c^16*f^4 - 4*a^12*b
^8*c^16*f^4 - 4*b^20*c^16*f^4 - 4*a^20*d^16*f^4 + 56*A*C*a^13*b*c*d^11*f^2
 - 48*A*C*a*b^13*c^11*d*f^2 + 48*A*C*a*b^13*c*d^11*f^2 + 5904*B*C*a^7*b^7*c^
6*d^6*f^2 - 5016*B*C*a^8*b^6*c^5*d^7*f^2 - 4608*B*C*a^6*b^8*c^7*d^5*f^2 - 4
512*B*C*a^6*b^8*c^5*d^7*f^2 - 4384*B*C*a^8*b^6*c^7*d^5*f^2 + 3056*B*C*a^7*b
^7*c^8*d^4*f^2 + 2256*B*C*a^7*b^7*c^4*d^8*f^2 - 1824*B*C*a^8*b^6*c^3*d^9*f^
2 + 1632*B*C*a^4*b^10*c^9*d^3*f^2 - 1400*B*C*a^3*b^11*c^8*d^4*f^2 - 1320*B*
C*a^11*b^3*c^4*d^8*f^2 - 1248*B*C*a^6*b^8*c^3*d^9*f^2 + 1152*B*C*a^10*b^4*c
^3*d^9*f^2 - 1072*B*C*a^6*b^8*c^9*d^3*f^2 + 1068*B*C*a^9*b^5*c^6*d^6*f^2 -
1004*B*C*a^5*b^9*c^4*d^8*f^2 - 968*B*C*a^3*b^11*c^6*d^6*f^2 - 864*B*C*a^5*b
^9*c^8*d^4*f^2 - 828*B*C*a^9*b^5*c^4*d^8*f^2 - 792*B*C*a^11*b^3*c^2*d^10*f^
2 - 792*B*C*a^3*b^11*c^4*d^8*f^2 - 776*B*C*a^8*b^6*c^9*d^3*f^2 + 688*B*C*a^
4*b^10*c^7*d^5*f^2 - 672*B*C*a^3*b^11*c^10*d^2*f^2 - 592*B*C*a^9*b^5*c^2*d^
10*f^2 + 544*B*C*a^7*b^7*c^10*d^2*f^2 - 492*B*C*a^5*b^9*c^2*d^10*f^2 + 480*
B*C*a^10*b^4*c^5*d^7*f^2 - 392*B*C*a^5*b^9*c^10*d^2*f^2 + 332*B*C*a^9*b^5*c
^8*d^4*f^2 - 328*B*C*a^11*b^3*c^6*d^6*f^2 + 320*B*C*a^2*b^12*c^9*d^3*f^2 +
272*B*C*a^12*b^2*c^3*d^9*f^2 - 248*B*C*a^4*b^10*c^5*d^7*f^2 - 248*B*C*a^3*b
^11*c^2*d^10*f^2 - 208*B*C*a^10*b^4*c^7*d^5*f^2 - 192*B*C*a^2*b^12*c^5*d^7*
f^2 + 144*B*C*a^7*b^7*c^2*d^10*f^2 - 96*B*C*a^4*b^10*c^3*d^9*f^2 + 88*B*C*a
^12*b^2*c^5*d^7*f^2 - 72*B*C*a^11*b^3*c^8*d^4*f^2 - 48*B*C*a^12*b^2*c^7*d^5
*f^2 + 48*B*C*a^10*b^4*c^9*d^3*f^2 - 48*B*C*a^2*b^12*c^7*d^5*f^2 - 48*B*C*a
^2*b^12*c^3*d^9*f^2 - 12*B*C*a^9*b^5*c^10*d^2*f^2 + 4*B*C*a^5*b^9*c^6*d^6*f
^2 + 5824*A*C*a^5*b^9*c^7*d^5*f^2 - 4378*A*C*a^6*b^8*c^8*d^4*f^2 + 4296*A*C
*a^5*b^9*c^5*d^7*f^2 - 3912*A*C*a^6*b^8*c^6*d^6*f^2 - 3672*A*C*a^9*b^5*c^5*
d^7*f^2 + 3594*A*C*a^8*b^6*c^4*d^8*f^2 + 3236*A*C*a^8*b^6*c^6*d^6*f^2 + 281
6*A*C*a^5*b^9*c^9*d^3*f^2 + 2624*A*C*a^5*b^9*c^3*d^9*f^2 + 2432*A*C*a^7*b^7
*c^7*d^5*f^2 - 2366*A*C*a^4*b^10*c^8*d^4*f^2 + 2298*A*C*a^10*b^4*c^4*d^8*f^
2 + 1872*A*C*a^7*b^7*c^3*d^9*f^2 + 1848*A*C*a^10*b^4*c^6*d^6*f^2 - 1644*A*C
*a^4*b^10*c^6*d^6*f^2 - 1488*A*C*a^9*b^5*c^7*d^5*f^2 - 1408*A*C*a^9*b^5*c^3
*d^9*f^2 - 1308*A*C*a^6*b^8*c^4*d^8*f^2 + 1248*A*C*a^7*b^7*c^5*d^7*f^2 - 10

12*A*C*a^6*b^8*c^10*d^2*f^2 + 1008*A*C*a^3*b^11*c^7*d^5*f^2 + 992*A*C*a^3*b^11*c^5*d^7*f^2 + 928*A*C*a^3*b^11*c^3*d^9*f^2 + 848*A*C*a^7*b^7*c^9*d^3*f^2 + 636*A*C*a^8*b^6*c^2*d^10*f^2 - 628*A*C*a^4*b^10*c^10*d^2*f^2 - 600*A*C*a^6*b^8*c^2*d^10*f^2 - 576*A*C*a^11*b^3*c^5*d^7*f^2 + 572*A*C*a^10*b^4*c^2*d^10*f^2 + 464*A*C*a^8*b^6*c^8*d^4*f^2 - 304*A*C*a^4*b^10*c^4*d^8*f^2 + 304*A*C*a^2*b^12*c^6*d^6*f^2 + 296*A*C*a^2*b^12*c^4*d^8*f^2 + 260*A*C*a^10*b^4*c^8*d^4*f^2 - 232*A*C*a^12*b^2*c^2*d^10*f^2 - 232*A*C*a^9*b^5*c^9*d^3*f^2 + 228*A*C*a^2*b^12*c^10*d^2*f^2 - 188*A*C*a^4*b^10*c^2*d^10*f^2 + 144*A*C*a^11*b^3*c^3*d^9*f^2 + 116*A*C*a^12*b^2*c^6*d^6*f^2 - 112*A*C*a^11*b^3*c^7*d^5*f^2 + 112*A*C*a^3*b^11*c^9*d^3*f^2 + 92*A*C*a^8*b^6*c^10*d^2*f^2 + 74*A*C*a^12*b^2*c^4*d^8*f^2 + 62*A*C*a^2*b^12*c^8*d^4*f^2 + 40*A*C*a^2*b^12*c^2*d^10*f^2 - 7008*A*B*a^7*b^7*c^6*d^6*f^2 - 4032*A*B*a^7*b^7*c^4*d^8*f^2 + 3952*A*B*a^8*b^6*c^7*d^5*f^2 + 3648*A*B*a^8*b^6*c^5*d^7*f^2 - 3392*A*B*a^7*b^7*c^8*d^4*f^2 + 3264*A*B*a^6*b^8*c^7*d^5*f^2 - 2992*A*B*a^4*b^10*c^5*d^7*f^2 - 2368*A*B*a^4*b^10*c^7*d^5*f^2 - 2304*A*B*a^4*b^10*c^3*d^9*f^2 - 1968*A*B*a^9*b^5*c^6*d^6*f^2 - 1872*A*B*a^4*b^10*c^9*d^3*f^2 - 1728*A*B*a^7*b^7*c^2*d^10*f^2 + 1712*A*B*a^3*b^11*c^8*d^4*f^2 - 1536*A*B*a^10*b^4*c^3*d^9*f^2 + 1536*A*B*a^6*b^8*c^5*d^7*f^2 - 1392*A*B*a^2*b^12*c^5*d^7*f^2 + 1328*A*B*a^3*b^11*c^6*d^6*f^2 - 1104*A*B*a^2*b^12*c^3*d^9*f^2 - 1056*A*B*a^6*b^8*c^3*d^9*f^2 + 976*A*B*a^6*b^8*c^9*d^3*f^2 + 960*A*B*a^11*b^3*c^4*d^8*f^2 + 936*A*B*a^5*b^9*c^8*d^4*f^2 - 912*A*B*a^10*b^4*c^5*d^7*f^2 + 848*A*B*a^8*b^6*c^9*d^3*f^2 + 816*A*B*a^3*b^11*c^4*d^8*f^2 - 816*A*B*a^2*b^12*c^7*d^5*f^2 + 768*A*B*a^3*b^11*c^10*d^2*f^2 + 672*A*B*a^8*b^6*c^3*d^9*f^2 - 632*A*B*a^9*b^5*c^8*d^4*f^2 - 608*A*B*a^9*b^5*c^2*d^10*f^2 - 552*A*B*a^9*b^5*c^4*d^8*f^2 - 544*A*B*a^7*b^7*c^10*d^2*f^2 - 480*A*B*a^5*b^9*c^2*d^10*f^2 + 464*A*B*a^5*b^9*c^10*d^2*f^2 - 464*A*B*a^2*b^12*c^9*d^3*f^2 + 432*A*B*a^11*b^3*c^2*d^10*f^2 - 368*A*B*a^12*b^2*c^3*d^9*f^2 - 256*A*B*a^5*b^9*c^6*d^6*f^2 - 208*A*B*a^12*b^2*c^5*d^7*f^2 + 176*A*B*a^5*b^9*c^4*d^8*f^2 + 112*A*B*a^11*b^3*c^6*d^6*f^2 + 112*A*B*a^10*b^4*c^7*d^5*f^2 - 16*A*B*a^3*b^11*c^2*d^10*f^2 - 576*B*C*a^8*b^6*c*d^11*f^2 + 400*B*C*a^4*b^10*c^11*d*f^2 - 288*B*C*a^6*b^8*c*d^11*f^2 - 176*B*C*a^6*b^8*c^11*d*f^2 + 128*B*C*a^10*b^4*c*d^11*f^2 - 108*B*C*a*b^13*c^4*d^8*f^2 - 104*B*C*a^4*b^10*c*d^11*f^2 - 92*B*C*a^13*b*c^4*d^8*f^2 - 60*B*C*a*b^13*c^8*d^4*f^2 - 60*B*C*a*b^13*c^6*d^6*f^2 + 48*B*C*a^2*b^12*c^11*d*f^2 - 40*B*C*a*b^13*c^2*d^10*f^2 - 28*B*C*a^13*b*c^2*d^10*f^2 - 24*B*C*a^12*b^2*c*d^11*f^2 + 20*B*C*a*b^13*c^10*d^2*f^2 - 16*B*C*a^2*b^12*c*d^11*f^2 + 12*B*C*a^13*b*c^6*d^6*f^2 + 912*A*C*a^7*b^7*c*d^11*f^2 + 808*A*C*a^5*b^9*c*d^11*f^2 + 432*A*C*a^5*b^9*c^11*d*f^2 + 336*A*C*a^3*b^11*c*d^11*f^2 + 224*A*C*a^11*b^3*c*d^11*f^2 - 112*A*C*a^3*b^11*c^11*d*f^2 + 112*A*C*a*b^13*c^3*d^9*f^2 - 88*A*C*a*b^13*c^9*d^3*f^2 + 80*A*C*a^13*b*c^3*d^9*f^2 + 56*A*C*a*b^13*c^5*d^7*f^2 + 48*A*C*a^9*b^5*c*d^11*f^2 - 40*A*C*a^13*b*c^5*d^7*f^2 - 16*A*C*a^7*b^7*c^11*d*f^2 + 16*A*C*a*b^13*c^7*d^5*f^2 - 496*A*B*a^4*b^10*c*d^11*f^2 - 400*A*B*a^4*b^10*c^11*d*f^2 + 288*A*B*a^8*b^6*c*d^11*f^2 - 288*A*B*a^6*b^8*c*d^11*f^2 - 272*A*B*a^2*b^12*c*d^11*f^2 + 240*A*B*a*b^13*c^6*d^6*f^2 - 224*A*B*a^10*b^4*c*d^11*f^2 + 192*A*B*a*b^13*c^8*d^4*f^2 + 192*A*B*a*b^13*c^4*d^8*f^2 + 176*A*B*a^6*b^8*c^11*d*f^2 + 104*A*B*a^13*b

*c^4*d^8*f^2 - 48*A*B*a^2*b^12*c^11*d*f^2 + 16*A*B*a^13*b*c^2*d^10*f^2 + 16
*A*B*a*b^13*c^10*d^2*f^2 + 16*A*B*a*b^13*c^2*d^10*f^2 - 96*B*C*b^14*c^7*d^5
*f^2 - 72*B*C*b^14*c^5*d^7*f^2 - 24*B*C*b^14*c^9*d^3*f^2 - 16*B*C*b^14*c^3*
d^9*f^2 + 116*A*C*b^14*c^6*d^6*f^2 + 100*A*C*b^14*c^4*d^8*f^2 + 24*A*C*b^14
*c^2*d^10*f^2 + 22*A*C*b^14*c^8*d^4*f^2 + 16*B*C*a^14*c^3*d^9*f^2 + 8*A*C*b
^14*c^10*d^2*f^2 - 192*A*B*b^14*c^5*d^7*f^2 - 176*A*B*b^14*c^3*d^9*f^2 - 11
2*B*C*a^11*b^3*d^12*f^2 - 48*A*B*b^14*c^7*d^5*f^2 - 28*A*C*a^14*c^2*d^10*f^
2 + 4*B*C*a^5*b^9*d^12*f^2 + 2*A*C*a^14*c^4*d^8*f^2 + 150*A*C*a^10*b^4*d^12
*f^2 - 80*B*C*a^3*b^11*c^12*f^2 + 66*A*C*a^8*b^6*d^12*f^2 - 30*A*C*a^12*b^2
*d^12*f^2 + 24*B*C*a^5*b^9*c^12*f^2 - 16*A*B*a^14*c^3*d^9*f^2 - 12*A*C*a^4*
b^10*d^12*f^2 - 576*A*B*a^7*b^7*d^12*f^2 - 432*A*B*a^9*b^5*d^12*f^2 - 400*A
*B*a^5*b^9*d^12*f^2 - 144*A*B*a^3*b^11*d^12*f^2 - 66*A*C*a^4*b^10*c^12*f^2
+ 54*A*C*a^2*b^12*c^12*f^2 - 32*A*B*a^11*b^3*d^12*f^2 + 2*A*C*a^6*b^8*c^12*
f^2 + 80*A*B*a^3*b^11*c^12*f^2 - 24*A*B*a^5*b^9*c^12*f^2 + 2508*C^2*a^6*b^8
*c^6*d^6*f^2 + 2376*C^2*a^9*b^5*c^5*d^7*f^2 + 2357*C^2*a^6*b^8*c^8*d^4*f^2
- 2048*C^2*a^5*b^9*c^7*d^5*f^2 + 1304*C^2*a^9*b^5*c^3*d^9*f^2 + 1303*C^2*a^
4*b^10*c^8*d^4*f^2 + 1212*C^2*a^4*b^10*c^6*d^6*f^2 - 1203*C^2*a^8*b^6*c^4*d
^8*f^2 - 1192*C^2*a^5*b^9*c^9*d^3*f^2 + 1062*C^2*a^6*b^8*c^4*d^8*f^2 + 984*
C^2*a^9*b^5*c^7*d^5*f^2 - 952*C^2*a^8*b^6*c^6*d^6*f^2 + 768*C^2*a^7*b^7*c^5
*d^7*f^2 - 681*C^2*a^10*b^4*c^4*d^8*f^2 - 672*C^2*a^5*b^9*c^5*d^7*f^2 - 480
*C^2*a^10*b^4*c^6*d^6*f^2 + 458*C^2*a^6*b^8*c^10*d^2*f^2 - 448*C^2*a^7*b^7*
c^7*d^5*f^2 + 422*C^2*a^4*b^10*c^4*d^8*f^2 + 372*C^2*a^6*b^8*c^2*d^10*f^2 +
360*C^2*a^11*b^3*c^5*d^7*f^2 + 312*C^2*a^7*b^7*c^3*d^9*f^2 + 278*C^2*a^4*b
^10*c^10*d^2*f^2 - 232*C^2*a^7*b^7*c^9*d^3*f^2 + 194*C^2*a^12*b^2*c^2*d^10*
f^2 + 176*C^2*a^9*b^5*c^9*d^3*f^2 + 152*C^2*a^3*b^11*c^5*d^7*f^2 + 124*C^2*
a^4*b^10*c^2*d^10*f^2 - 120*C^2*a^3*b^11*c^7*d^5*f^2 - 114*C^2*a^2*b^12*c^1
0*d^2*f^2 - 102*C^2*a^8*b^6*c^2*d^10*f^2 + 101*C^2*a^12*b^2*c^4*d^8*f^2 + 1
00*C^2*a^2*b^12*c^6*d^6*f^2 - 88*C^2*a^5*b^9*c^3*d^9*f^2 + 77*C^2*a^2*b^12*
c^8*d^4*f^2 + 72*C^2*a^11*b^3*c^3*d^9*f^2 - 64*C^2*a^8*b^6*c^10*d^2*f^2 + 6
4*C^2*a^3*b^11*c^3*d^9*f^2 - 58*C^2*a^10*b^4*c^2*d^10*f^2 + 56*C^2*a^12*b^2
*c^6*d^6*f^2 + 56*C^2*a^11*b^3*c^7*d^5*f^2 + 40*C^2*a^3*b^11*c^9*d^3*f^2 +
36*C^2*a^12*b^2*c^8*d^4*f^2 + 32*C^2*a^2*b^12*c^4*d^8*f^2 + 26*C^2*a^10*b^4
*c^8*d^4*f^2 + 16*C^2*a^2*b^12*c^2*d^10*f^2 + 2*C^2*a^8*b^6*c^8*d^4*f^2 + 2
277*B^2*a^8*b^6*c^4*d^8*f^2 + 2144*B^2*a^5*b^9*c^7*d^5*f^2 - 2112*B^2*a^9*b
^5*c^5*d^7*f^2 + 2028*B^2*a^8*b^6*c^6*d^6*f^2 - 1671*B^2*a^6*b^8*c^8*d^4*f^
2 + 1275*B^2*a^10*b^4*c^4*d^8*f^2 + 1176*B^2*a^5*b^9*c^5*d^7*f^2 + 1096*B^2
*a^5*b^9*c^9*d^3*f^2 - 1044*B^2*a^6*b^8*c^6*d^6*f^2 + 984*B^2*a^10*b^4*c^6*
d^6*f^2 - 968*B^2*a^9*b^5*c^3*d^9*f^2 - 888*B^2*a^9*b^5*c^7*d^5*f^2 + 672*B
^2*a^7*b^7*c^7*d^5*f^2 + 664*B^2*a^5*b^9*c^3*d^9*f^2 - 649*B^2*a^4*b^10*c^8
*d^4*f^2 + 618*B^2*a^8*b^6*c^2*d^10*f^2 + 514*B^2*a^4*b^10*c^4*d^8*f^2 + 46
0*B^2*a^2*b^12*c^6*d^6*f^2 + 422*B^2*a^8*b^6*c^8*d^4*f^2 + 406*B^2*a^10*b^4
*c^2*d^10*f^2 - 382*B^2*a^6*b^8*c^10*d^2*f^2 + 368*B^2*a^2*b^12*c^4*d^8*f^2
- 312*B^2*a^11*b^3*c^5*d^7*f^2 + 312*B^2*a^7*b^7*c^3*d^9*f^2 + 248*B^2*a^7
*b^7*c^9*d^3*f^2 + 245*B^2*a^2*b^12*c^8*d^4*f^2 - 192*B^2*a^7*b^7*c^5*d^7*f
^2 - 184*B^2*a^3*b^11*c^9*d^3*f^2 + 182*B^2*a^2*b^12*c^10*d^2*f^2 + 176*B^2

*a^3*b^11*c^3*d^9*f^2 + 174*B^2*a^6*b^8*c^4*d^8*f^2 - 170*B^2*a^4*b^10*c^10
*d^2*f^2 - 152*B^2*a^9*b^5*c^9*d^3*f^2 + 152*B^2*a^4*b^10*c^2*d^10*f^2 + 14
2*B^2*a^10*b^4*c^8*d^4*f^2 - 90*B^2*a^12*b^2*c^2*d^10*f^2 + 88*B^2*a^2*b^12
*c^2*d^10*f^2 + 84*B^2*a^8*b^6*c^10*d^2*f^2 + 84*B^2*a^6*b^8*c^2*d^10*f^2 +
 60*B^2*a^12*b^2*c^6*d^6*f^2 - 56*B^2*a^11*b^3*c^7*d^5*f^2 + 53*B^2*a^12*b^
2*c^4*d^8*f^2 + 24*B^2*a^11*b^3*c^3*d^9*f^2 + 24*B^2*a^4*b^10*c^6*d^6*f^2 +
 24*B^2*a^3*b^11*c^7*d^5*f^2 - 8*B^2*a^3*b^11*c^5*d^7*f^2 + 4566*A^2*a^6*b^
8*c^4*d^8*f^2 + 4284*A^2*a^6*b^8*c^6*d^6*f^2 - 3776*A^2*a^5*b^9*c^7*d^5*f^2
 - 3624*A^2*a^5*b^9*c^5*d^7*f^2 + 3122*A^2*a^4*b^10*c^4*d^8*f^2 + 3108*A^2*
a^6*b^8*c^2*d^10*f^2 + 2741*A^2*a^6*b^8*c^8*d^4*f^2 + 2592*A^2*a^4*b^10*c^6
*d^6*f^2 - 2536*A^2*a^5*b^9*c^3*d^9*f^2 + 2224*A^2*a^4*b^10*c^2*d^10*f^2 -
2184*A^2*a^7*b^7*c^3*d^9*f^2 - 2016*A^2*a^7*b^7*c^5*d^7*f^2 - 1984*A^2*a^7*
b^7*c^7*d^5*f^2 + 1626*A^2*a^8*b^6*c^2*d^10*f^2 - 1624*A^2*a^5*b^9*c^9*d^3*
f^2 + 1603*A^2*a^4*b^10*c^8*d^4*f^2 + 1296*A^2*a^9*b^5*c^5*d^7*f^2 - 1144*A
^2*a^3*b^11*c^5*d^7*f^2 - 992*A^2*a^3*b^11*c^3*d^9*f^2 + 968*A^2*a^2*b^12*c
^4*d^8*f^2 - 888*A^2*a^3*b^11*c^7*d^5*f^2 + 849*A^2*a^8*b^6*c^4*d^8*f^2 + 8
08*A^2*a^2*b^12*c^2*d^10*f^2 - 616*A^2*a^7*b^7*c^9*d^3*f^2 + 554*A^2*a^6*b^
8*c^10*d^2*f^2 - 504*A^2*a^10*b^4*c^6*d^6*f^2 + 504*A^2*a^9*b^5*c^7*d^5*f^2
 + 460*A^2*a^2*b^12*c^6*d^6*f^2 + 350*A^2*a^10*b^4*c^2*d^10*f^2 + 350*A^2*a
^4*b^10*c^10*d^2*f^2 - 321*A^2*a^10*b^4*c^4*d^8*f^2 + 216*A^2*a^11*b^3*c^5*
d^7*f^2 - 216*A^2*a^11*b^3*c^3*d^9*f^2 + 182*A^2*a^12*b^2*c^2*d^10*f^2 - 15
2*A^2*a^3*b^11*c^9*d^3*f^2 - 124*A^2*a^8*b^6*c^6*d^6*f^2 - 114*A^2*a^2*b^12
*c^10*d^2*f^2 + 104*A^2*a^9*b^5*c^3*d^9*f^2 + 77*A^2*a^2*b^12*c^8*d^4*f^2 +
 74*A^2*a^8*b^6*c^8*d^4*f^2 - 70*A^2*a^10*b^4*c^8*d^4*f^2 + 56*A^2*a^11*b^3
*c^7*d^5*f^2 + 56*A^2*a^9*b^5*c^9*d^3*f^2 + 41*A^2*a^12*b^2*c^4*d^8*f^2 - 2
8*A^2*a^12*b^2*c^6*d^6*f^2 - 28*A^2*a^8*b^6*c^10*d^2*f^2 - 16*B*C*b^14*c^11
*d*f^2 - 16*B*C*a^14*c*d^11*f^2 - 48*A*B*b^14*c*d^11*f^2 + 16*A*B*b^14*c^11
*d*f^2 + 12*B*C*a^13*b*d^12*f^2 + 24*B*C*a*b^13*c^12*f^2 + 16*A*B*a^14*c*d^
11*f^2 - 24*A*B*a^13*b*d^12*f^2 - 24*A*B*a*b^13*d^12*f^2 - 24*A*B*a*b^13*c^
12*f^2 + 216*C^2*a^9*b^5*c*d^11*f^2 - 216*C^2*a^5*b^9*c^11*d*f^2 + 56*C^2*a
^3*b^11*c^11*d*f^2 + 56*C^2*a*b^13*c^9*d^3*f^2 + 56*C^2*a*b^13*c^5*d^7*f^2
- 40*C^2*a^11*b^3*c*d^11*f^2 + 40*C^2*a*b^13*c^7*d^5*f^2 + 32*C^2*a^13*b*c^
5*d^7*f^2 - 24*C^2*a^7*b^7*c*d^11*f^2 - 16*C^2*a^13*b*c^3*d^9*f^2 + 16*C^2*
a*b^13*c^3*d^9*f^2 + 8*C^2*a^7*b^7*c^11*d*f^2 - 8*C^2*a^5*b^9*c*d^11*f^2 +
264*B^2*a^7*b^7*c*d^11*f^2 + 224*B^2*a^5*b^9*c*d^11*f^2 + 168*B^2*a^5*b^9*c
^11*d*f^2 - 112*B^2*a*b^13*c^9*d^3*f^2 - 104*B^2*a^3*b^11*c^11*d*f^2 - 104*
B^2*a*b^13*c^7*d^5*f^2 + 96*B^2*a^3*b^11*c*d^11*f^2 + 88*B^2*a^11*b^3*c*d^1
1*f^2 - 72*B^2*a^9*b^5*c*d^11*f^2 - 64*B^2*a*b^13*c^5*d^7*f^2 + 32*B^2*a^13
*b*c^3*d^9*f^2 - 24*B^2*a^13*b*c^5*d^7*f^2 - 24*B^2*a^7*b^7*c^11*d*f^2 + 16
*B^2*a*b^13*c^3*d^9*f^2 - 888*A^2*a^7*b^7*c*d^11*f^2 - 800*A^2*a^5*b^9*c*d^
11*f^2 - 336*A^2*a^3*b^11*c*d^11*f^2 - 264*A^2*a^9*b^5*c*d^11*f^2 - 216*A^2
*a^5*b^9*c^11*d*f^2 - 184*A^2*a^11*b^3*c*d^11*f^2 - 128*A^2*a*b^13*c^3*d^9*
f^2 - 112*A^2*a*b^13*c^5*d^7*f^2 - 64*A^2*a^13*b*c^3*d^9*f^2 + 56*A^2*a^3*b
^11*c^11*d*f^2 - 56*A^2*a*b^13*c^7*d^5*f^2 + 32*A^2*a*b^13*c^9*d^3*f^2 + 8*
A^2*a^13*b*c^5*d^7*f^2 + 8*A^2*a^7*b^7*c^11*d*f^2 + 24*C^2*a*b^13*c^11*d*f^

2 - 16*C^2*a^13*b*c*d^11*f^2 - 40*B^2*a*b^13*c^11*d*f^2 + 24*B^2*a^13*b*c*d
^11*f^2 + 16*B^2*a*b^13*c*d^11*f^2 - 48*A^2*a*b^13*c*d^11*f^2 - 40*A^2*a^13
*b*c*d^11*f^2 + 24*A^2*a*b^13*c^11*d*f^2 - 6*A*C*b^14*c^12*f^2 + 2*A*C*a^14
*d^12*f^2 + 31*C^2*b^14*c^8*d^4*f^2 + 20*C^2*b^14*c^6*d^6*f^2 + 4*C^2*b^14*
c^4*d^8*f^2 + 2*C^2*b^14*c^10*d^2*f^2 + 80*B^2*b^14*c^6*d^6*f^2 + 64*B^2*b^
14*c^4*d^8*f^2 + 31*B^2*b^14*c^8*d^4*f^2 + 16*B^2*b^14*c^2*d^10*f^2 + 14*C^
2*a^14*c^2*d^10*f^2 + 14*B^2*b^14*c^10*d^2*f^2 - C^2*a^14*c^4*d^8*f^2 + 120
*A^2*b^14*c^2*d^10*f^2 + 112*A^2*b^14*c^4*d^8*f^2 + 33*C^2*a^12*b^2*d^12*f^
2 - 27*C^2*a^10*b^4*d^12*f^2 - 17*A^2*b^14*c^8*d^4*f^2 - 10*B^2*a^14*c^2*d^
10*f^2 - 10*A^2*b^14*c^10*d^2*f^2 + 8*A^2*b^14*c^6*d^6*f^2 + 3*C^2*a^8*b^6*
d^12*f^2 + 3*B^2*a^14*c^4*d^8*f^2 + 117*B^2*a^10*b^4*d^12*f^2 + 111*B^2*a^8
*b^6*d^12*f^2 + 72*B^2*a^6*b^8*d^12*f^2 + 33*C^2*a^4*b^10*c^12*f^2 - 27*C^2
*a^2*b^12*c^12*f^2 + 24*B^2*a^4*b^10*d^12*f^2 + 14*A^2*a^14*c^2*d^10*f^2 +
4*B^2*a^2*b^12*d^12*f^2 - 3*B^2*a^12*b^2*d^12*f^2 - C^2*a^6*b^8*c^12*f^2 -
A^2*a^14*c^4*d^8*f^2 + 720*A^2*a^6*b^8*d^12*f^2 + 552*A^2*a^4*b^10*d^12*f^2
 + 471*A^2*a^8*b^6*d^12*f^2 + 216*A^2*a^2*b^12*d^12*f^2 + 93*A^2*a^10*b^4*d
^12*f^2 + 33*B^2*a^2*b^12*c^12*f^2 + 33*A^2*a^12*b^2*d^12*f^2 - 27*B^2*a^4*
b^10*c^12*f^2 + 3*B^2*a^6*b^8*c^12*f^2 + 33*A^2*a^4*b^10*c^12*f^2 - 27*A^2*
a^2*b^12*c^12*f^2 - A^2*a^6*b^8*c^12*f^2 + 3*C^2*b^14*c^12*f^2 - C^2*a^14*d
^12*f^2 + 36*A^2*b^14*d^12*f^2 + 3*B^2*a^14*d^12*f^2 - B^2*b^14*c^12*f^2 +
3*A^2*b^14*c^12*f^2 - A^2*a^14*d^12*f^2 - 44*A*B*C*a^10*b*c*d^9*f + 3816*A*
B*C*a^4*b^7*c^5*d^5*f + 2920*A*B*C*a^5*b^6*c^2*d^8*f - 2736*A*B*C*a^6*b^5*c
^3*d^7*f - 2672*A*B*C*a^3*b^8*c^4*d^6*f + 1996*A*B*C*a^7*b^4*c^4*d^6*f - 14
12*A*B*C*a^5*b^6*c^6*d^4*f + 1120*A*B*C*a^2*b^9*c^3*d^7*f + 1080*A*B*C*a^7*
b^4*c^2*d^8*f + 1040*A*B*C*a^2*b^9*c^5*d^5*f + 684*A*B*C*a^5*b^6*c^4*d^6*f
+ 592*A*B*C*a^4*b^7*c^3*d^7*f - 560*A*B*C*a^2*b^9*c^7*d^3*f - 448*A*B*C*a^3
*b^8*c^2*d^8*f - 400*A*B*C*a^8*b^3*c^5*d^5*f - 398*A*B*C*a^9*b^2*c^2*d^8*f
- 312*A*B*C*a^3*b^8*c^6*d^4*f + 166*A*B*C*a^3*b^8*c^8*d^2*f + 136*A*B*C*a^6
*b^5*c^5*d^5*f + 128*A*B*C*a^6*b^5*c^7*d^3*f - 100*A*B*C*a^7*b^4*c^6*d^4*f
- 64*A*B*C*a^9*b^2*c^4*d^6*f + 64*A*B*C*a^4*b^7*c^7*d^3*f - 32*A*B*C*a^8*b^
3*c^3*d^7*f - 16*A*B*C*a^5*b^6*c^8*d^2*f - 1312*A*B*C*a^4*b^7*c*d^9*f + 996
*A*B*C*a^8*b^3*c*d^9*f + 728*A*B*C*a*b^10*c^6*d^4*f - 624*A*B*C*a^6*b^5*c*d
^9*f - 584*A*B*C*a*b^10*c^2*d^8*f - 512*A*B*C*a*b^10*c^4*d^6*f - 320*A*B*C*
a^2*b^9*c*d^9*f - 98*A*B*C*a*b^10*c^8*d^2*f + 36*A*B*C*a^2*b^9*c^9*d*f + 32
*A*B*C*a^10*b*c^3*d^7*f - 16*A*B*C*a^4*b^7*c^9*d*f + 46*B*C^2*a^10*b*c*d^9*
f - 16*B^2*C*a*b^10*c*d^9*f - 2*B^2*C*a*b^10*c^9*d*f + 312*A^2*C*a*b^10*c*d
^9*f - 48*A*C^2*a*b^10*c*d^9*f - 6*A^2*C*a*b^10*c^9*d*f + 6*A*C^2*a*b^10*c^
9*d*f + 208*A*B^2*a*b^10*c*d^9*f - 2*A^2*B*a^10*b*c*d^9*f + 2*A*B^2*a*b^10*
c^9*d*f - 224*A*B*C*b^11*c^5*d^5*f + 80*A*B*C*b^11*c^7*d^3*f - 32*A*B*C*b^1
1*c^3*d^7*f + 2*A*B*C*a^11*c^2*d^8*f - 480*A*B*C*a^7*b^4*d^10*f + 78*A*B*C*
a^9*b^2*d^10*f - 64*A*B*C*a^5*b^6*d^10*f + 2*A*B*C*a^3*b^8*c^10*f - 1692*B*
C^2*a^4*b^7*c^5*d^5*f - 1500*B^2*C*a^5*b^6*c^5*d^5*f - 1464*B^2*C*a^5*b^6*c
^3*d^7*f + 1426*B*C^2*a^5*b^6*c^6*d^4*f - 1158*B^2*C*a^4*b^7*c^6*d^4*f + 11
52*B*C^2*a^6*b^5*c^3*d^7*f + 1026*B^2*C*a^6*b^5*c^4*d^6*f - 974*B*C^2*a^7*b
^4*c^4*d^6*f + 960*B^2*C*a^3*b^8*c^5*d^5*f - 884*B*C^2*a^5*b^6*c^2*d^8*f -

764*B^2*C*a^7*b^4*c^5*d^5*f + 752*B^2*C*a^4*b^7*c^2*d^8*f - 752*B*C^2*a^4*b
^7*c^3*d^7*f + 738*B^2*C*a^4*b^7*c^4*d^6*f - 688*B^2*C*a^2*b^9*c^6*d^4*f -
675*B^2*C*a^8*b^3*c^2*d^8*f + 560*B*C^2*a^8*b^3*c^5*d^5*f + 496*B*C^2*a^3*b
^8*c^4*d^6*f + 496*B*C^2*a^2*b^9*c^7*d^3*f - 468*B*C^2*a^7*b^4*c^2*d^8*f +
456*B^2*C*a^3*b^8*c^7*d^3*f - 452*B^2*C*a^8*b^3*c^4*d^6*f - 416*B*C^2*a^2*b
^9*c^3*d^7*f + 378*B*C^2*a^5*b^6*c^4*d^6*f + 376*B*C^2*a^8*b^3*c^3*d^7*f -
360*B^2*C*a^6*b^5*c^2*d^8*f + 355*B*C^2*a^9*b^2*c^2*d^8*f + 346*B^2*C*a^6*b
^5*c^6*d^4*f - 320*B^2*C*a^2*b^9*c^4*d^6*f + 268*B^2*C*a^2*b^9*c^2*d^8*f +
216*B^2*C*a^7*b^4*c^3*d^7*f - 203*B*C^2*a^3*b^8*c^8*d^2*f - 184*B*C^2*a^6*b
^5*c^7*d^3*f + 170*B*C^2*a^7*b^4*c^6*d^4*f + 160*B^2*C*a^5*b^6*c^7*d^3*f -
160*B*C^2*a^2*b^9*c^5*d^5*f - 140*B^2*C*a^4*b^7*c^8*d^2*f - 136*B*C^2*a^3*b
^8*c^2*d^8*f + 112*B^2*C*a^9*b^2*c^3*d^7*f + 91*B^2*C*a^2*b^9*c^8*d^2*f + 8
8*B*C^2*a^4*b^7*c^7*d^3*f + 72*B^2*C*a^8*b^3*c^6*d^4*f - 64*B^2*C*a^3*b^8*c
^3*d^7*f - 60*B*C^2*a^3*b^8*c^6*d^4*f + 56*B*C^2*a^9*b^2*c^4*d^6*f + 52*B*C
^2*a^6*b^5*c^5*d^5*f + 48*B^2*C*a^9*b^2*c^5*d^5*f - 48*B^2*C*a^7*b^4*c^7*d^
3*f + 44*B*C^2*a^5*b^6*c^8*d^2*f - 36*B*C^2*a^9*b^2*c^6*d^4*f + 12*B^2*C*a^
6*b^5*c^8*d^2*f - 2958*A^2*C*a^4*b^7*c^4*d^6*f - 1932*A^2*C*a^4*b^7*c^2*d^8
*f + 1848*A^2*C*a^5*b^6*c^3*d^7*f + 1728*A^2*C*a^3*b^8*c^3*d^7*f + 1524*A^2
*C*a^5*b^6*c^5*d^5*f + 1374*A*C^2*a^4*b^7*c^4*d^6*f - 1272*A*C^2*a^5*b^6*c^
3*d^7*f - 1236*A*C^2*a^5*b^6*c^5*d^5*f + 1116*A*C^2*a^4*b^7*c^2*d^8*f - 111
0*A^2*C*a^6*b^5*c^4*d^6*f + 1038*A*C^2*a^6*b^5*c^4*d^6*f - 768*A^2*C*a^2*b^
9*c^2*d^8*f - 696*A^2*C*a^7*b^4*c^3*d^7*f - 666*A*C^2*a^4*b^7*c^6*d^4*f + 5
64*A^2*C*a^6*b^5*c^2*d^8*f - 564*A*C^2*a^7*b^4*c^5*d^5*f - 555*A*C^2*a^8*b^
3*c^2*d^8*f + 519*A^2*C*a^8*b^3*c^2*d^8*f - 480*A*C^2*a^3*b^8*c^3*d^7*f + 4
56*A*C^2*a^3*b^8*c^5*d^5*f - 420*A*C^2*a^2*b^9*c^6*d^4*f + 408*A*C^2*a^7*b^
4*c^3*d^7*f + 408*A*C^2*a^2*b^9*c^2*d^8*f + 348*A^2*C*a^2*b^9*c^6*d^4*f - 3
48*A*C^2*a^6*b^5*c^2*d^8*f + 342*A*C^2*a^6*b^5*c^6*d^4*f - 336*A*C^2*a^8*b^
3*c^4*d^6*f + 324*A^2*C*a^7*b^4*c^5*d^5*f - 312*A^2*C*a^2*b^9*c^4*d^6*f + 2
64*A^2*C*a^8*b^3*c^4*d^6*f + 240*A*C^2*a^5*b^6*c^7*d^3*f + 195*A*C^2*a^2*b^
9*c^8*d^2*f - 174*A^2*C*a^6*b^5*c^6*d^4*f + 144*A*C^2*a^9*b^2*c^3*d^7*f - 1
23*A^2*C*a^2*b^9*c^8*d^2*f + 120*A*C^2*a^3*b^8*c^7*d^3*f + 108*A*C^2*a^8*b^
3*c^6*d^4*f - 102*A^2*C*a^4*b^7*c^6*d^4*f - 96*A^2*C*a^4*b^7*c^8*d^2*f + 72
*A^2*C*a^3*b^8*c^7*d^3*f + 72*A*C^2*a^9*b^2*c^5*d^5*f - 48*A^2*C*a^9*b^2*c^
3*d^7*f + 48*A^2*C*a^5*b^6*c^7*d^3*f - 48*A*C^2*a^2*b^9*c^4*d^6*f - 24*A^2*
C*a^3*b^8*c^5*d^5*f - 12*A*C^2*a^4*b^7*c^8*d^2*f + 2736*A^2*B*a^6*b^5*c^3*d
^7*f + 2464*A^2*B*a^3*b^8*c^4*d^6*f - 2298*A*B^2*a^4*b^7*c^4*d^6*f - 2252*A
^2*B*a^5*b^6*c^2*d^8*f - 1692*A^2*B*a^4*b^7*c^5*d^5*f - 1592*A*B^2*a^4*b^7*
c^2*d^8*f - 1338*A*B^2*a^6*b^5*c^4*d^6*f + 1320*A*B^2*a^5*b^6*c^3*d^7*f + 1
212*A*B^2*a^5*b^6*c^5*d^5*f - 1056*A*B^2*a^3*b^8*c^5*d^5*f + 1024*A^2*B*a^4
*b^7*c^3*d^7*f - 1022*A^2*B*a^7*b^4*c^4*d^6*f - 880*A^2*B*a^2*b^9*c^5*d^5*f
 - 846*A^2*B*a^5*b^6*c^4*d^6*f - 840*A*B^2*a^7*b^4*c^3*d^7*f + 760*A*B^2*a^
2*b^9*c^6*d^4*f - 704*A^2*B*a^2*b^9*c^3*d^7*f + 688*A*B^2*a^3*b^8*c^3*d^7*f
 + 660*A^2*B*a^3*b^8*c^6*d^4*f - 612*A^2*B*a^7*b^4*c^2*d^8*f + 462*A*B^2*a^
4*b^7*c^6*d^4*f + 459*A*B^2*a^8*b^3*c^2*d^8*f - 412*A*B^2*a^2*b^9*c^2*d^8*f
 - 408*A*B^2*a^3*b^8*c^7*d^3*f + 388*A^2*B*a^6*b^5*c^5*d^5*f + 296*A^2*B*a^

3*b^8*c^2*d^8*f + 288*A*B^2*a^6*b^5*c^2*d^8*f + 284*A*B^2*a^7*b^4*c^5*d^5*f
+ 236*A*B^2*a^8*b^3*c^4*d^6*f - 226*A*B^2*a^6*b^5*c^6*d^4*f + 212*A*B^2*a^
2*b^9*c^4*d^6*f + 202*A^2*B*a^5*b^6*c^6*d^4*f - 152*A^2*B*a^4*b^7*c^7*d^3*f
+ 88*A^2*B*a^8*b^3*c^3*d^7*f + 79*A^2*B*a^9*b^2*c^2*d^8*f - 70*A^2*B*a^7*b
^4*c^6*d^4*f + 68*A*B^2*a^4*b^7*c^8*d^2*f + 64*A^2*B*a^2*b^9*c^7*d^3*f - 64
*A*B^2*a^9*b^2*c^3*d^7*f + 56*A^2*B*a^8*b^3*c^5*d^5*f + 56*A^2*B*a^6*b^5*c^
7*d^3*f + 37*A^2*B*a^3*b^8*c^8*d^2*f - 28*A^2*B*a^9*b^2*c^4*d^6*f - 28*A^2*
B*a^5*b^6*c^8*d^2*f + 17*A*B^2*a^2*b^9*c^8*d^2*f - 16*A*B^2*a^5*b^6*c^7*d^3
*f + 48*A*B*C*b^11*c*d^9*f + 4*A*B*C*b^11*c^9*d*f + 24*A*B*C*a*b^10*d^10*f
- 6*A*B*C*a*b^10*c^10*f + 432*B^2*C*a^7*b^4*c*d^9*f - 376*B*C^2*a*b^10*c^6*
d^4*f - 354*B*C^2*a^8*b^3*c*d^9*f + 352*B^2*C*a*b^10*c^5*d^5*f + 320*B^2*C*
a^5*b^6*c*d^9*f + 256*B^2*C*a*b^10*c^3*d^7*f - 232*B^2*C*a*b^10*c^7*d^3*f -
210*B^2*C*a^9*b^2*c*d^9*f - 152*B*C^2*a*b^10*c^4*d^6*f + 85*B*C^2*a*b^10*c
^8*d^2*f + 72*B^2*C*a^3*b^8*c*d^9*f - 48*B*C^2*a^6*b^5*c*d^9*f - 40*B*C^2*a
^10*b*c^3*d^7*f + 40*B*C^2*a*b^10*c^2*d^8*f + 37*B^2*C*a^10*b*c^2*d^8*f + 2
2*B^2*C*a^3*b^8*c^9*d*f - 18*B*C^2*a^2*b^9*c^9*d*f + 16*B*C^2*a^2*b^9*c*d^9
*f - 12*B^2*C*a^10*b*c^4*d^6*f + 8*B*C^2*a^4*b^7*c^9*d*f + 8*B*C^2*a^4*b^7*
c*d^9*f - 984*A^2*C*a^7*b^4*c*d^9*f + 672*A^2*C*a^3*b^8*c*d^9*f + 552*A*C^2
*a^7*b^4*c*d^9*f - 504*A^2*C*a*b^10*c^5*d^5*f - 408*A^2*C*a^5*b^6*c*d^9*f +
408*A*C^2*a^5*b^6*c*d^9*f + 336*A*C^2*a*b^10*c^5*d^5*f - 216*A*C^2*a*b^10*
c^7*d^3*f + 192*A*C^2*a*b^10*c^3*d^7*f - 162*A*C^2*a^9*b^2*c*d^9*f + 120*A^
2*C*a*b^10*c^7*d^3*f + 96*A^2*C*a*b^10*c^3*d^7*f + 90*A^2*C*a^9*b^2*c*d^9*f
+ 66*A^2*C*a^3*b^8*c^9*d*f - 66*A*C^2*a^3*b^8*c^9*d*f + 57*A*C^2*a^10*b*c^
2*d^8*f - 48*A*C^2*a^3*b^8*c*d^9*f - 9*A^2*C*a^10*b*c^2*d^8*f + 1736*A^2*B*
a^4*b^7*c*d^9*f + 1248*A^2*B*a^6*b^5*c*d^9*f - 1008*A*B^2*a^7*b^4*c*d^9*f +
772*A^2*B*a*b^10*c^4*d^6*f - 688*A*B^2*a*b^10*c^5*d^5*f - 608*A*B^2*a^5*b^
6*c*d^9*f + 436*A^2*B*a*b^10*c^2*d^8*f - 426*A^2*B*a^8*b^3*c*d^9*f + 312*A*
B^2*a^3*b^8*c*d^9*f + 304*A^2*B*a^2*b^9*c*d^9*f - 244*A^2*B*a*b^10*c^6*d^4*
f - 160*A*B^2*a*b^10*c^3*d^7*f + 114*A*B^2*a^9*b^2*c*d^9*f + 88*A*B^2*a*b^1
0*c^7*d^3*f - 22*A*B^2*a^3*b^8*c^9*d*f - 18*A^2*B*a^2*b^9*c^9*d*f + 13*A^2*
B*a*b^10*c^8*d^2*f - 13*A*B^2*a^10*b*c^2*d^8*f + 8*A^2*B*a^10*b*c^3*d^7*f +
8*A^2*B*a^4*b^7*c^9*d*f + 112*B^2*C*b^11*c^6*d^4*f - 64*B*C^2*b^11*c^7*d^3
*f + 16*B^2*C*b^11*c^4*d^6*f - 16*B^2*C*b^11*c^2*d^8*f + 16*B*C^2*b^11*c^5*
d^5*f + 16*B*C^2*b^11*c^3*d^7*f - B^2*C*b^11*c^8*d^2*f + 96*A^2*C*b^11*c^4*
d^6*f - 84*A^2*C*b^11*c^6*d^4*f + 72*A*C^2*b^11*c^6*d^4*f - 24*A*C^2*b^11*c
^4*d^6*f - 24*A*C^2*b^11*c^2*d^8*f - 21*A*C^2*b^11*c^8*d^2*f + 12*A^2*C*b^1
1*c^2*d^8*f + 9*A^2*C*b^11*c^8*d^2*f - B*C^2*a^11*c^2*d^8*f + 176*A*B^2*b^1
1*c^4*d^6*f + 136*A^2*B*b^11*c^5*d^5*f - 128*A^2*B*b^11*c^3*d^7*f + 112*A*B
^2*b^11*c^2*d^8*f + 111*B^2*C*a^8*b^3*d^10*f - 64*A*B^2*b^11*c^6*d^4*f - 39
*B*C^2*a^9*b^2*d^10*f + 24*B*C^2*a^7*b^4*d^10*f - 16*A^2*B*b^11*c^7*d^3*f -
4*B^2*C*a^2*b^9*d^10*f - 4*B*C^2*a^5*b^6*d^10*f + 432*A^2*C*a^6*b^5*d^10*f
+ 192*A^2*C*a^4*b^7*d^10*f - 111*A^2*C*a^8*b^3*d^10*f + 111*A*C^2*a^8*b^3*
d^10*f - 72*A*C^2*a^6*b^5*d^10*f + 12*A*C^2*a^4*b^7*d^10*f - 3*B^2*C*a^2*b^
9*c^10*f - A^2*B*a^11*c^2*d^8*f - B*C^2*a^3*b^8*c^10*f + 456*A^2*B*a^7*b^4*
d^10*f - 288*A^2*B*a^3*b^8*d^10*f + 252*A*B^2*a^6*b^5*d^10*f + 192*A*B^2*a^

4*b^7*d^10*f - 183*A*B^2*a^8*b^3*d^10*f - 148*A^2*B*a^5*b^6*d^10*f + 76*A*B
^2*a^2*b^9*d^10*f - 9*A^2*C*a^2*b^9*c^10*f + 9*A*C^2*a^2*b^9*c^10*f - 3*A^2
*B*a^9*b^2*d^10*f + 3*A*B^2*a^2*b^9*c^10*f - A^2*B*a^3*b^8*c^10*f - 2*C^3*a
*b^10*c^9*d*f - 2*B^3*a^10*b*c*d^9*f - 264*A^3*a*b^10*c*d^9*f + 2*A^3*a*b^1
0*c^9*d*f - 2*B*C^2*b^11*c^9*d*f - 2*B^2*C*a^11*c*d^9*f - 120*A^2*B*b^11*c*
d^9*f - 9*B^2*C*a^10*b*d^10*f - 6*A^2*C*a^11*c*d^9*f + 6*A*C^2*a^11*c*d^9*f
 - 2*A^2*B*b^11*c^9*d*f + 9*A^2*C*a^10*b*d^10*f - 9*A*C^2*a^10*b*d^10*f + 3
*B*C^2*a*b^10*c^10*f + 2*A*B^2*a^11*c*d^9*f - 132*A^2*B*a*b^10*d^10*f - 3*A
*B^2*a^10*b*d^10*f + 3*A^2*B*a*b^10*c^10*f + 520*C^3*a^5*b^6*c^3*d^7*f + 46
0*C^3*a^5*b^6*c^5*d^5*f - 418*C^3*a^6*b^5*c^4*d^6*f + 406*C^3*a^4*b^7*c^6*d
^4*f + 268*C^3*a^7*b^4*c^5*d^5*f - 266*C^3*a^6*b^5*c^6*d^4*f + 233*C^3*a^8*
b^3*c^2*d^8*f - 176*C^3*a^5*b^6*c^7*d^3*f + 164*C^3*a^2*b^9*c^6*d^4*f + 140
*C^3*a^6*b^5*c^2*d^8*f + 136*C^3*a^2*b^9*c^4*d^6*f - 128*C^3*a^9*b^2*c^3*d^
7*f + 128*C^3*a^3*b^8*c^3*d^7*f - 108*C^3*a^8*b^3*c^6*d^4*f - 104*C^3*a^3*b
^8*c^7*d^3*f - 104*C^3*a^3*b^8*c^5*d^5*f + 100*C^3*a^8*b^3*c^4*d^6*f - 89*C
^3*a^2*b^9*c^8*d^2*f - 72*C^3*a^9*b^2*c^5*d^5*f - 40*C^3*a^7*b^4*c^3*d^7*f
+ 40*C^3*a^4*b^7*c^8*d^2*f - 28*C^3*a^4*b^7*c^2*d^8*f - 16*C^3*a^2*b^9*c^2*
d^8*f - 2*C^3*a^4*b^7*c^4*d^6*f + 828*B^3*a^4*b^7*c^5*d^5*f + 408*B^3*a^5*b
^6*c^2*d^8*f + 390*B^3*a^7*b^4*c^4*d^6*f - 372*B^3*a^3*b^8*c^4*d^6*f - 336*
B^3*a^6*b^5*c^3*d^7*f - 314*B^3*a^5*b^6*c^6*d^4*f + 288*B^3*a^4*b^7*c^3*d^7
*f + 216*B^3*a^7*b^4*c^2*d^8*f - 176*B^3*a^2*b^9*c^7*d^3*f + 128*B^3*a^2*b^
9*c^3*d^7*f + 108*B^3*a^6*b^5*c^5*d^5*f + 88*B^3*a^4*b^7*c^7*d^3*f + 72*B^3
*a^2*b^9*c^5*d^5*f - 68*B^3*a^3*b^8*c^2*d^8*f - 65*B^3*a^9*b^2*c^2*d^8*f -
56*B^3*a^8*b^3*c^5*d^5*f + 40*B^3*a^6*b^5*c^7*d^3*f + 37*B^3*a^3*b^8*c^8*d^
2*f + 30*B^3*a^5*b^6*c^4*d^6*f - 28*B^3*a^5*b^6*c^8*d^2*f + 24*B^3*a^8*b^3*
c^3*d^7*f - 4*B^3*a^9*b^2*c^4*d^6*f - 2*B^3*a^7*b^4*c^6*d^4*f + 1586*A^3*a^
4*b^7*c^4*d^6*f - 1376*A^3*a^3*b^8*c^3*d^7*f - 1096*A^3*a^5*b^6*c^3*d^7*f +
 844*A^3*a^4*b^7*c^2*d^8*f - 748*A^3*a^5*b^6*c^5*d^5*f + 490*A^3*a^6*b^5*c^
4*d^6*f + 376*A^3*a^2*b^9*c^2*d^8*f + 362*A^3*a^4*b^7*c^6*d^4*f - 356*A^3*a
^6*b^5*c^2*d^8*f + 328*A^3*a^7*b^4*c^3*d^7*f - 328*A^3*a^3*b^8*c^5*d^5*f +
224*A^3*a^2*b^9*c^4*d^6*f - 197*A^3*a^8*b^3*c^2*d^8*f - 112*A^3*a^5*b^6*c^7
*d^3*f + 98*A^3*a^6*b^5*c^6*d^4*f - 92*A^3*a^2*b^9*c^6*d^4*f - 88*A^3*a^3*b
^8*c^7*d^3*f + 68*A^3*a^4*b^7*c^8*d^2*f + 32*A^3*a^9*b^2*c^3*d^7*f - 28*A^3
*a^8*b^3*c^4*d^6*f - 28*A^3*a^7*b^4*c^5*d^5*f + 17*A^3*a^2*b^9*c^8*d^2*f +
104*C^3*a*b^10*c^7*d^3*f + 54*C^3*a^9*b^2*c*d^9*f - 40*C^3*a^7*b^4*c*d^9*f
- 35*C^3*a^10*b*c^2*d^8*f + 22*C^3*a^3*b^8*c^9*d*f + 16*C^3*a*b^10*c^5*d^5*
f - 16*C^3*a*b^10*c^3*d^7*f + 8*C^3*a^5*b^6*c*d^9*f - 2*A*B*C*a^11*d^10*f +
 198*B^3*a^8*b^3*c*d^9*f + 192*B^3*a*b^10*c^6*d^4*f - 128*B^3*a^4*b^7*c*d^9
*f - 80*B^3*a*b^10*c^2*d^8*f - 56*B^3*a^2*b^9*c*d^9*f - 24*B^3*a^6*b^5*c*d^
9*f - 18*B^3*a^2*b^9*c^9*d*f - 16*B^3*a*b^10*c^4*d^6*f + 13*B^3*a*b^10*c^8*
d^2*f + 8*B^3*a^10*b*c^3*d^7*f + 8*B^3*a^4*b^7*c^9*d*f - 624*A^3*a^3*b^8*c*
d^9*f + 472*A^3*a^7*b^4*c*d^9*f - 272*A^3*a*b^10*c^3*d^7*f + 152*A^3*a*b^10
*c^5*d^5*f - 22*A^3*a^3*b^8*c^9*d*f + 18*A^3*a^9*b^2*c*d^9*f - 13*A^3*a^10*
b*c^2*d^8*f - 8*A^3*a^5*b^6*c*d^9*f - 8*A^3*a*b^10*c^7*d^3*f + A*B^2*b^11*c
^8*d^2*f + 11*C^3*b^11*c^8*d^2*f - 8*C^3*b^11*c^6*d^4*f - 4*C^3*b^11*c^4*d^

6*f - 64*B^3*b^11*c^5*d^5*f - 32*B^3*b^11*c^3*d^7*f - 68*A^3*b^11*c^4*d^6*f
+ 20*A^3*b^11*c^6*d^4*f + 12*A^3*b^11*c^2*d^8*f - C^3*a^8*b^3*d^10*f - B^3
*a^11*c^2*d^8*f - 60*B^3*a^7*b^4*d^10*f - 32*B^3*a^5*b^6*d^10*f + 21*B^3*a^
9*b^2*d^10*f - 12*B^3*a^3*b^8*d^10*f - 3*C^3*a^2*b^9*c^10*f - 360*A^3*a^6*b
^5*d^10*f - 204*A^3*a^4*b^7*d^10*f - B^3*a^3*b^8*c^10*f + 3*A^3*a^2*b^9*c^1
0*f - 2*C^3*a^11*c*d^9*f - 2*B^3*b^11*c^9*d*f + 3*C^3*a^10*b*d^10*f + 2*A^3
*a^11*c*d^9*f + 3*B^3*a*b^10*c^10*f - 3*A^3*a^10*b*d^10*f - 36*A^2*C*b^11*d
^10*f + 3*A^2*C*b^11*c^10*f - 3*A*C^2*b^11*c^10*f - A*B^2*b^11*c^10*f + 36*
A^3*b^11*d^10*f - A^3*b^11*c^10*f + A^3*b^11*c^8*d^2*f + A^3*a^8*b^3*d^10*f
+ B^2*C*b^11*c^10*f + B*C^2*a^11*d^10*f + A^2*B*a^11*d^10*f + C^3*b^11*c^1
0*f + B^3*a^11*d^10*f - 6*A*B^2*C*a^7*b*c*d^7 + 4*A*B^2*C*a*b^7*c*d^7 + 168
*A^2*B*C*a^2*b^6*c^3*d^5 + 144*A*B*C^2*a^3*b^5*c^4*d^4 - 129*A^2*B*C*a^3*b^
5*c^4*d^4 - 96*A*B*C^2*a^2*b^6*c^3*d^5 + 84*A*B*C^2*a^3*b^5*c^2*d^6 + 72*A^
2*B*C*a^4*b^4*c^3*d^5 - 72*A^2*B*C*a^3*b^5*c^2*d^6 + 64*A*B^2*C*a^4*b^4*c^4
*d^4 - 60*A*B*C^2*a^4*b^4*c^3*d^5 + 57*A^2*B*C*a^5*b^3*c^2*d^6 - 56*A*B^2*C
*a^5*b^3*c^3*d^5 - 39*A*B^2*C*a^2*b^6*c^4*d^4 - 38*A*B^2*C*a^3*b^5*c^5*d^3
+ 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^5*b^3*c^4*d^4 - 30*A*B*C^2*a^5*
b^3*c^2*d^6 + 27*A*B^2*C*a^6*b^2*c^2*d^6 - 24*A*B^2*C*a^2*b^6*c^2*d^6 + 24*
A*B*C^2*a^6*b^2*c^3*d^5 - 24*A*B*C^2*a^4*b^4*c^5*d^3 - 18*A^2*B*C*a^5*b^3*c
^4*d^4 + 18*A^2*B*C*a^2*b^6*c^5*d^3 - 15*A*B^2*C*a^4*b^4*c^2*d^6 - 12*A^2*B
*C*a^6*b^2*c^3*d^5 + 12*A^2*B*C*a^4*b^4*c^5*d^3 + 9*A*B^2*C*a^2*b^6*c^6*d^2
+ 6*A*B*C^2*a^3*b^5*c^6*d^2 - 3*A^2*B*C*a^3*b^5*c^6*d^2 + 60*A^2*B*C*a^2*b
^6*c*d^7 - 51*A^2*B*C*a*b^7*c^4*d^4 + 48*A*B*C^2*a^6*b^2*c*d^7 - 42*A^2*B*C
*a^6*b^2*c*d^7 - 42*A^2*B*C*a*b^7*c^2*d^6 + 36*A*B*C^2*a^4*b^4*c*d^7 + 36*A
*B*C^2*a*b^7*c^4*d^4 + 36*A*B*C^2*a*b^7*c^2*d^6 - 30*A^2*B*C*a^4*b^4*c*d^7
+ 24*A*B^2*C*a^3*b^5*c*d^7 - 24*A*B*C^2*a^2*b^6*c*d^7 + 18*A*B^2*C*a*b^7*c^
5*d^3 - 18*A*B*C^2*a*b^7*c^6*d^2 + 12*A*B^2*C*a*b^7*c^3*d^5 + 9*A^2*B*C*a*b
^7*c^6*d^2 + 6*A*B^2*C*a^5*b^3*c*d^7 - 6*A*B*C^2*a^7*b*c^2*d^6 + 3*A^2*B*C*
a^7*b*c^2*d^6 - 18*B^3*C*a^6*b^2*c*d^7 - 18*B*C^3*a^6*b^2*c*d^7 - 14*B^3*C*
a^4*b^4*c*d^7 - 14*B*C^3*a^4*b^4*c*d^7 - 10*B^3*C*a*b^7*c^2*d^6 - 10*B*C^3*
a*b^7*c^2*d^6 + 9*B^3*C*a*b^7*c^6*d^2 + 9*B*C^3*a*b^7*c^6*d^2 - 7*B^3*C*a*b
^7*c^4*d^4 - 7*B*C^3*a*b^7*c^4*d^4 + 6*B^2*C^2*a^7*b*c*d^7 - 4*B^3*C*a^2*b^
6*c*d^7 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a^2*b^6*c*d^7 + 3*B^3*C*a^7*b*c^2
*d^6 + 3*B*C^3*a^7*b*c^2*d^6 + 144*A^3*C*a^3*b^5*c*d^7 + 62*A^3*C*a^5*b^3*c
*d^7 + 48*A*C^3*a^3*b^5*c*d^7 - 36*A^2*C^2*a*b^7*c*d^7 + 26*A*C^3*a^5*b^3*c
*d^7 + 20*A^3*C*a*b^7*c^3*d^5 + 18*A^2*C^2*a^7*b*c*d^7 - 18*A*C^3*a*b^7*c^5
*d^3 - 6*A^3*C*a*b^7*c^5*d^3 - 4*A*C^3*a*b^7*c^3*d^5 - 32*A^3*B*a^2*b^6*c*d
^7 - 32*A*B^3*a^2*b^6*c*d^7 + 22*A^3*B*a*b^7*c^4*d^4 + 22*A*B^3*a*b^7*c^4*d
^4 + 16*A^3*B*a*b^7*c^2*d^6 + 16*A*B^3*a*b^7*c^2*d^6 + 12*A^3*B*a^6*b^2*c*d
^7 + 12*A*B^3*a^6*b^2*c*d^7 + 8*A^3*B*a^4*b^4*c*d^7 - 8*A^2*B^2*a*b^7*c*d^7
+ 8*A*B^3*a^4*b^4*c*d^7 + 36*A^2*B*C*b^8*c^3*d^5 + 24*A*B*C^2*b^8*c^5*d^3
- 18*A^2*B*C*b^8*c^5*d^3 - 12*A*B*C^2*b^8*c^3*d^5 - 3*A*B^2*C*b^8*c^6*d^2 -
3*A*B^2*C*b^8*c^4*d^4 - 2*A*B^2*C*b^8*c^2*d^6 + 57*A^2*B*C*a^5*b^3*d^8 + 3
6*A^2*B*C*a^3*b^5*d^8 - 30*A*B*C^2*a^5*b^3*d^8 - 18*A*B*C^2*a^3*b^5*d^8 - 9
*A*B^2*C*a^4*b^4*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 2*A*B^2*C*a^2*b^6*d^8 + 34*B

^2*C^2*a^3*b^5*c^5*d^3 + 28*B^2*C^2*a^5*b^3*c^3*d^5 + 24*B^2*C^2*a^2*b^6*c^
4*d^4 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*d^5 + 12*B^2*C^
2*a^2*b^6*c^2*d^6 + 9*B^2*C^2*a^6*b^2*c^4*d^4 + 9*B^2*C^2*a^4*b^4*c^2*d^6 -
 9*B^2*C^2*a^2*b^6*c^6*d^2 - 3*B^2*C^2*a^6*b^2*c^2*d^6 + 159*A^2*C^2*a^4*b^
4*c^2*d^6 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^3*b^5*c^5*d^3 + 78*A
^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2*C^2*a^6*b^2*c^
2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^2*b^6*c^4*d^4 + 9*A^2*C^2
*a^6*b^2*c^4*d^4 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a^4*b^4*c^2*d^6
- 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^2*b^6*c^4*d^4 + 28*A^2*B^2*a^5*
b^3*c^3*d^5 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^6*b^2*c^2*d^6 + 4*A^
2*B^2*a^3*b^5*c^5*d^3 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a^7*b*c*d^7 + 12*A*
C^3*a*b^7*c*d^7 - 6*A^3*C*a^7*b*c*d^7 + 24*A^2*B*C*b^8*c*d^7 - 12*A*B*C^2*b
^8*c*d^7 + 12*A^2*B*C*a*b^7*d^8 + 6*A*B*C^2*a^7*b*d^8 - 6*A*B*C^2*a*b^7*d^8
 - 3*A^2*B*C*a^7*b*d^8 - 53*B^3*C*a^3*b^5*c^4*d^4 - 53*B*C^3*a^3*b^5*c^4*d^
4 - 32*B^3*C*a^3*b^5*c^2*d^6 - 32*B*C^3*a^3*b^5*c^2*d^6 - 18*B^3*C*a^5*b^3*
c^4*d^4 - 18*B*C^3*a^5*b^3*c^4*d^4 + 16*B^3*C*a^4*b^4*c^3*d^5 + 16*B*C^3*a^
4*b^4*c^3*d^5 - 12*B^3*C*a^6*b^2*c^3*d^5 + 12*B^3*C*a^4*b^4*c^5*d^3 + 12*B^
2*C^2*a^3*b^5*c*d^7 - 12*B*C^3*a^6*b^2*c^3*d^5 + 12*B*C^3*a^4*b^4*c^5*d^3 +
 8*B^3*C*a^2*b^6*c^3*d^5 + 8*B*C^3*a^2*b^6*c^3*d^5 - 6*B^3*C*a^2*b^6*c^5*d^
3 + 6*B^2*C^2*a^5*b^3*c*d^7 - 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*C^3*a^2*b^6*c^5
*d^3 - 3*B^3*C*a^3*b^5*c^6*d^2 - 3*B*C^3*a^3*b^5*c^6*d^2 - 175*A^3*C*a^4*b^
4*c^2*d^6 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a^3*b^5*c*d^7 - 124*A^3
*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^3*b^5*c^5*d^3 - 73*A*C^3*a^4*b^4*c^2*d^6 -
66*A^2*C^2*a^5*b^3*c*d^7 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C^3*a^4*b^4*c^4*
d^4 + 30*A^3*C*a^4*b^4*c^4*d^4 - 30*A^3*C*a^3*b^5*c^5*d^3 + 27*A*C^3*a^2*b^
6*c^6*d^2 + 21*A*C^3*a^2*b^6*c^4*d^4 + 18*A^2*C^2*a*b^7*c^5*d^3 - 18*A*C^3*
a^6*b^2*c^4*d^4 - 16*A*C^3*a^2*b^6*c^2*d^6 + 15*A^3*C*a^6*b^2*c^2*d^6 - 15*
A^3*C*a^2*b^6*c^4*d^4 - 12*A^2*C^2*a*b^7*c^3*d^5 + 9*A^3*C*a^2*b^6*c^6*d^2
+ 9*A*C^3*a^6*b^2*c^2*d^6 - 80*A^3*B*a^2*b^6*c^3*d^5 - 80*A*B^3*a^2*b^6*c^3
*d^5 + 38*A^3*B*a^3*b^5*c^4*d^4 + 38*A*B^3*a^3*b^5*c^4*d^4 - 36*A^2*B^2*a^3
*b^5*c*d^7 - 28*A^3*B*a^5*b^3*c^2*d^6 - 28*A^3*B*a^4*b^4*c^3*d^5 - 28*A*B^3
*a^5*b^3*c^2*d^6 - 28*A*B^3*a^4*b^4*c^3*d^5 + 20*A^3*B*a^3*b^5*c^2*d^6 + 20
*A*B^3*a^3*b^5*c^2*d^6 - 12*A^3*B*a^2*b^6*c^5*d^3 - 12*A^2*B^2*a^5*b^3*c*d^
7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A^2*B^2*a*b^7*c^3*d^5 - 12*A*B^3*a^2*b^6*
c^5*d^3 + 9*B^2*C^2*b^8*c^4*d^4 + 4*B^2*C^2*b^8*c^2*d^6 + 3*B^2*C^2*b^8*c^6
*d^2 - 30*A^2*C^2*b^8*c^4*d^4 + 9*A^2*C^2*b^8*c^6*d^2 + 16*A^2*B^2*b^8*c^2*
d^6 + 6*B^2*C^2*a^6*b^2*d^8 + 3*B^2*C^2*a^4*b^4*d^8 + 3*A^2*B^2*b^8*c^4*d^4
 + 36*A^2*C^2*a^4*b^4*d^8 + 27*A^2*C^2*a^2*b^6*d^8 - 18*A^2*C^2*a^6*b^2*d^8
 + 33*A^2*B^2*a^4*b^4*d^8 + 28*A^2*B^2*a^2*b^6*d^8 + 6*A^2*B^2*a^6*b^2*d^8
+ 6*C^4*a*b^7*c^5*d^3 + 4*C^4*a*b^7*c^3*d^5 - 2*C^4*a^5*b^3*c*d^7 + 12*B^4*
a^3*b^5*c*d^7 - 12*B^4*a*b^7*c^5*d^3 + 8*B^4*a^5*b^3*c*d^7 - 4*B^4*a*b^7*c^
3*d^5 - 48*A^4*a^3*b^5*c*d^7 - 20*A^4*a^5*b^3*c*d^7 - 8*A^4*a*b^7*c^3*d^5 -
 10*B^3*C*b^8*c^5*d^3 - 10*B*C^3*b^8*c^5*d^3 - 4*B^3*C*b^8*c^3*d^5 - 4*B*C^
3*b^8*c^3*d^5 + 23*A^3*C*b^8*c^4*d^4 - 18*A^3*C*b^8*c^2*d^6 + 11*A*C^3*b^8*
c^4*d^4 - 9*A*C^3*b^8*c^6*d^2 + 6*A*C^3*b^8*c^2*d^6 - 3*A^3*C*b^8*c^6*d^2 -

20*A^3*B*b^8*c^3*d^5 - 20*A*B^3*b^8*c^3*d^5 + 4*A^3*B*b^8*c^5*d^3 + 4*A*B^
3*b^8*c^5*d^3 - 63*A^3*C*a^4*b^4*d^8 - 54*A^3*C*a^2*b^6*d^8 + 9*A^3*C*a^6*b
^2*d^8 + 9*A*C^3*a^6*b^2*d^8 - 3*A*C^3*a^4*b^4*d^8 - 28*A^3*B*a^5*b^3*d^8 -
28*A*B^3*a^5*b^3*d^8 - 18*A^3*B*a^3*b^5*d^8 - 18*A*B^3*a^3*b^5*d^8 + B^3*C
*a^5*b^3*c^2*d^6 + B*C^3*a^5*b^3*c^2*d^6 + 6*C^4*a^7*b*c*d^7 + 4*B^4*a*b^7*
c*d^7 - 12*A^4*a*b^7*c*d^7 - 12*A^3*B*b^8*c*d^7 - 12*A*B^3*b^8*c*d^7 - 3*B^
3*C*a^7*b*d^8 - 3*B*C^3*a^7*b*d^8 - 6*A^3*B*a*b^7*d^8 - 6*A*B^3*a*b^7*d^8 +
30*C^4*a^3*b^5*c^5*d^3 + 19*C^4*a^4*b^4*c^2*d^6 + 9*C^4*a^6*b^2*c^4*d^4 -
9*C^4*a^2*b^6*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*b^6*c^2*d^6 + 3*C
^4*a^6*b^2*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^2*b^6*c^4*d^4 + 28*B^4
*a^5*b^3*c^3*d^5 + 27*B^4*a^2*b^6*c^4*d^4 - 17*B^4*a^4*b^4*c^4*d^4 - 10*B^4
*a^4*b^4*c^2*d^6 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c^2*d^6 - 6*B^4*a^
6*b^2*c^2*d^6 + 4*B^4*a^3*b^5*c^5*d^3 + 70*A^4*a^4*b^4*c^2*d^6 + 58*A^4*a^2
*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^2*b^6*c^4*d^4 + B^2*C^2*a^
2*b^6*d^8 - 18*A^3*C*b^8*d^8 + B^3*C*a^5*b^3*d^8 + B*C^3*a^5*b^3*d^8 + 3*C^
4*b^8*c^6*d^2 + 8*B^4*b^8*c^4*d^4 + 4*B^4*b^8*c^2*d^6 + 12*A^4*b^8*c^2*d^6
- 5*A^4*b^8*c^4*d^4 + 6*B^4*a^6*b^2*d^8 + 3*B^4*a^4*b^4*d^8 + 30*A^4*a^4*b^
4*d^8 + 27*A^4*a^2*b^6*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2*b^8*d^8 + 9*A^4*
b^8*d^8 + C^4*b^8*c^4*d^4 + B^4*a^2*b^6*d^8, f, k)*(root(640*a^13*b^7*c*d^1
5*f^4 + 640*a^7*b^13*c^15*d*f^4 + 480*a^15*b^5*c*d^15*f^4 + 480*a^11*b^9*c*
d^15*f^4 + 480*a^9*b^11*c^15*d*f^4 + 480*a^5*b^15*c^15*d*f^4 + 192*a^19*b*c
^5*d^11*f^4 + 192*a^17*b^3*c*d^15*f^4 + 192*a^11*b^9*c^15*d*f^4 + 192*a^9*b
^11*c*d^15*f^4 + 192*a^3*b^17*c^15*d*f^4 + 192*a*b^19*c^11*d^5*f^4 + 128*a^
19*b*c^7*d^9*f^4 + 128*a^19*b*c^3*d^13*f^4 + 128*a*b^19*c^13*d^3*f^4 + 128*
a*b^19*c^9*d^7*f^4 + 32*a^19*b*c^9*d^7*f^4 + 32*a^13*b^7*c^15*d*f^4 + 32*a^
7*b^13*c*d^15*f^4 + 32*a*b^19*c^7*d^9*f^4 + 32*a^19*b*c*d^15*f^4 + 32*a*b^1
9*c^15*d*f^4 - 47088*a^10*b^10*c^8*d^8*f^4 + 42432*a^11*b^9*c^7*d^9*f^4 + 4
2432*a^9*b^11*c^9*d^7*f^4 + 39328*a^11*b^9*c^9*d^7*f^4 + 39328*a^9*b^11*c^7
*d^9*f^4 - 36912*a^12*b^8*c^8*d^8*f^4 - 36912*a^8*b^12*c^8*d^8*f^4 - 34256*
a^10*b^10*c^10*d^6*f^4 - 34256*a^10*b^10*c^6*d^10*f^4 - 31152*a^12*b^8*c^6*
d^10*f^4 - 31152*a^8*b^12*c^10*d^6*f^4 + 28128*a^13*b^7*c^7*d^9*f^4 + 28128
*a^7*b^13*c^9*d^7*f^4 + 24160*a^11*b^9*c^5*d^11*f^4 + 24160*a^9*b^11*c^11*d
^5*f^4 - 23088*a^12*b^8*c^10*d^6*f^4 - 23088*a^8*b^12*c^6*d^10*f^4 + 22272*
a^13*b^7*c^9*d^7*f^4 + 22272*a^7*b^13*c^7*d^9*f^4 + 19072*a^11*b^9*c^11*d^5
*f^4 + 19072*a^9*b^11*c^5*d^11*f^4 + 18624*a^13*b^7*c^5*d^11*f^4 + 18624*a^
7*b^13*c^11*d^5*f^4 - 17328*a^14*b^6*c^8*d^8*f^4 - 17328*a^6*b^14*c^8*d^8*f
^4 - 17232*a^14*b^6*c^6*d^10*f^4 - 17232*a^6*b^14*c^10*d^6*f^4 - 13520*a^12
*b^8*c^4*d^12*f^4 - 13520*a^8*b^12*c^12*d^4*f^4 - 12464*a^10*b^10*c^12*d^4*
f^4 - 12464*a^10*b^10*c^4*d^12*f^4 + 10880*a^15*b^5*c^7*d^9*f^4 + 10880*a^5
*b^15*c^9*d^7*f^4 - 9072*a^14*b^6*c^10*d^6*f^4 - 9072*a^6*b^14*c^6*d^10*f^4
+ 8928*a^13*b^7*c^11*d^5*f^4 + 8928*a^7*b^13*c^5*d^11*f^4 - 8880*a^14*b^6*
c^4*d^12*f^4 - 8880*a^6*b^14*c^12*d^4*f^4 + 8480*a^15*b^5*c^5*d^11*f^4 + 84
80*a^5*b^15*c^11*d^5*f^4 + 7200*a^15*b^5*c^9*d^7*f^4 + 7200*a^5*b^15*c^7*d^
9*f^4 - 6912*a^12*b^8*c^12*d^4*f^4 - 6912*a^8*b^12*c^4*d^12*f^4 + 6400*a^11
*b^9*c^3*d^13*f^4 + 6400*a^9*b^11*c^13*d^3*f^4 + 5920*a^13*b^7*c^3*d^13*f^4

+ 5920*a^7*b^13*c^13*d^3*f^4 - 5392*a^16*b^4*c^6*d^10*f^4 - 5392*a^4*b^16*
c^10*d^6*f^4 - 4428*a^16*b^4*c^8*d^8*f^4 - 4428*a^4*b^16*c^8*d^8*f^4 + 4128
*a^11*b^9*c^13*d^3*f^4 + 4128*a^9*b^11*c^3*d^13*f^4 - 3328*a^16*b^4*c^4*d^1
2*f^4 - 3328*a^4*b^16*c^12*d^4*f^4 + 3264*a^15*b^5*c^3*d^13*f^4 + 3264*a^5*
b^15*c^13*d^3*f^4 - 2480*a^12*b^8*c^2*d^14*f^4 - 2480*a^8*b^12*c^14*d^2*f^4
+ 2240*a^15*b^5*c^11*d^5*f^4 + 2240*a^5*b^15*c^5*d^11*f^4 - 2128*a^14*b^6*
c^12*d^4*f^4 - 2128*a^6*b^14*c^4*d^12*f^4 + 2112*a^17*b^3*c^7*d^9*f^4 + 211
2*a^3*b^17*c^9*d^7*f^4 + 2048*a^17*b^3*c^5*d^11*f^4 + 2048*a^3*b^17*c^11*d^
5*f^4 - 2000*a^14*b^6*c^2*d^14*f^4 - 2000*a^6*b^14*c^14*d^2*f^4 - 1792*a^16
*b^4*c^10*d^6*f^4 - 1792*a^4*b^16*c^6*d^10*f^4 - 1776*a^10*b^10*c^14*d^2*f^
4 - 1776*a^10*b^10*c^2*d^14*f^4 + 1472*a^13*b^7*c^13*d^3*f^4 + 1472*a^7*b^1
3*c^3*d^13*f^4 + 1088*a^17*b^3*c^9*d^7*f^4 + 1088*a^3*b^17*c^7*d^9*f^4 + 99
2*a^17*b^3*c^3*d^13*f^4 + 992*a^3*b^17*c^13*d^3*f^4 - 912*a^16*b^4*c^2*d^14
*f^4 - 912*a^4*b^16*c^14*d^2*f^4 - 768*a^18*b^2*c^6*d^10*f^4 - 768*a^2*b^18
*c^10*d^6*f^4 - 688*a^12*b^8*c^14*d^2*f^4 - 688*a^8*b^12*c^2*d^14*f^4 - 592
*a^18*b^2*c^4*d^12*f^4 - 592*a^2*b^18*c^12*d^4*f^4 - 472*a^18*b^2*c^8*d^8*f
^4 - 472*a^2*b^18*c^8*d^8*f^4 - 280*a^16*b^4*c^12*d^4*f^4 - 280*a^4*b^16*c^
4*d^12*f^4 + 224*a^17*b^3*c^11*d^5*f^4 + 224*a^15*b^5*c^13*d^3*f^4 + 224*a^
5*b^15*c^3*d^13*f^4 + 224*a^3*b^17*c^5*d^11*f^4 - 208*a^18*b^2*c^2*d^14*f^4
- 208*a^2*b^18*c^14*d^2*f^4 - 112*a^18*b^2*c^10*d^6*f^4 - 112*a^14*b^6*c^1
4*d^2*f^4 - 112*a^6*b^14*c^2*d^14*f^4 - 112*a^2*b^18*c^6*d^10*f^4 - 24*b^20
*c^12*d^4*f^4 - 16*b^20*c^14*d^2*f^4 - 16*b^20*c^10*d^6*f^4 - 4*b^20*c^8*d^
8*f^4 - 24*a^20*c^4*d^12*f^4 - 16*a^20*c^6*d^10*f^4 - 16*a^20*c^2*d^14*f^4
- 4*a^20*c^8*d^8*f^4 - 80*a^14*b^6*d^16*f^4 - 60*a^16*b^4*d^16*f^4 - 60*a^1
2*b^8*d^16*f^4 - 24*a^18*b^2*d^16*f^4 - 24*a^10*b^10*d^16*f^4 - 4*a^8*b^12*
d^16*f^4 - 80*a^6*b^14*c^16*f^4 - 60*a^8*b^12*c^16*f^4 - 60*a^4*b^16*c^16*f
^4 - 24*a^10*b^10*c^16*f^4 - 24*a^2*b^18*c^16*f^4 - 4*a^12*b^8*c^16*f^4 - 4
*b^20*c^16*f^4 - 4*a^20*d^16*f^4 + 56*A*C*a^13*b*c*d^11*f^2 - 48*A*C*a*b^13
*c^11*d*f^2 + 48*A*C*a*b^13*c*d^11*f^2 + 5904*B*C*a^7*b^7*c^6*d^6*f^2 - 501
6*B*C*a^8*b^6*c^5*d^7*f^2 - 4608*B*C*a^6*b^8*c^7*d^5*f^2 - 4512*B*C*a^6*b^8
*c^5*d^7*f^2 - 4384*B*C*a^8*b^6*c^7*d^5*f^2 + 3056*B*C*a^7*b^7*c^8*d^4*f^2
+ 2256*B*C*a^7*b^7*c^4*d^8*f^2 - 1824*B*C*a^8*b^6*c^3*d^9*f^2 + 1632*B*C*a^
4*b^10*c^9*d^3*f^2 - 1400*B*C*a^3*b^11*c^8*d^4*f^2 - 1320*B*C*a^11*b^3*c^4*
d^8*f^2 - 1248*B*C*a^6*b^8*c^3*d^9*f^2 + 1152*B*C*a^10*b^4*c^3*d^9*f^2 - 10
72*B*C*a^6*b^8*c^9*d^3*f^2 + 1068*B*C*a^9*b^5*c^6*d^6*f^2 - 1004*B*C*a^5*b^
9*c^4*d^8*f^2 - 968*B*C*a^3*b^11*c^6*d^6*f^2 - 864*B*C*a^5*b^9*c^8*d^4*f^2
- 828*B*C*a^9*b^5*c^4*d^8*f^2 - 792*B*C*a^11*b^3*c^2*d^10*f^2 - 792*B*C*a^3
*b^11*c^4*d^8*f^2 - 776*B*C*a^8*b^6*c^9*d^3*f^2 + 688*B*C*a^4*b^10*c^7*d^5*
f^2 - 672*B*C*a^3*b^11*c^10*d^2*f^2 - 592*B*C*a^9*b^5*c^2*d^10*f^2 + 544*B*
C*a^7*b^7*c^10*d^2*f^2 - 492*B*C*a^5*b^9*c^2*d^10*f^2 + 480*B*C*a^10*b^4*c^
5*d^7*f^2 - 392*B*C*a^5*b^9*c^10*d^2*f^2 + 332*B*C*a^9*b^5*c^8*d^4*f^2 - 32
8*B*C*a^11*b^3*c^6*d^6*f^2 + 320*B*C*a^2*b^12*c^9*d^3*f^2 + 272*B*C*a^12*b^
2*c^3*d^9*f^2 - 248*B*C*a^4*b^10*c^5*d^7*f^2 - 248*B*C*a^3*b^11*c^2*d^10*f^
2 - 208*B*C*a^10*b^4*c^7*d^5*f^2 - 192*B*C*a^2*b^12*c^5*d^7*f^2 + 144*B*C*a
^7*b^7*c^2*d^10*f^2 - 96*B*C*a^4*b^10*c^3*d^9*f^2 + 88*B*C*a^12*b^2*c^5*d^7

*f^2 - 72*B*C*a^11*b^3*c^8*d^4*f^2 - 48*B*C*a^12*b^2*c^7*d^5*f^2 + 48*B*C*a
^10*b^4*c^9*d^3*f^2 - 48*B*C*a^2*b^12*c^7*d^5*f^2 - 48*B*C*a^2*b^12*c^3*d^9
*f^2 - 12*B*C*a^9*b^5*c^10*d^2*f^2 + 4*B*C*a^5*b^9*c^6*d^6*f^2 + 5824*A*C*a
^5*b^9*c^7*d^5*f^2 - 4378*A*C*a^6*b^8*c^8*d^4*f^2 + 4296*A*C*a^5*b^9*c^5*d^
7*f^2 - 3912*A*C*a^6*b^8*c^6*d^6*f^2 - 3672*A*C*a^9*b^5*c^5*d^7*f^2 + 3594*
A*C*a^8*b^6*c^4*d^8*f^2 + 3236*A*C*a^8*b^6*c^6*d^6*f^2 + 2816*A*C*a^5*b^9*c
^9*d^3*f^2 + 2624*A*C*a^5*b^9*c^3*d^9*f^2 + 2432*A*C*a^7*b^7*c^7*d^5*f^2 -
2366*A*C*a^4*b^10*c^8*d^4*f^2 + 2298*A*C*a^10*b^4*c^4*d^8*f^2 + 1872*A*C*a^
7*b^7*c^3*d^9*f^2 + 1848*A*C*a^10*b^4*c^6*d^6*f^2 - 1644*A*C*a^4*b^10*c^6*d
^6*f^2 - 1488*A*C*a^9*b^5*c^7*d^5*f^2 - 1408*A*C*a^9*b^5*c^3*d^9*f^2 - 1308
*A*C*a^6*b^8*c^4*d^8*f^2 + 1248*A*C*a^7*b^7*c^5*d^7*f^2 - 1012*A*C*a^6*b^8*
c^10*d^2*f^2 + 1008*A*C*a^3*b^11*c^7*d^5*f^2 + 992*A*C*a^3*b^11*c^5*d^7*f^2
 + 928*A*C*a^3*b^11*c^3*d^9*f^2 + 848*A*C*a^7*b^7*c^9*d^3*f^2 + 636*A*C*a^8
*b^6*c^2*d^10*f^2 - 628*A*C*a^4*b^10*c^10*d^2*f^2 - 600*A*C*a^6*b^8*c^2*d^1
0*f^2 - 576*A*C*a^11*b^3*c^5*d^7*f^2 + 572*A*C*a^10*b^4*c^2*d^10*f^2 + 464*
A*C*a^8*b^6*c^8*d^4*f^2 - 304*A*C*a^4*b^10*c^4*d^8*f^2 + 304*A*C*a^2*b^12*c
^6*d^6*f^2 + 296*A*C*a^2*b^12*c^4*d^8*f^2 + 260*A*C*a^10*b^4*c^8*d^4*f^2 -
232*A*C*a^12*b^2*c^2*d^10*f^2 - 232*A*C*a^9*b^5*c^9*d^3*f^2 + 228*A*C*a^2*b
^12*c^10*d^2*f^2 - 188*A*C*a^4*b^10*c^2*d^10*f^2 + 144*A*C*a^11*b^3*c^3*d^9
*f^2 + 116*A*C*a^12*b^2*c^6*d^6*f^2 - 112*A*C*a^11*b^3*c^7*d^5*f^2 + 112*A*
C*a^3*b^11*c^9*d^3*f^2 + 92*A*C*a^8*b^6*c^10*d^2*f^2 + 74*A*C*a^12*b^2*c^4*
d^8*f^2 + 62*A*C*a^2*b^12*c^8*d^4*f^2 + 40*A*C*a^2*b^12*c^2*d^10*f^2 - 7008
*A*B*a^7*b^7*c^6*d^6*f^2 - 4032*A*B*a^7*b^7*c^4*d^8*f^2 + 3952*A*B*a^8*b^6*
c^7*d^5*f^2 + 3648*A*B*a^8*b^6*c^5*d^7*f^2 - 3392*A*B*a^7*b^7*c^8*d^4*f^2 +
 3264*A*B*a^6*b^8*c^7*d^5*f^2 - 2992*A*B*a^4*b^10*c^5*d^7*f^2 - 2368*A*B*a^
4*b^10*c^7*d^5*f^2 - 2304*A*B*a^4*b^10*c^3*d^9*f^2 - 1968*A*B*a^9*b^5*c^6*d
^6*f^2 - 1872*A*B*a^4*b^10*c^9*d^3*f^2 - 1728*A*B*a^7*b^7*c^2*d^10*f^2 + 17
12*A*B*a^3*b^11*c^8*d^4*f^2 - 1536*A*B*a^10*b^4*c^3*d^9*f^2 + 1536*A*B*a^6*
b^8*c^5*d^7*f^2 - 1392*A*B*a^2*b^12*c^5*d^7*f^2 + 1328*A*B*a^3*b^11*c^6*d^6
*f^2 - 1104*A*B*a^2*b^12*c^3*d^9*f^2 - 1056*A*B*a^6*b^8*c^3*d^9*f^2 + 976*A
*B*a^6*b^8*c^9*d^3*f^2 + 960*A*B*a^11*b^3*c^4*d^8*f^2 + 936*A*B*a^5*b^9*c^8
*d^4*f^2 - 912*A*B*a^10*b^4*c^5*d^7*f^2 + 848*A*B*a^8*b^6*c^9*d^3*f^2 + 816
*A*B*a^3*b^11*c^4*d^8*f^2 - 816*A*B*a^2*b^12*c^7*d^5*f^2 + 768*A*B*a^3*b^11
*c^10*d^2*f^2 + 672*A*B*a^8*b^6*c^3*d^9*f^2 - 632*A*B*a^9*b^5*c^8*d^4*f^2 -
 608*A*B*a^9*b^5*c^2*d^10*f^2 - 552*A*B*a^9*b^5*c^4*d^8*f^2 - 544*A*B*a^7*b
^7*c^10*d^2*f^2 - 480*A*B*a^5*b^9*c^2*d^10*f^2 + 464*A*B*a^5*b^9*c^10*d^2*f
^2 - 464*A*B*a^2*b^12*c^9*d^3*f^2 + 432*A*B*a^11*b^3*c^2*d^10*f^2 - 368*A*B
*a^12*b^2*c^3*d^9*f^2 - 256*A*B*a^5*b^9*c^6*d^6*f^2 - 208*A*B*a^12*b^2*c^5*
d^7*f^2 + 176*A*B*a^5*b^9*c^4*d^8*f^2 + 112*A*B*a^11*b^3*c^6*d^6*f^2 + 112*
A*B*a^10*b^4*c^7*d^5*f^2 - 16*A*B*a^3*b^11*c^2*d^10*f^2 - 576*B*C*a^8*b^6*c
*d^11*f^2 + 400*B*C*a^4*b^10*c^11*d*f^2 - 288*B*C*a^6*b^8*c*d^11*f^2 - 176*
B*C*a^6*b^8*c^11*d*f^2 + 128*B*C*a^10*b^4*c*d^11*f^2 - 108*B*C*a*b^13*c^4*d
^8*f^2 - 104*B*C*a^4*b^10*c*d^11*f^2 - 92*B*C*a^13*b*c^4*d^8*f^2 - 60*B*C*a
*b^13*c^8*d^4*f^2 - 60*B*C*a*b^13*c^6*d^6*f^2 + 48*B*C*a^2*b^12*c^11*d*f^2
- 40*B*C*a*b^13*c^2*d^10*f^2 - 28*B*C*a^13*b*c^2*d^10*f^2 - 24*B*C*a^12*b^2

*c*d^11*f^2 + 20*B*C*a*b^13*c^10*d^2*f^2 - 16*B*C*a^2*b^12*c*d^11*f^2 + 12*
B*C*a^13*b*c^6*d^6*f^2 + 912*A*C*a^7*b^7*c*d^11*f^2 + 808*A*C*a^5*b^9*c*d^1
1*f^2 + 432*A*C*a^5*b^9*c^11*d*f^2 + 336*A*C*a^3*b^11*c*d^11*f^2 + 224*A*C*
a^11*b^3*c*d^11*f^2 - 112*A*C*a^3*b^11*c^11*d*f^2 + 112*A*C*a*b^13*c^3*d^9*
f^2 - 88*A*C*a*b^13*c^9*d^3*f^2 + 80*A*C*a^13*b*c^3*d^9*f^2 + 56*A*C*a*b^13
*c^5*d^7*f^2 + 48*A*C*a^9*b^5*c*d^11*f^2 - 40*A*C*a^13*b*c^5*d^7*f^2 - 16*A
*C*a^7*b^7*c^11*d*f^2 + 16*A*C*a*b^13*c^7*d^5*f^2 - 496*A*B*a^4*b^10*c*d^11
*f^2 - 400*A*B*a^4*b^10*c^11*d*f^2 + 288*A*B*a^8*b^6*c*d^11*f^2 - 288*A*B*a
^6*b^8*c*d^11*f^2 - 272*A*B*a^2*b^12*c*d^11*f^2 + 240*A*B*a*b^13*c^6*d^6*f^
2 - 224*A*B*a^10*b^4*c*d^11*f^2 + 192*A*B*a*b^13*c^8*d^4*f^2 + 192*A*B*a*b^
13*c^4*d^8*f^2 + 176*A*B*a^6*b^8*c^11*d*f^2 + 104*A*B*a^13*b*c^4*d^8*f^2 -
48*A*B*a^2*b^12*c^11*d*f^2 + 16*A*B*a^13*b*c^2*d^10*f^2 + 16*A*B*a*b^13*c^1
0*d^2*f^2 + 16*A*B*a*b^13*c^2*d^10*f^2 - 96*B*C*b^14*c^7*d^5*f^2 - 72*B*C*b
^14*c^5*d^7*f^2 - 24*B*C*b^14*c^9*d^3*f^2 - 16*B*C*b^14*c^3*d^9*f^2 + 116*A
*C*b^14*c^6*d^6*f^2 + 100*A*C*b^14*c^4*d^8*f^2 + 24*A*C*b^14*c^2*d^10*f^2 +
22*A*C*b^14*c^8*d^4*f^2 + 16*B*C*a^14*c^3*d^9*f^2 + 8*A*C*b^14*c^10*d^2*f^
2 - 192*A*B*b^14*c^5*d^7*f^2 - 176*A*B*b^14*c^3*d^9*f^2 - 112*B*C*a^11*b^3*
d^12*f^2 - 48*A*B*b^14*c^7*d^5*f^2 - 28*A*C*a^14*c^2*d^10*f^2 + 4*B*C*a^5*b
^9*d^12*f^2 + 2*A*C*a^14*c^4*d^8*f^2 + 150*A*C*a^10*b^4*d^12*f^2 - 80*B*C*a
^3*b^11*c^12*f^2 + 66*A*C*a^8*b^6*d^12*f^2 - 30*A*C*a^12*b^2*d^12*f^2 + 24*
B*C*a^5*b^9*c^12*f^2 - 16*A*B*a^14*c^3*d^9*f^2 - 12*A*C*a^4*b^10*d^12*f^2 -
576*A*B*a^7*b^7*d^12*f^2 - 432*A*B*a^9*b^5*d^12*f^2 - 400*A*B*a^5*b^9*d^12
*f^2 - 144*A*B*a^3*b^11*d^12*f^2 - 66*A*C*a^4*b^10*c^12*f^2 + 54*A*C*a^2*b^
12*c^12*f^2 - 32*A*B*a^11*b^3*d^12*f^2 + 2*A*C*a^6*b^8*c^12*f^2 + 80*A*B*a^
3*b^11*c^12*f^2 - 24*A*B*a^5*b^9*c^12*f^2 + 2508*C^2*a^6*b^8*c^6*d^6*f^2 +
2376*C^2*a^9*b^5*c^5*d^7*f^2 + 2357*C^2*a^6*b^8*c^8*d^4*f^2 - 2048*C^2*a^5*
b^9*c^7*d^5*f^2 + 1304*C^2*a^9*b^5*c^3*d^9*f^2 + 1303*C^2*a^4*b^10*c^8*d^4*
f^2 + 1212*C^2*a^4*b^10*c^6*d^6*f^2 - 1203*C^2*a^8*b^6*c^4*d^8*f^2 - 1192*C
^2*a^5*b^9*c^9*d^3*f^2 + 1062*C^2*a^6*b^8*c^4*d^8*f^2 + 984*C^2*a^9*b^5*c^7
*d^5*f^2 - 952*C^2*a^8*b^6*c^6*d^6*f^2 + 768*C^2*a^7*b^7*c^5*d^7*f^2 - 681*
C^2*a^10*b^4*c^4*d^8*f^2 - 672*C^2*a^5*b^9*c^5*d^7*f^2 - 480*C^2*a^10*b^4*c
^6*d^6*f^2 + 458*C^2*a^6*b^8*c^10*d^2*f^2 - 448*C^2*a^7*b^7*c^7*d^5*f^2 + 4
22*C^2*a^4*b^10*c^4*d^8*f^2 + 372*C^2*a^6*b^8*c^2*d^10*f^2 + 360*C^2*a^11*b
^3*c^5*d^7*f^2 + 312*C^2*a^7*b^7*c^3*d^9*f^2 + 278*C^2*a^4*b^10*c^10*d^2*f^
2 - 232*C^2*a^7*b^7*c^9*d^3*f^2 + 194*C^2*a^12*b^2*c^2*d^10*f^2 + 176*C^2*a
^9*b^5*c^9*d^3*f^2 + 152*C^2*a^3*b^11*c^5*d^7*f^2 + 124*C^2*a^4*b^10*c^2*d^
10*f^2 - 120*C^2*a^3*b^11*c^7*d^5*f^2 - 114*C^2*a^2*b^12*c^10*d^2*f^2 - 102
*C^2*a^8*b^6*c^2*d^10*f^2 + 101*C^2*a^12*b^2*c^4*d^8*f^2 + 100*C^2*a^2*b^12
*c^6*d^6*f^2 - 88*C^2*a^5*b^9*c^3*d^9*f^2 + 77*C^2*a^2*b^12*c^8*d^4*f^2 + 7
2*C^2*a^11*b^3*c^3*d^9*f^2 - 64*C^2*a^8*b^6*c^10*d^2*f^2 + 64*C^2*a^3*b^11*
c^3*d^9*f^2 - 58*C^2*a^10*b^4*c^2*d^10*f^2 + 56*C^2*a^12*b^2*c^6*d^6*f^2 +
56*C^2*a^11*b^3*c^7*d^5*f^2 + 40*C^2*a^3*b^11*c^9*d^3*f^2 + 36*C^2*a^12*b^2
*c^8*d^4*f^2 + 32*C^2*a^2*b^12*c^4*d^8*f^2 + 26*C^2*a^10*b^4*c^8*d^4*f^2 +
16*C^2*a^2*b^12*c^2*d^10*f^2 + 2*C^2*a^8*b^6*c^8*d^4*f^2 + 2277*B^2*a^8*b^6
*c^4*d^8*f^2 + 2144*B^2*a^5*b^9*c^7*d^5*f^2 - 2112*B^2*a^9*b^5*c^5*d^7*f^2

+ 2028*B^2*a^8*b^6*c^6*d^6*f^2 - 1671*B^2*a^6*b^8*c^8*d^4*f^2 + 1275*B^2*a^10*b^4*c^4*d^8*f^2 + 1176*B^2*a^5*b^9*c^5*d^7*f^2 + 1096*B^2*a^5*b^9*c^9*d^3*f^2 - 1044*B^2*a^6*b^8*c^6*d^6*f^2 + 984*B^2*a^10*b^4*c^6*d^6*f^2 - 968*B^2*a^9*b^5*c^3*d^9*f^2 - 888*B^2*a^9*b^5*c^7*d^5*f^2 + 672*B^2*a^7*b^7*c^7*d^5*f^2 + 664*B^2*a^5*b^9*c^3*d^9*f^2 - 649*B^2*a^4*b^10*c^8*d^4*f^2 + 618*B^2*a^8*b^6*c^2*d^10*f^2 + 514*B^2*a^4*b^10*c^4*d^8*f^2 + 460*B^2*a^2*b^12*c^6*d^6*f^2 + 422*B^2*a^8*b^6*c^8*d^4*f^2 + 406*B^2*a^10*b^4*c^2*d^10*f^2 - 382*B^2*a^6*b^8*c^10*d^2*f^2 + 368*B^2*a^2*b^12*c^4*d^8*f^2 - 312*B^2*a^11*b^3*c^5*d^7*f^2 + 312*B^2*a^7*b^7*c^3*d^9*f^2 + 248*B^2*a^7*b^7*c^9*d^3*f^2 + 245*B^2*a^2*b^12*c^8*d^4*f^2 - 192*B^2*a^7*b^7*c^5*d^7*f^2 - 184*B^2*a^3*b^11*c^9*d^3*f^2 + 182*B^2*a^2*b^12*c^10*d^2*f^2 + 176*B^2*a^3*b^11*c^3*d^9*f^2 + 174*B^2*a^6*b^8*c^4*d^8*f^2 - 170*B^2*a^4*b^10*c^10*d^2*f^2 - 152*B^2*a^9*b^5*c^9*d^3*f^2 + 152*B^2*a^4*b^10*c^2*d^10*f^2 + 142*B^2*a^10*b^4*c^8*d^4*f^2 - 90*B^2*a^12*b^2*c^2*d^10*f^2 + 88*B^2*a^2*b^12*c^2*d^10*f^2 + 84*B^2*a^8*b^6*c^10*d^2*f^2 + 84*B^2*a^6*b^8*c^2*d^10*f^2 + 60*B^2*a^12*b^2*c^6*d^6*f^2 - 56*B^2*a^11*b^3*c^7*d^5*f^2 + 53*B^2*a^12*b^2*c^4*d^8*f^2 + 24*B^2*a^11*b^3*c^3*d^9*f^2 + 24*B^2*a^4*b^10*c^6*d^6*f^2 + 24*B^2*a^3*b^11*c^7*d^5*f^2 - 8*B^2*a^3*b^11*c^5*d^7*f^2 + 4566*A^2*a^6*b^8*c^4*d^8*f^2 + 4284*A^2*a^6*b^8*c^6*d^6*f^2 - 3776*A^2*a^5*b^9*c^7*d^5*f^2 - 3624*A^2*a^5*b^9*c^5*d^7*f^2 + 3122*A^2*a^4*b^10*c^4*d^8*f^2 + 3108*A^2*a^6*b^8*c^2*d^10*f^2 + 2741*A^2*a^6*b^8*c^8*d^4*f^2 + 2592*A^2*a^4*b^10*c^6*d^6*f^2 - 2536*A^2*a^5*b^9*c^3*d^9*f^2 + 2224*A^2*a^4*b^10*c^2*d^10*f^2 - 2184*A^2*a^7*b^7*c^3*d^9*f^2 - 2016*A^2*a^7*b^7*c^5*d^7*f^2 - 1984*A^2*a^7*b^7*c^7*d^5*f^2 + 1626*A^2*a^8*b^6*c^2*d^10*f^2 - 1624*A^2*a^5*b^9*c^9*d^3*f^2 + 1603*A^2*a^4*b^10*c^8*d^4*f^2 + 1296*A^2*a^9*b^5*c^5*d^7*f^2 - 1144*A^2*a^3*b^11*c^5*d^7*f^2 - 992*A^2*a^3*b^11*c^3*d^9*f^2 + 968*A^2*a^2*b^12*c^4*d^8*f^2 - 888*A^2*a^3*b^11*c^7*d^5*f^2 + 849*A^2*a^8*b^6*c^4*d^8*f^2 + 808*A^2*a^2*b^12*c^2*d^10*f^2 - 616*A^2*a^7*b^7*c^9*d^3*f^2 + 554*A^2*a^6*b^8*c^10*d^2*f^2 - 504*A^2*a^10*b^4*c^6*d^6*f^2 + 504*A^2*a^9*b^5*c^7*d^5*f^2 + 460*A^2*a^2*b^12*c^6*d^6*f^2 + 350*A^2*a^10*b^4*c^2*d^10*f^2 + 350*A^2*a^4*b^10*c^10*d^2*f^2 - 321*A^2*a^10*b^4*c^4*d^8*f^2 + 216*A^2*a^11*b^3*c^5*d^7*f^2 - 216*A^2*a^11*b^3*c^3*d^9*f^2 + 182*A^2*a^12*b^2*c^2*d^10*f^2 - 152*A^2*a^3*b^11*c^9*d^3*f^2 - 124*A^2*a^8*b^6*c^6*d^6*f^2 - 114*A^2*a^2*b^12*c^10*d^2*f^2 + 104*A^2*a^9*b^5*c^3*d^9*f^2 + 77*A^2*a^2*b^12*c^8*d^4*f^2 + 74*A^2*a^8*b^6*c^8*d^4*f^2 - 70*A^2*a^10*b^4*c^8*d^4*f^2 + 56*A^2*a^11*b^3*c^7*d^5*f^2 + 56*A^2*a^9*b^5*c^9*d^3*f^2 + 41*A^2*a^12*b^2*c^4*d^8*f^2 - 28*A^2*a^12*b^2*c^6*d^6*f^2 - 28*A^2*a^8*b^6*c^10*d^2*f^2 - 16*B*C*b^14*c^11*d*f^2 - 16*B*C*a^14*c*d^11*f^2 - 48*A*B*b^14*c*d^11*f^2 + 16*A*B*b^14*c^11*d*f^2 + 12*B*C*a^13*b*d^12*f^2 + 24*B*C*a*b^13*c^12*f^2 + 16*A*B*a^14*c*d^11*f^2 - 24*A*B*a^13*b*d^12*f^2 - 24*A*B*a*b^13*d^12*f^2 - 24*A*B*a*b^13*c^12*f^2 + 216*C^2*a^9*b^5*c*d^11*f^2 - 216*C^2*a^5*b^9*c^11*d*f^2 + 56*C^2*a^3*b^11*c^11*d*f^2 + 56*C^2*a*b^13*c^9*d^3*f^2 + 56*C^2*a*b^13*c^5*d^7*f^2 - 40*C^2*a^11*b^3*c*d^11*f^2 + 40*C^2*a*b^13*c^7*d^5*f^2 + 32*C^2*a^13*b*c^5*d^7*f^2 - 24*C^2*a^7*b^7*c*d^11*f^2 - 16*C^2*a^13*b*c^3*d^9*f^2 + 16*C^2*a*b^13*c^3*d^9*f^2 + 8*C^2*a^7*b^7*c^11*d*f^2 - 8*C^2*a^5*b^9*c*d^11*f^2 + 264*B^2*a^7*b^7

*c*d^11*f^2 + 224*B^2*a^5*b^9*c*d^11*f^2 + 168*B^2*a^5*b^9*c^11*d*f^2 - 112
*B^2*a*b^13*c^9*d^3*f^2 - 104*B^2*a^3*b^11*c^11*d*f^2 - 104*B^2*a*b^13*c^7*
d^5*f^2 + 96*B^2*a^3*b^11*c*d^11*f^2 + 88*B^2*a^11*b^3*c*d^11*f^2 - 72*B^2*
a^9*b^5*c*d^11*f^2 - 64*B^2*a*b^13*c^5*d^7*f^2 + 32*B^2*a^13*b*c^3*d^9*f^2
- 24*B^2*a^13*b*c^5*d^7*f^2 - 24*B^2*a^7*b^7*c^11*d*f^2 + 16*B^2*a*b^13*c^3
*d^9*f^2 - 888*A^2*a^7*b^7*c*d^11*f^2 - 800*A^2*a^5*b^9*c*d^11*f^2 - 336*A^
2*a^3*b^11*c*d^11*f^2 - 264*A^2*a^9*b^5*c*d^11*f^2 - 216*A^2*a^5*b^9*c^11*d
*f^2 - 184*A^2*a^11*b^3*c*d^11*f^2 - 128*A^2*a*b^13*c^3*d^9*f^2 - 112*A^2*a
*b^13*c^5*d^7*f^2 - 64*A^2*a^13*b*c^3*d^9*f^2 + 56*A^2*a^3*b^11*c^11*d*f^2
- 56*A^2*a*b^13*c^7*d^5*f^2 + 32*A^2*a*b^13*c^9*d^3*f^2 + 8*A^2*a^13*b*c^5*
d^7*f^2 + 8*A^2*a^7*b^7*c^11*d*f^2 + 24*C^2*a*b^13*c^11*d*f^2 - 16*C^2*a^13
*b*c*d^11*f^2 - 40*B^2*a*b^13*c^11*d*f^2 + 24*B^2*a^13*b*c*d^11*f^2 + 16*B^
2*a*b^13*c*d^11*f^2 - 48*A^2*a*b^13*c*d^11*f^2 - 40*A^2*a^13*b*c*d^11*f^2 +
24*A^2*a*b^13*c^11*d*f^2 - 6*A*C*b^14*c^12*f^2 + 2*A*C*a^14*d^12*f^2 + 31*
C^2*b^14*c^8*d^4*f^2 + 20*C^2*b^14*c^6*d^6*f^2 + 4*C^2*b^14*c^4*d^8*f^2 + 2
*C^2*b^14*c^10*d^2*f^2 + 80*B^2*b^14*c^6*d^6*f^2 + 64*B^2*b^14*c^4*d^8*f^2
+ 31*B^2*b^14*c^8*d^4*f^2 + 16*B^2*b^14*c^2*d^10*f^2 + 14*C^2*a^14*c^2*d^10
*f^2 + 14*B^2*b^14*c^10*d^2*f^2 - C^2*a^14*c^4*d^8*f^2 + 120*A^2*b^14*c^2*d
^10*f^2 + 112*A^2*b^14*c^4*d^8*f^2 + 33*C^2*a^12*b^2*d^12*f^2 - 27*C^2*a^10
*b^4*d^12*f^2 - 17*A^2*b^14*c^8*d^4*f^2 - 10*B^2*a^14*c^2*d^10*f^2 - 10*A^2
*b^14*c^10*d^2*f^2 + 8*A^2*b^14*c^6*d^6*f^2 + 3*C^2*a^8*b^6*d^12*f^2 + 3*B^
2*a^14*c^4*d^8*f^2 + 117*B^2*a^10*b^4*d^12*f^2 + 111*B^2*a^8*b^6*d^12*f^2 +
72*B^2*a^6*b^8*d^12*f^2 + 33*C^2*a^4*b^10*c^12*f^2 - 27*C^2*a^2*b^12*c^12*
f^2 + 24*B^2*a^4*b^10*d^12*f^2 + 14*A^2*a^14*c^2*d^10*f^2 + 4*B^2*a^2*b^12*
d^12*f^2 - 3*B^2*a^12*b^2*d^12*f^2 - C^2*a^6*b^8*c^12*f^2 - A^2*a^14*c^4*d^
8*f^2 + 720*A^2*a^6*b^8*d^12*f^2 + 552*A^2*a^4*b^10*d^12*f^2 + 471*A^2*a^8*
b^6*d^12*f^2 + 216*A^2*a^2*b^12*d^12*f^2 + 93*A^2*a^10*b^4*d^12*f^2 + 33*B^
2*a^2*b^12*c^12*f^2 + 33*A^2*a^12*b^2*d^12*f^2 - 27*B^2*a^4*b^10*c^12*f^2 +
3*B^2*a^6*b^8*c^12*f^2 + 33*A^2*a^4*b^10*c^12*f^2 - 27*A^2*a^2*b^12*c^12*f
^2 - A^2*a^6*b^8*c^12*f^2 + 3*C^2*b^14*c^12*f^2 - C^2*a^14*d^12*f^2 + 36*A^
2*b^14*d^12*f^2 + 3*B^2*a^14*d^12*f^2 - B^2*b^14*c^12*f^2 + 3*A^2*b^14*c^12
*f^2 - A^2*a^14*d^12*f^2 - 44*A*B*C*a^10*b*c*d^9*f + 3816*A*B*C*a^4*b^7*c^5
*d^5*f + 2920*A*B*C*a^5*b^6*c^2*d^8*f - 2736*A*B*C*a^6*b^5*c^3*d^7*f - 2672
*A*B*C*a^3*b^8*c^4*d^6*f + 1996*A*B*C*a^7*b^4*c^4*d^6*f - 1412*A*B*C*a^5*b^
6*c^6*d^4*f + 1120*A*B*C*a^2*b^9*c^3*d^7*f + 1080*A*B*C*a^7*b^4*c^2*d^8*f +
1040*A*B*C*a^2*b^9*c^5*d^5*f + 684*A*B*C*a^5*b^6*c^4*d^6*f + 592*A*B*C*a^4
*b^7*c^3*d^7*f - 560*A*B*C*a^2*b^9*c^7*d^3*f - 448*A*B*C*a^3*b^8*c^2*d^8*f
- 400*A*B*C*a^8*b^3*c^5*d^5*f - 398*A*B*C*a^9*b^2*c^2*d^8*f - 312*A*B*C*a^3
*b^8*c^6*d^4*f + 166*A*B*C*a^3*b^8*c^8*d^2*f + 136*A*B*C*a^6*b^5*c^5*d^5*f
+ 128*A*B*C*a^6*b^5*c^7*d^3*f - 100*A*B*C*a^7*b^4*c^6*d^4*f - 64*A*B*C*a^9*
b^2*c^4*d^6*f + 64*A*B*C*a^4*b^7*c^7*d^3*f - 32*A*B*C*a^8*b^3*c^3*d^7*f - 1
6*A*B*C*a^5*b^6*c^8*d^2*f - 1312*A*B*C*a^4*b^7*c*d^9*f + 996*A*B*C*a^8*b^3*
c*d^9*f + 728*A*B*C*a*b^10*c^6*d^4*f - 624*A*B*C*a^6*b^5*c*d^9*f - 584*A*B*
C*a*b^10*c^2*d^8*f - 512*A*B*C*a*b^10*c^4*d^6*f - 320*A*B*C*a^2*b^9*c*d^9*f
- 98*A*B*C*a*b^10*c^8*d^2*f + 36*A*B*C*a^2*b^9*c^9*d*f + 32*A*B*C*a^10*b*c

^3*d^7*f - 16*A*B*C*a^4*b^7*c^9*d*f + 46*B*C^2*a^10*b*c*d^9*f - 16*B^2*C*a*
b^10*c*d^9*f - 2*B^2*C*a*b^10*c^9*d*f + 312*A^2*C*a*b^10*c*d^9*f - 48*A*C^2
*a*b^10*c*d^9*f - 6*A^2*C*a*b^10*c^9*d*f + 6*A*C^2*a*b^10*c^9*d*f + 208*A*B
^2*a*b^10*c*d^9*f - 2*A^2*B*a^10*b*c*d^9*f + 2*A*B^2*a*b^10*c^9*d*f - 224*A
*B*C*b^11*c^5*d^5*f + 80*A*B*C*b^11*c^7*d^3*f - 32*A*B*C*b^11*c^3*d^7*f + 2
*A*B*C*a^11*c^2*d^8*f - 480*A*B*C*a^7*b^4*d^10*f + 78*A*B*C*a^9*b^2*d^10*f
- 64*A*B*C*a^5*b^6*d^10*f + 2*A*B*C*a^3*b^8*c^10*f - 1692*B*C^2*a^4*b^7*c^5
*d^5*f - 1500*B^2*C*a^5*b^6*c^5*d^5*f - 1464*B^2*C*a^5*b^6*c^3*d^7*f + 1426
*B*C^2*a^5*b^6*c^6*d^4*f - 1158*B^2*C*a^4*b^7*c^6*d^4*f + 1152*B*C^2*a^6*b^
5*c^3*d^7*f + 1026*B^2*C*a^6*b^5*c^4*d^6*f - 974*B*C^2*a^7*b^4*c^4*d^6*f +
960*B^2*C*a^3*b^8*c^5*d^5*f - 884*B*C^2*a^5*b^6*c^2*d^8*f - 764*B^2*C*a^7*b
^4*c^5*d^5*f + 752*B^2*C*a^4*b^7*c^2*d^8*f - 752*B*C^2*a^4*b^7*c^3*d^7*f +
738*B^2*C*a^4*b^7*c^4*d^6*f - 688*B^2*C*a^2*b^9*c^6*d^4*f - 675*B^2*C*a^8*b
^3*c^2*d^8*f + 560*B*C^2*a^8*b^3*c^5*d^5*f + 496*B*C^2*a^3*b^8*c^4*d^6*f +
496*B*C^2*a^2*b^9*c^7*d^3*f - 468*B*C^2*a^7*b^4*c^2*d^8*f + 456*B^2*C*a^3*b
^8*c^7*d^3*f - 452*B^2*C*a^8*b^3*c^4*d^6*f - 416*B*C^2*a^2*b^9*c^3*d^7*f +
378*B*C^2*a^5*b^6*c^4*d^6*f + 376*B*C^2*a^8*b^3*c^3*d^7*f - 360*B^2*C*a^6*b
^5*c^2*d^8*f + 355*B*C^2*a^9*b^2*c^2*d^8*f + 346*B^2*C*a^6*b^5*c^6*d^4*f -
320*B^2*C*a^2*b^9*c^4*d^6*f + 268*B^2*C*a^2*b^9*c^2*d^8*f + 216*B^2*C*a^7*b
^4*c^3*d^7*f - 203*B*C^2*a^3*b^8*c^8*d^2*f - 184*B*C^2*a^6*b^5*c^7*d^3*f +
170*B*C^2*a^7*b^4*c^6*d^4*f + 160*B^2*C*a^5*b^6*c^7*d^3*f - 160*B*C^2*a^2*b
^9*c^5*d^5*f - 140*B^2*C*a^4*b^7*c^8*d^2*f - 136*B*C^2*a^3*b^8*c^2*d^8*f +
112*B^2*C*a^9*b^2*c^3*d^7*f + 91*B^2*C*a^2*b^9*c^8*d^2*f + 88*B*C^2*a^4*b^7
*c^7*d^3*f + 72*B^2*C*a^8*b^3*c^6*d^4*f - 64*B^2*C*a^3*b^8*c^3*d^7*f - 60*B
*C^2*a^3*b^8*c^6*d^4*f + 56*B*C^2*a^9*b^2*c^4*d^6*f + 52*B*C^2*a^6*b^5*c^5*
d^5*f + 48*B^2*C*a^9*b^2*c^5*d^5*f - 48*B^2*C*a^7*b^4*c^7*d^3*f + 44*B*C^2*
a^5*b^6*c^8*d^2*f - 36*B*C^2*a^9*b^2*c^6*d^4*f + 12*B^2*C*a^6*b^5*c^8*d^2*f
- 2958*A^2*C*a^4*b^7*c^4*d^6*f - 1932*A^2*C*a^4*b^7*c^2*d^8*f + 1848*A^2*C
*a^5*b^6*c^3*d^7*f + 1728*A^2*C*a^3*b^8*c^3*d^7*f + 1524*A^2*C*a^5*b^6*c^5*
d^5*f + 1374*A*C^2*a^4*b^7*c^4*d^6*f - 1272*A*C^2*a^5*b^6*c^3*d^7*f - 1236*
A*C^2*a^5*b^6*c^5*d^5*f + 1116*A*C^2*a^4*b^7*c^2*d^8*f - 1110*A^2*C*a^6*b^5
*c^4*d^6*f + 1038*A*C^2*a^6*b^5*c^4*d^6*f - 768*A^2*C*a^2*b^9*c^2*d^8*f - 6
96*A^2*C*a^7*b^4*c^3*d^7*f - 666*A*C^2*a^4*b^7*c^6*d^4*f + 564*A^2*C*a^6*b^
5*c^2*d^8*f - 564*A*C^2*a^7*b^4*c^5*d^5*f - 555*A*C^2*a^8*b^3*c^2*d^8*f + 5
19*A^2*C*a^8*b^3*c^2*d^8*f - 480*A*C^2*a^3*b^8*c^3*d^7*f + 456*A*C^2*a^3*b^
8*c^5*d^5*f - 420*A*C^2*a^2*b^9*c^6*d^4*f + 408*A*C^2*a^7*b^4*c^3*d^7*f + 4
08*A*C^2*a^2*b^9*c^2*d^8*f + 348*A^2*C*a^2*b^9*c^6*d^4*f - 348*A*C^2*a^6*b^
5*c^2*d^8*f + 342*A*C^2*a^6*b^5*c^6*d^4*f - 336*A*C^2*a^8*b^3*c^4*d^6*f + 3
24*A^2*C*a^7*b^4*c^5*d^5*f - 312*A^2*C*a^2*b^9*c^4*d^6*f + 264*A^2*C*a^8*b^
3*c^4*d^6*f + 240*A*C^2*a^5*b^6*c^7*d^3*f + 195*A*C^2*a^2*b^9*c^8*d^2*f - 1
74*A^2*C*a^6*b^5*c^6*d^4*f + 144*A*C^2*a^9*b^2*c^3*d^7*f - 123*A^2*C*a^2*b^
9*c^8*d^2*f + 120*A*C^2*a^3*b^8*c^7*d^3*f + 108*A*C^2*a^8*b^3*c^6*d^4*f - 1
02*A^2*C*a^4*b^7*c^6*d^4*f - 96*A^2*C*a^4*b^7*c^8*d^2*f + 72*A^2*C*a^3*b^8*
c^7*d^3*f + 72*A*C^2*a^9*b^2*c^5*d^5*f - 48*A^2*C*a^9*b^2*c^3*d^7*f + 48*A^
2*C*a^5*b^6*c^7*d^3*f - 48*A*C^2*a^2*b^9*c^4*d^6*f - 24*A^2*C*a^3*b^8*c^5*d

^5*f - 12*A*C^2*a^4*b^7*c^8*d^2*f + 2736*A^2*B*a^6*b^5*c^3*d^7*f + 2464*A^2
*B*a^3*b^8*c^4*d^6*f - 2298*A*B^2*a^4*b^7*c^4*d^6*f - 2252*A^2*B*a^5*b^6*c^
2*d^8*f - 1692*A^2*B*a^4*b^7*c^5*d^5*f - 1592*A*B^2*a^4*b^7*c^2*d^8*f - 133
8*A*B^2*a^6*b^5*c^4*d^6*f + 1320*A*B^2*a^5*b^6*c^3*d^7*f + 1212*A*B^2*a^5*b
^6*c^5*d^5*f - 1056*A*B^2*a^3*b^8*c^5*d^5*f + 1024*A^2*B*a^4*b^7*c^3*d^7*f
- 1022*A^2*B*a^7*b^4*c^4*d^6*f - 880*A^2*B*a^2*b^9*c^5*d^5*f - 846*A^2*B*a^
5*b^6*c^4*d^6*f - 840*A*B^2*a^7*b^4*c^3*d^7*f + 760*A*B^2*a^2*b^9*c^6*d^4*f
- 704*A^2*B*a^2*b^9*c^3*d^7*f + 688*A*B^2*a^3*b^8*c^3*d^7*f + 660*A^2*B*a^
3*b^8*c^6*d^4*f - 612*A^2*B*a^7*b^4*c^2*d^8*f + 462*A*B^2*a^4*b^7*c^6*d^4*f
+ 459*A*B^2*a^8*b^3*c^2*d^8*f - 412*A*B^2*a^2*b^9*c^2*d^8*f - 408*A*B^2*a^
3*b^8*c^7*d^3*f + 388*A^2*B*a^6*b^5*c^5*d^5*f + 296*A^2*B*a^3*b^8*c^2*d^8*f
+ 288*A*B^2*a^6*b^5*c^2*d^8*f + 284*A*B^2*a^7*b^4*c^5*d^5*f + 236*A*B^2*a^
8*b^3*c^4*d^6*f - 226*A*B^2*a^6*b^5*c^6*d^4*f + 212*A*B^2*a^2*b^9*c^4*d^6*f
+ 202*A^2*B*a^5*b^6*c^6*d^4*f - 152*A^2*B*a^4*b^7*c^7*d^3*f + 88*A^2*B*a^8
*b^3*c^3*d^7*f + 79*A^2*B*a^9*b^2*c^2*d^8*f - 70*A^2*B*a^7*b^4*c^6*d^4*f +
68*A*B^2*a^4*b^7*c^8*d^2*f + 64*A^2*B*a^2*b^9*c^7*d^3*f - 64*A*B^2*a^9*b^2*
c^3*d^7*f + 56*A^2*B*a^8*b^3*c^5*d^5*f + 56*A^2*B*a^6*b^5*c^7*d^3*f + 37*A^
2*B*a^3*b^8*c^8*d^2*f - 28*A^2*B*a^9*b^2*c^4*d^6*f - 28*A^2*B*a^5*b^6*c^8*d
^2*f + 17*A*B^2*a^2*b^9*c^8*d^2*f - 16*A*B^2*a^5*b^6*c^7*d^3*f + 48*A*B*C*b
^11*c*d^9*f + 4*A*B*C*b^11*c^9*d*f + 24*A*B*C*a*b^10*d^10*f - 6*A*B*C*a*b^1
0*c^10*f + 432*B^2*C*a^7*b^4*c*d^9*f - 376*B*C^2*a*b^10*c^6*d^4*f - 354*B*C
^2*a^8*b^3*c*d^9*f + 352*B^2*C*a*b^10*c^5*d^5*f + 320*B^2*C*a^5*b^6*c*d^9*f
+ 256*B^2*C*a*b^10*c^3*d^7*f - 232*B^2*C*a*b^10*c^7*d^3*f - 210*B^2*C*a^9*
b^2*c*d^9*f - 152*B*C^2*a*b^10*c^4*d^6*f + 85*B*C^2*a*b^10*c^8*d^2*f + 72*B
^2*C*a^3*b^8*c*d^9*f - 48*B*C^2*a^6*b^5*c*d^9*f - 40*B*C^2*a^10*b*c^3*d^7*f
+ 40*B*C^2*a*b^10*c^2*d^8*f + 37*B^2*C*a^10*b*c^2*d^8*f + 22*B^2*C*a^3*b^8
*c^9*d*f - 18*B*C^2*a^2*b^9*c^9*d*f + 16*B*C^2*a^2*b^9*c*d^9*f - 12*B^2*C*a
^10*b*c^4*d^6*f + 8*B*C^2*a^4*b^7*c^9*d*f + 8*B*C^2*a^4*b^7*c*d^9*f - 984*A
^2*C*a^7*b^4*c*d^9*f + 672*A^2*C*a^3*b^8*c*d^9*f + 552*A*C^2*a^7*b^4*c*d^9*
f - 504*A^2*C*a*b^10*c^5*d^5*f - 408*A^2*C*a^5*b^6*c*d^9*f + 408*A*C^2*a^5*
b^6*c*d^9*f + 336*A*C^2*a*b^10*c^5*d^5*f - 216*A*C^2*a*b^10*c^7*d^3*f + 192
*A*C^2*a*b^10*c^3*d^7*f - 162*A*C^2*a^9*b^2*c*d^9*f + 120*A^2*C*a*b^10*c^7*
d^3*f + 96*A^2*C*a*b^10*c^3*d^7*f + 90*A^2*C*a^9*b^2*c*d^9*f + 66*A^2*C*a^3
*b^8*c^9*d*f - 66*A*C^2*a^3*b^8*c^9*d*f + 57*A*C^2*a^10*b*c^2*d^8*f - 48*A*
C^2*a^3*b^8*c*d^9*f - 9*A^2*C*a^10*b*c^2*d^8*f + 1736*A^2*B*a^4*b^7*c*d^9*f
+ 1248*A^2*B*a^6*b^5*c*d^9*f - 1008*A*B^2*a^7*b^4*c*d^9*f + 772*A^2*B*a*b^
10*c^4*d^6*f - 688*A*B^2*a*b^10*c^5*d^5*f - 608*A*B^2*a^5*b^6*c*d^9*f + 436
*A^2*B*a*b^10*c^2*d^8*f - 426*A^2*B*a^8*b^3*c*d^9*f + 312*A*B^2*a^3*b^8*c*d
^9*f + 304*A^2*B*a^2*b^9*c*d^9*f - 244*A^2*B*a*b^10*c^6*d^4*f - 160*A*B^2*a
*b^10*c^3*d^7*f + 114*A*B^2*a^9*b^2*c*d^9*f + 88*A*B^2*a*b^10*c^7*d^3*f - 2
2*A*B^2*a^3*b^8*c^9*d*f - 18*A^2*B*a^2*b^9*c^9*d*f + 13*A^2*B*a*b^10*c^8*d^
2*f - 13*A*B^2*a^10*b*c^2*d^8*f + 8*A^2*B*a^10*b*c^3*d^7*f + 8*A^2*B*a^4*b^
7*c^9*d*f + 112*B^2*C*b^11*c^6*d^4*f - 64*B*C^2*b^11*c^7*d^3*f + 16*B^2*C*b
^11*c^4*d^6*f - 16*B^2*C*b^11*c^2*d^8*f + 16*B*C^2*b^11*c^5*d^5*f + 16*B*C^
2*b^11*c^3*d^7*f - B^2*C*b^11*c^8*d^2*f + 96*A^2*C*b^11*c^4*d^6*f - 84*A^2*

C*b^11*c^6*d^4*f + 72*A*C^2*b^11*c^6*d^4*f - 24*A*C^2*b^11*c^4*d^6*f - 24*A
*C^2*b^11*c^2*d^8*f - 21*A*C^2*b^11*c^8*d^2*f + 12*A^2*C*b^11*c^2*d^8*f + 9
*A^2*C*b^11*c^8*d^2*f - B*C^2*a^11*c^2*d^8*f + 176*A*B^2*b^11*c^4*d^6*f + 1
36*A^2*B*b^11*c^5*d^5*f - 128*A^2*B*b^11*c^3*d^7*f + 112*A*B^2*b^11*c^2*d^8
*f + 111*B^2*C*a^8*b^3*d^10*f - 64*A*B^2*b^11*c^6*d^4*f - 39*B*C^2*a^9*b^2*
d^10*f + 24*B*C^2*a^7*b^4*d^10*f - 16*A^2*B*b^11*c^7*d^3*f - 4*B^2*C*a^2*b^
9*d^10*f - 4*B*C^2*a^5*b^6*d^10*f + 432*A^2*C*a^6*b^5*d^10*f + 192*A^2*C*a^
4*b^7*d^10*f - 111*A^2*C*a^8*b^3*d^10*f + 111*A*C^2*a^8*b^3*d^10*f - 72*A*C
^2*a^6*b^5*d^10*f + 12*A*C^2*a^4*b^7*d^10*f - 3*B^2*C*a^2*b^9*c^10*f - A^2*
B*a^11*c^2*d^8*f - B*C^2*a^3*b^8*c^10*f + 456*A^2*B*a^7*b^4*d^10*f - 288*A^
2*B*a^3*b^8*d^10*f + 252*A*B^2*a^6*b^5*d^10*f + 192*A*B^2*a^4*b^7*d^10*f -
183*A*B^2*a^8*b^3*d^10*f - 148*A^2*B*a^5*b^6*d^10*f + 76*A*B^2*a^2*b^9*d^10
*f - 9*A^2*C*a^2*b^9*c^10*f + 9*A*C^2*a^2*b^9*c^10*f - 3*A^2*B*a^9*b^2*d^10
*f + 3*A*B^2*a^2*b^9*c^10*f - A^2*B*a^3*b^8*c^10*f - 2*C^3*a*b^10*c^9*d*f -
 2*B^3*a^10*b*c*d^9*f - 264*A^3*a*b^10*c*d^9*f + 2*A^3*a*b^10*c^9*d*f - 2*B
*C^2*b^11*c^9*d*f - 2*B^2*C*a^11*c*d^9*f - 120*A^2*B*b^11*c*d^9*f - 9*B^2*C
*a^10*b*d^10*f - 6*A^2*C*a^11*c*d^9*f + 6*A*C^2*a^11*c*d^9*f - 2*A^2*B*b^11
*c^9*d*f + 9*A^2*C*a^10*b*d^10*f - 9*A*C^2*a^10*b*d^10*f + 3*B*C^2*a*b^10*c
^10*f + 2*A*B^2*a^11*c*d^9*f - 132*A^2*B*a*b^10*d^10*f - 3*A*B^2*a^10*b*d^1
0*f + 3*A^2*B*a*b^10*c^10*f + 520*C^3*a^5*b^6*c^3*d^7*f + 460*C^3*a^5*b^6*c
^5*d^5*f - 418*C^3*a^6*b^5*c^4*d^6*f + 406*C^3*a^4*b^7*c^6*d^4*f + 268*C^3*
a^7*b^4*c^5*d^5*f - 266*C^3*a^6*b^5*c^6*d^4*f + 233*C^3*a^8*b^3*c^2*d^8*f -
 176*C^3*a^5*b^6*c^7*d^3*f + 164*C^3*a^2*b^9*c^6*d^4*f + 140*C^3*a^6*b^5*c^
2*d^8*f + 136*C^3*a^2*b^9*c^4*d^6*f - 128*C^3*a^9*b^2*c^3*d^7*f + 128*C^3*a
^3*b^8*c^3*d^7*f - 108*C^3*a^8*b^3*c^6*d^4*f - 104*C^3*a^3*b^8*c^7*d^3*f -
104*C^3*a^3*b^8*c^5*d^5*f + 100*C^3*a^8*b^3*c^4*d^6*f - 89*C^3*a^2*b^9*c^8*
d^2*f - 72*C^3*a^9*b^2*c^5*d^5*f - 40*C^3*a^7*b^4*c^3*d^7*f + 40*C^3*a^4*b^
7*c^8*d^2*f - 28*C^3*a^4*b^7*c^2*d^8*f - 16*C^3*a^2*b^9*c^2*d^8*f - 2*C^3*a
^4*b^7*c^4*d^6*f + 828*B^3*a^4*b^7*c^5*d^5*f + 408*B^3*a^5*b^6*c^2*d^8*f +
390*B^3*a^7*b^4*c^4*d^6*f - 372*B^3*a^3*b^8*c^4*d^6*f - 336*B^3*a^6*b^5*c^3
*d^7*f - 314*B^3*a^5*b^6*c^6*d^4*f + 288*B^3*a^4*b^7*c^3*d^7*f + 216*B^3*a^
7*b^4*c^2*d^8*f - 176*B^3*a^2*b^9*c^7*d^3*f + 128*B^3*a^2*b^9*c^3*d^7*f + 1
08*B^3*a^6*b^5*c^5*d^5*f + 88*B^3*a^4*b^7*c^7*d^3*f + 72*B^3*a^2*b^9*c^5*d^
5*f - 68*B^3*a^3*b^8*c^2*d^8*f - 65*B^3*a^9*b^2*c^2*d^8*f - 56*B^3*a^8*b^3*
c^5*d^5*f + 40*B^3*a^6*b^5*c^7*d^3*f + 37*B^3*a^3*b^8*c^8*d^2*f + 30*B^3*a^
5*b^6*c^4*d^6*f - 28*B^3*a^5*b^6*c^8*d^2*f + 24*B^3*a^8*b^3*c^3*d^7*f - 4*B
^3*a^9*b^2*c^4*d^6*f - 2*B^3*a^7*b^4*c^6*d^4*f + 1586*A^3*a^4*b^7*c^4*d^6*f
 - 1376*A^3*a^3*b^8*c^3*d^7*f - 1096*A^3*a^5*b^6*c^3*d^7*f + 844*A^3*a^4*b^
7*c^2*d^8*f - 748*A^3*a^5*b^6*c^5*d^5*f + 490*A^3*a^6*b^5*c^4*d^6*f + 376*A
^3*a^2*b^9*c^2*d^8*f + 362*A^3*a^4*b^7*c^6*d^4*f - 356*A^3*a^6*b^5*c^2*d^8*
f + 328*A^3*a^7*b^4*c^3*d^7*f - 328*A^3*a^3*b^8*c^5*d^5*f + 224*A^3*a^2*b^9
*c^4*d^6*f - 197*A^3*a^8*b^3*c^2*d^8*f - 112*A^3*a^5*b^6*c^7*d^3*f + 98*A^3
*a^6*b^5*c^6*d^4*f - 92*A^3*a^2*b^9*c^6*d^4*f - 88*A^3*a^3*b^8*c^7*d^3*f +
68*A^3*a^4*b^7*c^8*d^2*f + 32*A^3*a^9*b^2*c^3*d^7*f - 28*A^3*a^8*b^3*c^4*d^
6*f - 28*A^3*a^7*b^4*c^5*d^5*f + 17*A^3*a^2*b^9*c^8*d^2*f + 104*C^3*a*b^10*

c^7*d^3*f + 54*C^3*a^9*b^2*c*d^9*f - 40*C^3*a^7*b^4*c*d^9*f - 35*C^3*a^10*b
*c^2*d^8*f + 22*C^3*a^3*b^8*c^9*d*f + 16*C^3*a*b^10*c^5*d^5*f - 16*C^3*a*b^
10*c^3*d^7*f + 8*C^3*a^5*b^6*c*d^9*f - 2*A*B*C*a^11*d^10*f + 198*B^3*a^8*b^
3*c*d^9*f + 192*B^3*a*b^10*c^6*d^4*f - 128*B^3*a^4*b^7*c*d^9*f - 80*B^3*a*b
^10*c^2*d^8*f - 56*B^3*a^2*b^9*c*d^9*f - 24*B^3*a^6*b^5*c*d^9*f - 18*B^3*a^
2*b^9*c^9*d*f - 16*B^3*a*b^10*c^4*d^6*f + 13*B^3*a*b^10*c^8*d^2*f + 8*B^3*a
^10*b*c^3*d^7*f + 8*B^3*a^4*b^7*c^9*d*f - 624*A^3*a^3*b^8*c*d^9*f + 472*A^3
*a^7*b^4*c*d^9*f - 272*A^3*a*b^10*c^3*d^7*f + 152*A^3*a*b^10*c^5*d^5*f - 22
*A^3*a^3*b^8*c^9*d*f + 18*A^3*a^9*b^2*c*d^9*f - 13*A^3*a^10*b*c^2*d^8*f - 8
*A^3*a^5*b^6*c*d^9*f - 8*A^3*a*b^10*c^7*d^3*f + A*B^2*b^11*c^8*d^2*f + 11*C
^3*b^11*c^8*d^2*f - 8*C^3*b^11*c^6*d^4*f - 4*C^3*b^11*c^4*d^6*f - 64*B^3*b^
11*c^5*d^5*f - 32*B^3*b^11*c^3*d^7*f - 68*A^3*b^11*c^4*d^6*f + 20*A^3*b^11*
c^6*d^4*f + 12*A^3*b^11*c^2*d^8*f - C^3*a^8*b^3*d^10*f - B^3*a^11*c^2*d^8*f
 - 60*B^3*a^7*b^4*d^10*f - 32*B^3*a^5*b^6*d^10*f + 21*B^3*a^9*b^2*d^10*f -
12*B^3*a^3*b^8*d^10*f - 3*C^3*a^2*b^9*c^10*f - 360*A^3*a^6*b^5*d^10*f - 204
*A^3*a^4*b^7*d^10*f - B^3*a^3*b^8*c^10*f + 3*A^3*a^2*b^9*c^10*f - 2*C^3*a^1
1*c*d^9*f - 2*B^3*b^11*c^9*d*f + 3*C^3*a^10*b*d^10*f + 2*A^3*a^11*c*d^9*f +
 3*B^3*a*b^10*c^10*f - 3*A^3*a^10*b*d^10*f - 36*A^2*C*b^11*d^10*f + 3*A^2*C
*b^11*c^10*f - 3*A*C^2*b^11*c^10*f - A*B^2*b^11*c^10*f + 36*A^3*b^11*d^10*f
 - A^3*b^11*c^10*f + A^3*b^11*c^8*d^2*f + A^3*a^8*b^3*d^10*f + B^2*C*b^11*c
^10*f + B*C^2*a^11*d^10*f + A^2*B*a^11*d^10*f + C^3*b^11*c^10*f + B^3*a^11*
d^10*f - 6*A*B^2*C*a^7*b*c*d^7 + 4*A*B^2*C*a*b^7*c*d^7 + 168*A^2*B*C*a^2*b^
6*c^3*d^5 + 144*A*B*C^2*a^3*b^5*c^4*d^4 - 129*A^2*B*C*a^3*b^5*c^4*d^4 - 96*
A*B*C^2*a^2*b^6*c^3*d^5 + 84*A*B*C^2*a^3*b^5*c^2*d^6 + 72*A^2*B*C*a^4*b^4*c
^3*d^5 - 72*A^2*B*C*a^3*b^5*c^2*d^6 + 64*A*B^2*C*a^4*b^4*c^4*d^4 - 60*A*B*C
^2*a^4*b^4*c^3*d^5 + 57*A^2*B*C*a^5*b^3*c^2*d^6 - 56*A*B^2*C*a^5*b^3*c^3*d^
5 - 39*A*B^2*C*a^2*b^6*c^4*d^4 - 38*A*B^2*C*a^3*b^5*c^5*d^3 + 36*A*B^2*C*a^
3*b^5*c^3*d^5 + 36*A*B*C^2*a^5*b^3*c^4*d^4 - 30*A*B*C^2*a^5*b^3*c^2*d^6 + 2
7*A*B^2*C*a^6*b^2*c^2*d^6 - 24*A*B^2*C*a^2*b^6*c^2*d^6 + 24*A*B*C^2*a^6*b^2
*c^3*d^5 - 24*A*B*C^2*a^4*b^4*c^5*d^3 - 18*A^2*B*C*a^5*b^3*c^4*d^4 + 18*A^2
*B*C*a^2*b^6*c^5*d^3 - 15*A*B^2*C*a^4*b^4*c^2*d^6 - 12*A^2*B*C*a^6*b^2*c^3*
d^5 + 12*A^2*B*C*a^4*b^4*c^5*d^3 + 9*A*B^2*C*a^2*b^6*c^6*d^2 + 6*A*B*C^2*a^
3*b^5*c^6*d^2 - 3*A^2*B*C*a^3*b^5*c^6*d^2 + 60*A^2*B*C*a^2*b^6*c*d^7 - 51*A
^2*B*C*a*b^7*c^4*d^4 + 48*A*B*C^2*a^6*b^2*c*d^7 - 42*A^2*B*C*a^6*b^2*c*d^7
 - 42*A^2*B*C*a*b^7*c^2*d^6 + 36*A*B*C^2*a^4*b^4*c*d^7 + 36*A*B*C^2*a*b^7*c^
4*d^4 + 36*A*B*C^2*a*b^7*c^2*d^6 - 30*A^2*B*C*a^4*b^4*c*d^7 + 24*A*B^2*C*a^
3*b^5*c*d^7 - 24*A*B*C^2*a^2*b^6*c*d^7 + 18*A*B^2*C*a*b^7*c^5*d^3 - 18*A*B*
C^2*a*b^7*c^6*d^2 + 12*A*B^2*C*a*b^7*c^3*d^5 + 9*A^2*B*C*a*b^7*c^6*d^2 + 6*
A*B^2*C*a^5*b^3*c*d^7 - 6*A*B*C^2*a^7*b*c^2*d^6 + 3*A^2*B*C*a^7*b*c^2*d^6 -
 18*B^3*C*a^6*b^2*c*d^7 - 18*B*C^3*a^6*b^2*c*d^7 - 14*B^3*C*a^4*b^4*c*d^7 -
 14*B*C^3*a^4*b^4*c*d^7 - 10*B^3*C*a*b^7*c^2*d^6 - 10*B*C^3*a*b^7*c^2*d^6 +
 9*B^3*C*a*b^7*c^6*d^2 + 9*B*C^3*a*b^7*c^6*d^2 - 7*B^3*C*a*b^7*c^4*d^4 - 7*
B*C^3*a*b^7*c^4*d^4 + 6*B^2*C^2*a^7*b*c*d^7 - 4*B^3*C*a^2*b^6*c*d^7 + 4*B^2
*C^2*a*b^7*c*d^7 - 4*B*C^3*a^2*b^6*c*d^7 + 3*B^3*C*a^7*b*c^2*d^6 + 3*B*C^3*
a^7*b*c^2*d^6 + 144*A^3*C*a^3*b^5*c*d^7 + 62*A^3*C*a^5*b^3*c*d^7 + 48*A*C^3

*a^3*b^5*c*d^7 - 36*A^2*C^2*a*b^7*c*d^7 + 26*A*C^3*a^5*b^3*c*d^7 + 20*A^3*C
*a*b^7*c^3*d^5 + 18*A^2*C^2*a^7*b*c*d^7 - 18*A*C^3*a*b^7*c^5*d^3 - 6*A^3*C*
a*b^7*c^5*d^3 - 4*A*C^3*a*b^7*c^3*d^5 - 32*A^3*B*a^2*b^6*c*d^7 - 32*A*B^3*a
^2*b^6*c*d^7 + 22*A^3*B*a*b^7*c^4*d^4 + 22*A*B^3*a*b^7*c^4*d^4 + 16*A^3*B*a
*b^7*c^2*d^6 + 16*A*B^3*a*b^7*c^2*d^6 + 12*A^3*B*a^6*b^2*c*d^7 + 12*A*B^3*a
^6*b^2*c*d^7 + 8*A^3*B*a^4*b^4*c*d^7 - 8*A^2*B^2*a*b^7*c*d^7 + 8*A*B^3*a^4*
b^4*c*d^7 + 36*A^2*B*C*b^8*c^3*d^5 + 24*A*B*C^2*b^8*c^5*d^3 - 18*A^2*B*C*b^
8*c^5*d^3 - 12*A*B*C^2*b^8*c^3*d^5 - 3*A*B^2*C*b^8*c^6*d^2 - 3*A*B^2*C*b^8*
c^4*d^4 - 2*A*B^2*C*b^8*c^2*d^6 + 57*A^2*B*C*a^5*b^3*d^8 + 36*A^2*B*C*a^3*b
^5*d^8 - 30*A*B*C^2*a^5*b^3*d^8 - 18*A*B*C^2*a^3*b^5*d^8 - 9*A*B^2*C*a^4*b^
4*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 2*A*B^2*C*a^2*b^6*d^8 + 34*B^2*C^2*a^3*b^5*
c^5*d^3 + 28*B^2*C^2*a^5*b^3*c^3*d^5 + 24*B^2*C^2*a^2*b^6*c^4*d^4 - 20*B^2*
C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*d^5 + 12*B^2*C^2*a^2*b^6*c^2*d
^6 + 9*B^2*C^2*a^6*b^2*c^4*d^4 + 9*B^2*C^2*a^4*b^4*c^2*d^6 - 9*B^2*C^2*a^2*
b^6*c^6*d^2 - 3*B^2*C^2*a^6*b^2*c^2*d^6 + 159*A^2*C^2*a^4*b^4*c^2*d^6 - 156
*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^3*b^5*c^5*d^3 + 78*A^2*C^2*a^2*b^6*
c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2*C^2*a^6*b^2*c^2*d^6 - 27*A^2*
C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^2*b^6*c^4*d^4 + 9*A^2*C^2*a^6*b^2*c^4*d^
4 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a^4*b^4*c^2*d^6 - 48*A^2*B^2*a^
3*b^5*c^3*d^5 + 42*A^2*B^2*a^2*b^6*c^4*d^4 + 28*A^2*B^2*a^5*b^3*c^3*d^5 - 1
7*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^6*b^2*c^2*d^6 + 4*A^2*B^2*a^3*b^5*c
^5*d^3 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a^7*b*c*d^7 + 12*A*C^3*a*b^7*c*d^7
 - 6*A^3*C*a^7*b*c*d^7 + 24*A^2*B*C*b^8*c*d^7 - 12*A*B*C^2*b^8*c*d^7 + 12*A
^2*B*C*a*b^7*d^8 + 6*A*B*C^2*a^7*b*d^8 - 6*A*B*C^2*a*b^7*d^8 - 3*A^2*B*C*a^
7*b*d^8 - 53*B^3*C*a^3*b^5*c^4*d^4 - 53*B*C^3*a^3*b^5*c^4*d^4 - 32*B^3*C*a^
3*b^5*c^2*d^6 - 32*B*C^3*a^3*b^5*c^2*d^6 - 18*B^3*C*a^5*b^3*c^4*d^4 - 18*B*
C^3*a^5*b^3*c^4*d^4 + 16*B^3*C*a^4*b^4*c^3*d^5 + 16*B*C^3*a^4*b^4*c^3*d^5 -
 12*B^3*C*a^6*b^2*c^3*d^5 + 12*B^3*C*a^4*b^4*c^5*d^3 + 12*B^2*C^2*a^3*b^5*c
*d^7 - 12*B*C^3*a^6*b^2*c^3*d^5 + 12*B*C^3*a^4*b^4*c^5*d^3 + 8*B^3*C*a^2*b^
6*c^3*d^5 + 8*B*C^3*a^2*b^6*c^3*d^5 - 6*B^3*C*a^2*b^6*c^5*d^3 + 6*B^2*C^2*a
^5*b^3*c*d^7 - 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*C^3*a^2*b^6*c^5*d^3 - 3*B^3*C*
a^3*b^5*c^6*d^2 - 3*B*C^3*a^3*b^5*c^6*d^2 - 175*A^3*C*a^4*b^4*c^2*d^6 + 164
*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a^3*b^5*c*d^7 - 124*A^3*C*a^2*b^6*c^2*
d^6 - 90*A*C^3*a^3*b^5*c^5*d^3 - 73*A*C^3*a^4*b^4*c^2*d^6 - 66*A^2*C^2*a^5*
b^3*c*d^7 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C^3*a^4*b^4*c^4*d^4 + 30*A^3*C*
a^4*b^4*c^4*d^4 - 30*A^3*C*a^3*b^5*c^5*d^3 + 27*A*C^3*a^2*b^6*c^6*d^2 + 21*
A*C^3*a^2*b^6*c^4*d^4 + 18*A^2*C^2*a*b^7*c^5*d^3 - 18*A*C^3*a^6*b^2*c^4*d^4
 - 16*A*C^3*a^2*b^6*c^2*d^6 + 15*A^3*C*a^6*b^2*c^2*d^6 - 15*A^3*C*a^2*b^6*c
^4*d^4 - 12*A^2*C^2*a*b^7*c^3*d^5 + 9*A^3*C*a^2*b^6*c^6*d^2 + 9*A*C^3*a^6*b
^2*c^2*d^6 - 80*A^3*B*a^2*b^6*c^3*d^5 - 80*A*B^3*a^2*b^6*c^3*d^5 + 38*A^3*B
*a^3*b^5*c^4*d^4 + 38*A*B^3*a^3*b^5*c^4*d^4 - 36*A^2*B^2*a^3*b^5*c*d^7 - 28
*A^3*B*a^5*b^3*c^2*d^6 - 28*A^3*B*a^4*b^4*c^3*d^5 - 28*A*B^3*a^5*b^3*c^2*d^
6 - 28*A*B^3*a^4*b^4*c^3*d^5 + 20*A^3*B*a^3*b^5*c^2*d^6 + 20*A*B^3*a^3*b^5*
c^2*d^6 - 12*A^3*B*a^2*b^6*c^5*d^3 - 12*A^2*B^2*a^5*b^3*c*d^7 - 12*A^2*B^2*
a*b^7*c^5*d^3 - 12*A^2*B^2*a*b^7*c^3*d^5 - 12*A*B^3*a^2*b^6*c^5*d^3 + 9*B^2

*C^2*b^8*c^4*d^4 + 4*B^2*C^2*b^8*c^2*d^6 + 3*B^2*C^2*b^8*c^6*d^2 - 30*A^2*C
^2*b^8*c^4*d^4 + 9*A^2*C^2*b^8*c^6*d^2 + 16*A^2*B^2*b^8*c^2*d^6 + 6*B^2*C^2
*a^6*b^2*d^8 + 3*B^2*C^2*a^4*b^4*d^8 + 3*A^2*B^2*b^8*c^4*d^4 + 36*A^2*C^2*a
^4*b^4*d^8 + 27*A^2*C^2*a^2*b^6*d^8 - 18*A^2*C^2*a^6*b^2*d^8 + 33*A^2*B^2*a
^4*b^4*d^8 + 28*A^2*B^2*a^2*b^6*d^8 + 6*A^2*B^2*a^6*b^2*d^8 + 6*C^4*a*b^7*c
^5*d^3 + 4*C^4*a*b^7*c^3*d^5 - 2*C^4*a^5*b^3*c*d^7 + 12*B^4*a^3*b^5*c*d^7 -
12*B^4*a*b^7*c^5*d^3 + 8*B^4*a^5*b^3*c*d^7 - 4*B^4*a*b^7*c^3*d^5 - 48*A^4*
a^3*b^5*c*d^7 - 20*A^4*a^5*b^3*c*d^7 - 8*A^4*a*b^7*c^3*d^5 - 10*B^3*C*b^8*c
^5*d^3 - 10*B*C^3*b^8*c^5*d^3 - 4*B^3*C*b^8*c^3*d^5 - 4*B*C^3*b^8*c^3*d^5 +
23*A^3*C*b^8*c^4*d^4 - 18*A^3*C*b^8*c^2*d^6 + 11*A*C^3*b^8*c^4*d^4 - 9*A*C
^3*b^8*c^6*d^2 + 6*A*C^3*b^8*c^2*d^6 - 3*A^3*C*b^8*c^6*d^2 - 20*A^3*B*b^8*c
^3*d^5 - 20*A*B^3*b^8*c^3*d^5 + 4*A^3*B*b^8*c^5*d^3 + 4*A*B^3*b^8*c^5*d^3 -
63*A^3*C*a^4*b^4*d^8 - 54*A^3*C*a^2*b^6*d^8 + 9*A^3*C*a^6*b^2*d^8 + 9*A*C^
3*a^6*b^2*d^8 - 3*A*C^3*a^4*b^4*d^8 - 28*A^3*B*a^5*b^3*d^8 - 28*A*B^3*a^5*b
^3*d^8 - 18*A^3*B*a^3*b^5*d^8 - 18*A*B^3*a^3*b^5*d^8 + B^3*C*a^5*b^3*c^2*d^
6 + B*C^3*a^5*b^3*c^2*d^6 + 6*C^4*a^7*b*c*d^7 + 4*B^4*a*b^7*c*d^7 - 12*A^4*
a*b^7*c*d^7 - 12*A^3*B*b^8*c*d^7 - 12*A*B^3*b^8*c*d^7 - 3*B^3*C*a^7*b*d^8 -
3*B*C^3*a^7*b*d^8 - 6*A^3*B*a*b^7*d^8 - 6*A*B^3*a*b^7*d^8 + 30*C^4*a^3*b^5
*c^5*d^3 + 19*C^4*a^4*b^4*c^2*d^6 + 9*C^4*a^6*b^2*c^4*d^4 - 9*C^4*a^2*b^6*c
^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*b^6*c^2*d^6 + 3*C^4*a^6*b^2*c^2*
d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^2*b^6*c^4*d^4 + 28*B^4*a^5*b^3*c^3*d^
5 + 27*B^4*a^2*b^6*c^4*d^4 - 17*B^4*a^4*b^4*c^4*d^4 - 10*B^4*a^4*b^4*c^2*d^
6 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c^2*d^6 - 6*B^4*a^6*b^2*c^2*d^6 +
4*B^4*a^3*b^5*c^5*d^3 + 70*A^4*a^4*b^4*c^2*d^6 + 58*A^4*a^2*b^6*c^2*d^6 -
56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^2*b^6*c^4*d^4 + B^2*C^2*a^2*b^6*d^8 - 18*
A^3*C*b^8*d^8 + B^3*C*a^5*b^3*d^8 + B*C^3*a^5*b^3*d^8 + 3*C^4*b^8*c^6*d^2 +
8*B^4*b^8*c^4*d^4 + 4*B^4*b^8*c^2*d^6 + 12*A^4*b^8*c^2*d^6 - 5*A^4*b^8*c^4
*d^4 + 6*B^4*a^6*b^2*d^8 + 3*B^4*a^4*b^4*d^8 + 30*A^4*a^4*b^4*d^8 + 27*A^4*
a^2*b^6*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2*b^8*d^8 + 9*A^4*b^8*d^8 + C^4*b
^8*c^4*d^4 + B^4*a^2*b^6*d^8, f, k)*((4*a^7*b^12*d^15 + 12*a^9*b^10*d^15 +
8*a^11*b^8*d^15 - 8*a^13*b^6*d^15 - 12*a^15*b^4*d^15 - 4*a^17*b^2*d^15 + 4*
b^19*c^7*d^8 + 4*b^19*c^9*d^6 - 4*b^19*c^11*d^4 - 4*b^19*c^13*d^2 - 20*a*b^
18*c^6*d^9 - 4*a*b^18*c^8*d^7 + 60*a*b^18*c^10*d^5 + 52*a*b^18*c^12*d^3 + 3
2*a^3*b^16*c^14*d + 48*a^5*b^14*c^14*d - 20*a^6*b^13*c*d^14 + 32*a^7*b^12*c
^14*d - 44*a^8*b^11*c*d^14 + 8*a^9*b^10*c^14*d + 32*a^10*b^9*c*d^14 + 168*a
^12*b^7*c*d^14 + 172*a^14*b^5*c*d^14 + 68*a^16*b^3*c*d^14 + 16*a^18*b*c^3*d
^12 + 8*a^18*b*c^5*d^10 + 36*a^2*b^17*c^5*d^10 - 32*a^2*b^17*c^7*d^8 - 240*
a^2*b^17*c^9*d^6 - 240*a^2*b^17*c^11*d^4 - 68*a^2*b^17*c^13*d^2 - 20*a^3*b^
16*c^4*d^11 + 64*a^3*b^16*c^6*d^9 + 472*a^3*b^16*c^8*d^7 + 704*a^3*b^16*c^1
0*d^5 + 348*a^3*b^16*c^12*d^3 - 20*a^4*b^15*c^3*d^12 + 8*a^4*b^15*c^5*d^10
- 568*a^4*b^15*c^7*d^8 - 1472*a^4*b^15*c^9*d^6 - 1108*a^4*b^15*c^11*d^4 - 2
32*a^4*b^15*c^13*d^2 + 36*a^5*b^14*c^2*d^13 - 104*a^5*b^14*c^4*d^11 + 392*a
^5*b^14*c^6*d^9 + 2016*a^5*b^14*c^8*d^7 + 2308*a^5*b^14*c^10*d^5 + 872*a^5*
b^14*c^12*d^3 + 64*a^6*b^13*c^3*d^12 + 112*a^6*b^13*c^5*d^10 - 1504*a^6*b^1
3*c^7*d^8 - 3316*a^6*b^13*c^9*d^6 - 2112*a^6*b^13*c^11*d^4 - 328*a^6*b^13*c

^13*d^2 + 32*a^7*b^12*c^2*d^13 - 640*a^7*b^12*c^4*d^11 + 32*a^7*b^12*c^6*d^9 + 3076*a^7*b^12*c^8*d^7 + 3392*a^7*b^12*c^10*d^5 + 1048*a^7*b^12*c^12*d^3 + 668*a^8*b^11*c^3*d^12 + 1404*a^8*b^11*c^5*d^10 - 976*a^8*b^11*c^7*d^8 - 3484*a^8*b^11*c^9*d^6 - 2028*a^8*b^11*c^11*d^4 - 212*a^8*b^11*c^13*d^2 - 292*a^9*b^10*c^2*d^13 - 2028*a^9*b^10*c^4*d^11 - 1864*a^9*b^10*c^6*d^9 + 1724*a^9*b^10*c^8*d^7 + 2468*a^9*b^10*c^10*d^5 + 612*a^9*b^10*c^12*d^3 + 1648*a^10*b^9*c^3*d^12 + 3404*a^10*b^9*c^5*d^10 + 1120*a^10*b^9*c^7*d^8 - 1592*a^10*b^9*c^9*d^6 - 976*a^10*b^9*c^11*d^4 - 52*a^10*b^9*c^13*d^2 - 768*a^11*b^8*c^2*d^13 - 3092*a^11*b^8*c^4*d^11 - 3296*a^11*b^8*c^6*d^9 - 288*a^11*b^8*c^8*d^7 + 832*a^11*b^8*c^10*d^5 + 140*a^11*b^8*c^12*d^3 + 1892*a^12*b^7*c^3*d^12 + 3552*a^12*b^7*c^5*d^10 + 1912*a^12*b^7*c^7*d^8 - 104*a^12*b^7*c^9*d^6 - 188*a^12*b^7*c^11*d^4 - 772*a^13*b^6*c^2*d^13 - 2368*a^13*b^6*c^4*d^11 - 2360*a^13*b^6*c^6*d^9 - 664*a^13*b^6*c^8*d^7 + 92*a^13*b^6*c^10*d^5 + 1088*a^14*b^5*c^3*d^12 + 1752*a^14*b^5*c^5*d^10 + 928*a^14*b^5*c^7*d^8 + 92*a^14*b^5*c^9*d^6 - 352*a^15*b^4*c^2*d^13 - 856*a^15*b^4*c^4*d^11 - 704*a^15*b^4*c^6*d^9 - 188*a^15*b^4*c^8*d^7 + 276*a^16*b^3*c^3*d^12 + 348*a^16*b^3*c^5*d^10 + 140*a^16*b^3*c^7*d^8 - 60*a^17*b^2*c^2*d^13 - 108*a^17*b^2*c^4*d^11 - 52*a^17*b^2*c^6*d^9 + 8*a*b^18*c^14*d + 8*a^18*b*c*d^14)/(a^14*d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^11*c^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8 + 90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^5*b^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 152*a^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 68*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12*b^2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*a^13*b*c*d^9) + (tan(e + f*x)*(6*a^18*b*d^15 + 6*b^19*c^14*d + 8*a^6*b^13*d^15 + 38*a^8*b^11*d^15 + 78*a^10*b^9*d^15 + 92*a^12*b^7*d^15 + 68*a^14*b^5*d^15 + 30*a^16*b^3*d^15 + 8*b^19*c^6*d^9 + 22*b^19*c^8*d^7 + 26*b^19*c^10*d^5 + 18*b^19*c^12*d^3 - 48*a*b^18*c^5*d^10 - 128*a*b^18*c^7*d^8 - 144*a*b^18*c^9*d^6 - 96*a*b^18*c^11*d^4 - 32*a*b^18*c^13*d^2 + 22*a^2*b^17*c^14*d + 28*a^4*b^15*c^14*d - 48*a^5*b^14*c*d^14 + 12*a^6*b^13*c^14*d - 224*a^7*b^12*c*d^14 - 2*a^8*b^11*c^14*d - 448*a^9*b^10*c*d^14 - 2*a^10*b^9*c^14*d - 512*a^11*b^8*c*d^14 - 368*a^13*b^6*c*d^14 - 160*a^15*b^4*c*d^14 - 32*a^17*b^2*c*d^14 + 10*a^18*b*c^2*d^13 + 2*a^18*b*c^4*d^11 - 2*a^18*b*c^6*d^9 + 120*a^2*b^17*c^4*d^11 + 344*a^2*b^17*c^6*d^9 + 406*a^2*b^17*c^8*d^7 + 282*a^2*b^17*c^

10*d^5 + 122*a^2*b^17*c^12*d^3 - 160*a^3*b^16*c^3*d^12 - 608*a^3*b^16*c^5*d
^10 - 848*a^3*b^16*c^7*d^8 - 624*a^3*b^16*c^9*d^6 - 336*a^3*b^16*c^11*d^4 -
 112*a^3*b^16*c^13*d^2 + 120*a^4*b^15*c^2*d^13 + 820*a^4*b^15*c^4*d^11 + 14
28*a^4*b^15*c^6*d^9 + 1072*a^4*b^15*c^8*d^7 + 568*a^4*b^15*c^10*d^5 + 252*a
^4*b^15*c^12*d^3 - 832*a^5*b^14*c^3*d^12 - 1904*a^5*b^14*c^5*d^10 - 1520*a^
5*b^14*c^7*d^8 - 544*a^5*b^14*c^9*d^6 - 272*a^5*b^14*c^11*d^4 - 128*a^5*b^1
4*c^13*d^2 + 568*a^6*b^13*c^2*d^13 + 2044*a^6*b^13*c^4*d^11 + 1988*a^6*b^13
*c^6*d^9 + 200*a^6*b^13*c^8*d^7 - 168*a^6*b^13*c^10*d^5 + 148*a^6*b^13*c^12
*d^3 - 1776*a^7*b^12*c^3*d^12 - 2384*a^7*b^12*c^5*d^10 + 80*a^7*b^12*c^7*d^
8 + 1296*a^7*b^12*c^9*d^6 + 352*a^7*b^12*c^11*d^4 - 32*a^7*b^12*c^13*d^2 +
1138*a^8*b^11*c^2*d^13 + 2434*a^8*b^11*c^4*d^11 + 214*a^8*b^11*c^6*d^9 - 26
26*a^8*b^11*c^8*d^7 - 1622*a^8*b^11*c^10*d^5 - 118*a^8*b^11*c^12*d^3 - 2032
*a^9*b^10*c^3*d^12 - 976*a^9*b^10*c^5*d^10 + 3056*a^9*b^10*c^7*d^8 + 3184*a
^9*b^10*c^9*d^6 + 768*a^9*b^10*c^11*d^4 + 32*a^9*b^10*c^13*d^2 + 1282*a^10*
b^9*c^2*d^13 + 1498*a^10*b^9*c^4*d^11 - 2058*a^10*b^9*c^6*d^9 - 4042*a^10*b
^9*c^8*d^7 - 1862*a^10*b^9*c^10*d^5 - 174*a^10*b^9*c^12*d^3 - 1408*a^11*b^8
*c^3*d^12 + 448*a^11*b^8*c^5*d^10 + 3536*a^11*b^8*c^7*d^8 + 2672*a^11*b^8*c
^9*d^6 + 496*a^11*b^8*c^11*d^4 + 16*a^11*b^8*c^13*d^2 + 908*a^12*b^7*c^2*d^
13 + 552*a^12*b^7*c^4*d^11 - 2000*a^12*b^7*c^6*d^9 - 2540*a^12*b^7*c^8*d^7
- 860*a^12*b^7*c^10*d^5 - 56*a^12*b^7*c^12*d^3 - 672*a^13*b^6*c^3*d^12 + 49
6*a^13*b^6*c^5*d^10 + 1648*a^13*b^6*c^7*d^8 + 960*a^13*b^6*c^9*d^6 + 112*a^
13*b^6*c^11*d^4 + 412*a^14*b^5*c^2*d^13 + 208*a^14*b^5*c^4*d^11 - 688*a^14*
b^5*c^6*d^9 - 692*a^14*b^5*c^8*d^7 - 140*a^14*b^5*c^10*d^5 - 240*a^15*b^4*c
^3*d^12 + 112*a^15*b^4*c^5*d^10 + 304*a^15*b^4*c^7*d^8 + 112*a^15*b^4*c^9*d
^6 + 106*a^16*b^3*c^2*d^13 + 66*a^16*b^3*c^4*d^11 - 66*a^16*b^3*c^6*d^9 - 5
6*a^16*b^3*c^8*d^7 - 48*a^17*b^2*c^3*d^12 + 16*a^17*b^2*c^7*d^8))/(a^14*d^1
0 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8*b^
6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^10
+ 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*
c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*
b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*
b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a
^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^11*c
^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8 +
90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^5*b
^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8
 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7
*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d
^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 152*a
^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2
*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 6
8*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12*b^
2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*
a^13*b*c*d^9)) - (C*a^15*b*d^13 - A*a^15*b*d^13 - B*b^16*c^12*d - 12*A*a^3*
b^13*d^13 - 48*A*a^5*b^11*d^13 - 76*A*a^7*b^9*d^13 - 45*A*a^9*b^7*d^13 + 5*

A*a^11*b^5*d^13 + 9*A*a^13*b^3*d^13 + 4*B*a^4*b^12*d^13 + 16*B*a^6*b^10*d^1
3 + 35*B*a^8*b^8*d^13 + 33*B*a^10*b^6*d^13 + 5*B*a^12*b^4*d^13 - 5*B*a^14*b
^2*d^13 + 12*A*b^16*c^3*d^10 + 20*A*b^16*c^5*d^8 - 4*A*b^16*c^9*d^4 + 4*A*b
^16*c^11*d^2 + 4*C*a^7*b^9*d^13 - 3*C*a^9*b^7*d^13 - 17*C*a^11*b^5*d^13 - 9
*C*a^13*b^3*d^13 - 8*B*b^16*c^4*d^9 - 16*B*b^16*c^6*d^7 - B*b^16*c^8*d^5 +
6*B*b^16*c^10*d^3 + 4*C*b^16*c^5*d^8 + 12*C*b^16*c^7*d^6 + 4*C*b^16*c^9*d^4
 - 4*C*b^16*c^11*d^2 - 36*A*a*b^15*c^2*d^11 - 92*A*a*b^15*c^4*d^9 - 56*A*a*
b^15*c^6*d^7 + 3*A*a*b^15*c^8*d^5 + 2*A*a*b^15*c^10*d^3 + 36*A*a^2*b^14*c*d
^12 - 3*A*a^3*b^13*c^12*d + 176*A*a^4*b^12*c*d^12 - 3*A*a^5*b^11*c^12*d + 3
80*A*a^6*b^10*c*d^12 - A*a^7*b^9*c^12*d + 396*A*a^8*b^8*c*d^12 + 176*A*a^10
*b^6*c*d^12 + 20*A*a^12*b^4*c*d^12 - 2*A*a^15*b*c^2*d^11 - A*a^15*b*c^4*d^9
 + 20*B*a*b^15*c^3*d^10 + 68*B*a*b^15*c^5*d^8 + 56*B*a*b^15*c^7*d^6 + 4*B*a
*b^15*c^9*d^4 - 4*B*a*b^15*c^11*d^2 - 3*B*a^2*b^14*c^12*d - 4*B*a^3*b^13*c*
d^12 - 3*B*a^4*b^12*c^12*d - 24*B*a^5*b^11*c*d^12 - B*a^6*b^10*c^12*d - 116
*B*a^7*b^9*c*d^12 - 196*B*a^9*b^7*c*d^12 - 120*B*a^11*b^5*c*d^12 - 20*B*a^1
3*b^3*c*d^12 - 4*C*a*b^15*c^4*d^9 - 40*C*a*b^15*c^6*d^7 - 51*C*a*b^15*c^8*d
^5 - 14*C*a*b^15*c^10*d^3 + 3*C*a^3*b^13*c^12*d - 8*C*a^4*b^12*c*d^12 + 3*C
*a^5*b^11*c^12*d - 56*C*a^6*b^10*c*d^12 + C*a^7*b^9*c^12*d - 60*C*a^8*b^8*c
*d^12 + 28*C*a^10*b^6*c*d^12 + 52*C*a^12*b^4*c*d^12 + 12*C*a^14*b^2*c*d^12
+ 2*C*a^15*b*c^2*d^11 + C*a^15*b*c^4*d^9 + 204*A*a^2*b^14*c^3*d^10 + 264*A*
a^2*b^14*c^5*d^8 + 24*A*a^2*b^14*c^7*d^6 - 68*A*a^2*b^14*c^9*d^4 + 4*A*a^2*
b^14*c^11*d^2 - 260*A*a^3*b^13*c^2*d^11 - 608*A*a^3*b^13*c^4*d^9 - 356*A*a^
3*b^13*c^6*d^7 + 33*A*a^3*b^13*c^8*d^5 + 26*A*a^3*b^13*c^10*d^3 + 876*A*a^4
*b^12*c^3*d^10 + 1180*A*a^4*b^12*c^5*d^8 + 368*A*a^4*b^12*c^7*d^6 - 108*A*a
^4*b^12*c^9*d^4 + 4*A*a^4*b^12*c^11*d^2 - 780*A*a^5*b^11*c^2*d^11 - 1866*A*
a^5*b^11*c^4*d^9 - 1320*A*a^5*b^11*c^6*d^7 - 165*A*a^5*b^11*c^8*d^5 + 18*A*
a^5*b^11*c^10*d^3 + 1812*A*a^6*b^10*c^3*d^10 + 2528*A*a^6*b^10*c^5*d^8 + 11
12*A*a^6*b^10*c^7*d^6 + 28*A*a^6*b^10*c^9*d^4 + 12*A*a^6*b^10*c^11*d^2 - 11
44*A*a^7*b^9*c^2*d^11 - 2802*A*a^7*b^9*c^4*d^9 - 2188*A*a^7*b^9*c^6*d^7 - 4
87*A*a^7*b^9*c^8*d^5 - 34*A*a^7*b^9*c^10*d^3 + 1872*A*a^8*b^8*c^3*d^10 + 26
28*A*a^8*b^8*c^5*d^8 + 1272*A*a^8*b^8*c^7*d^6 + 128*A*a^8*b^8*c^9*d^4 + 8*A
*a^8*b^8*c^11*d^2 - 798*A*a^9*b^7*c^2*d^11 - 2007*A*a^9*b^7*c^4*d^9 - 1588*
A*a^9*b^7*c^6*d^7 - 362*A*a^9*b^7*c^8*d^5 - 28*A*a^9*b^7*c^10*d^3 + 872*A*a
^10*b^6*c^3*d^10 + 1200*A*a^10*b^6*c^5*d^8 + 560*A*a^10*b^6*c^7*d^6 + 56*A*
a^10*b^6*c^9*d^4 - 202*A*a^11*b^5*c^2*d^11 - 585*A*a^11*b^5*c^4*d^9 - 448*A
*a^11*b^5*c^6*d^7 - 70*A*a^11*b^5*c^8*d^5 + 136*A*a^12*b^4*c^3*d^10 + 172*A
*a^12*b^4*c^5*d^8 + 56*A*a^12*b^4*c^7*d^6 + 6*A*a^13*b^3*c^2*d^11 - 31*A*a^
13*b^3*c^4*d^9 - 28*A*a^13*b^3*c^6*d^7 + 8*A*a^14*b^2*c^3*d^10 + 8*A*a^14*b
^2*c^5*d^8 - 12*B*a^2*b^14*c^2*d^11 - 132*B*a^2*b^14*c^4*d^9 - 244*B*a^2*b^
14*c^6*d^7 - 103*B*a^2*b^14*c^8*d^5 + 18*B*a^2*b^14*c^10*d^3 + 132*B*a^3*b^
13*c^3*d^10 + 496*B*a^3*b^13*c^5*d^8 + 488*B*a^3*b^13*c^7*d^6 + 132*B*a^3*b
^13*c^9*d^4 + 4*B*a^3*b^13*c^11*d^2 - 44*B*a^4*b^12*c^2*d^11 - 558*B*a^4*b^
12*c^4*d^9 - 1064*B*a^4*b^12*c^6*d^7 - 581*B*a^4*b^12*c^8*d^5 - 30*B*a^4*b^
12*c^10*d^3 + 284*B*a^5*b^11*c^3*d^10 + 1196*B*a^5*b^11*c^5*d^8 + 1224*B*a^
5*b^11*c^7*d^6 + 356*B*a^5*b^11*c^9*d^4 + 20*B*a^5*b^11*c^11*d^2 + 48*B*a^6

*b^10*c^2*d^11 - 694*B*a^6*b^10*c^4*d^9 - 1596*B*a^6*b^10*c^6*d^7 - 959*B*a
^6*b^10*c^8*d^5 - 90*B*a^6*b^10*c^10*d^3 + 28*B*a^7*b^9*c^3*d^10 + 1032*B*a
^7*b^9*c^5*d^8 + 1208*B*a^7*b^9*c^7*d^6 + 332*B*a^7*b^9*c^9*d^4 + 12*B*a^7*
b^9*c^11*d^2 + 302*B*a^8*b^8*c^2*d^11 - 27*B*a^8*b^8*c^4*d^9 - 828*B*a^8*b^
8*c^6*d^7 - 582*B*a^8*b^8*c^8*d^5 - 48*B*a^8*b^8*c^10*d^3 - 424*B*a^9*b^7*c
^3*d^10 + 84*B*a^9*b^7*c^5*d^8 + 416*B*a^9*b^7*c^7*d^6 + 104*B*a^9*b^7*c^9*
d^4 + 342*B*a^10*b^6*c^2*d^11 + 411*B*a^10*b^6*c^4*d^9 - 102*B*a^10*b^6*c^8
*d^5 - 336*B*a^11*b^5*c^3*d^10 - 216*B*a^11*b^5*c^5*d^8 + 118*B*a^12*b^4*c^
2*d^11 + 181*B*a^12*b^4*c^4*d^9 + 68*B*a^12*b^4*c^6*d^7 - 56*B*a^13*b^3*c^3
*d^10 - 36*B*a^13*b^3*c^5*d^8 - 2*B*a^14*b^2*c^2*d^11 + 3*B*a^14*b^2*c^4*d^
9 - 12*C*a^2*b^14*c^3*d^10 + 36*C*a^2*b^14*c^5*d^8 + 144*C*a^2*b^14*c^7*d^6
 + 92*C*a^2*b^14*c^9*d^4 - 4*C*a^2*b^14*c^11*d^2 + 20*C*a^3*b^13*c^2*d^11 +
 56*C*a^3*b^13*c^4*d^9 - 124*C*a^3*b^13*c^6*d^7 - 237*C*a^3*b^13*c^8*d^5 -
74*C*a^3*b^13*c^10*d^3 - 168*C*a^4*b^12*c^3*d^10 - 172*C*a^4*b^12*c^5*d^8 +
 196*C*a^4*b^12*c^7*d^6 + 204*C*a^4*b^12*c^9*d^4 - 4*C*a^4*b^12*c^11*d^2 +
156*C*a^5*b^11*c^2*d^11 + 570*C*a^5*b^11*c^4*d^9 + 336*C*a^5*b^11*c^6*d^7 -
 171*C*a^5*b^11*c^8*d^5 - 90*C*a^5*b^11*c^10*d^3 - 636*C*a^6*b^10*c^3*d^10
- 1004*C*a^6*b^10*c^5*d^8 - 296*C*a^6*b^10*c^7*d^6 + 116*C*a^6*b^10*c^9*d^4
 - 12*C*a^6*b^10*c^11*d^2 + 328*C*a^7*b^9*c^2*d^11 + 1218*C*a^7*b^9*c^4*d^9
 + 1132*C*a^7*b^9*c^6*d^7 + 223*C*a^7*b^9*c^8*d^5 - 14*C*a^7*b^9*c^10*d^3 -
 828*C*a^8*b^8*c^3*d^10 - 1452*C*a^8*b^8*c^5*d^8 - 708*C*a^8*b^8*c^7*d^6 -
32*C*a^8*b^8*c^9*d^4 - 8*C*a^8*b^8*c^11*d^2 + 234*C*a^9*b^7*c^2*d^11 + 951*
C*a^9*b^7*c^4*d^9 + 964*C*a^9*b^7*c^6*d^7 + 266*C*a^9*b^7*c^8*d^5 + 16*C*a^
9*b^7*c^10*d^3 - 344*C*a^10*b^6*c^3*d^10 - 732*C*a^10*b^6*c^5*d^8 - 392*C*a
^10*b^6*c^7*d^6 - 32*C*a^10*b^6*c^9*d^4 + 10*C*a^11*b^5*c^2*d^11 + 225*C*a^
11*b^5*c^4*d^9 + 256*C*a^11*b^5*c^6*d^7 + 58*C*a^11*b^5*c^8*d^5 + 20*C*a^12
*b^4*c^3*d^10 - 76*C*a^12*b^4*c^5*d^8 - 44*C*a^12*b^4*c^7*d^6 - 30*C*a^13*b
^3*c^2*d^11 - 17*C*a^13*b^3*c^4*d^9 + 4*C*a^13*b^3*c^6*d^7 + 16*C*a^14*b^2*
c^3*d^10 + 4*C*a^14*b^2*c^5*d^8 - A*a*b^15*c^12*d + C*a*b^15*c^12*d)/(a^14*
d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8
*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^
10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^
13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a
^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a
^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 1
5*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^1
1*c^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8
 + 90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^
5*b^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*
d^8 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*
a^7*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^
2*d^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 15
2*a^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*
c^2*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2
- 68*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12

*b^2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*a^13*b*c*d^9) + (tan(e + f*x)*(3*B*a^15*b*d^13 - 3*A*b^16*c^12*d + 3*C*b^16*c^12*d - 24*A*a^4*b^12*d^13 - 104*A*a^6*b^10*d^13 - 199*A*a^8*b^8*d^13 - 189*A*a^10*b^6*d^13 - 77*A*a^12*b^4*d^13 - 7*A*a^14*b^2*d^13 + 8*B*a^5*b^11*d^13 + 24*B*a^7*b^9*d^13 + 51*B*a^9*b^7*d^13 + 65*B*a^11*b^5*d^13 + 33*B*a^13*b^3*d^13 + 24*A*b^16*c^4*d^9 + 56*A*b^16*c^6*d^7 + 25*A*b^16*c^8*d^5 - 10*A*b^16*c^10*d^3 - 4*C*a^6*b^10*d^13 + 7*C*a^8*b^8*d^13 + 21*C*a^10*b^6*d^13 + 5*C*a^12*b^4*d^13 - 5*C*a^14*b^2*d^13 - 16*B*b^16*c^5*d^8 - 48*B*b^16*c^7*d^6 - 36*B*b^16*c^9*d^4 - 4*B*b^16*c^11*d^2 + 4*C*b^16*c^6*d^7 + 23*C*b^16*c^8*d^5 + 22*C*b^16*c^10*d^3 - 48*A*a*b^15*c^3*d^10 - 144*A*a*b^15*c^5*d^8 - 104*A*a*b^15*c^7*d^6 + 4*A*a*b^15*c^9*d^4 + 12*A*a*b^15*c^11*d^2 - A*a^2*b^14*c^12*d + 48*A*a^3*b^13*c*d^12 + 7*A*a^4*b^12*c^12*d + 208*A*a^5*b^11*c*d^12 + 5*A*a^6*b^10*c^12*d + 472*A*a^7*b^9*c*d^12 + 572*A*a^9*b^7*c*d^12 + 324*A*a^11*b^5*c*d^12 + 68*A*a^13*b^3*c*d^12 + 4*A*a^15*b*c^3*d^10 + 24*B*a*b^15*c^4*d^9 + 120*B*a*b^15*c^6*d^7 + 147*B*a*b^15*c^8*d^5 + 58*B*a*b^15*c^10*d^3 + 13*B*a^3*b^13*c^12*d + 5*B*a^5*b^11*c^12*d + 64*B*a^6*b^10*c*d^12 - B*a^7*b^9*c^12*d + 100*B*a^8*b^8*c*d^12 - 4*B*a^10*b^6*c*d^12 - 52*B*a^12*b^4*c*d^12 - 12*B*a^14*b^2*c*d^12 + 2*B*a^15*b*c^2*d^11 - B*a^15*b*c^4*d^9 + 24*C*a*b^15*c^5*d^8 + 8*C*a*b^15*c^7*d^6 - 28*C*a*b^15*c^9*d^4 - 12*C*a*b^15*c^11*d^2 + C*a^2*b^14*c^12*d - 7*C*a^4*b^12*c^12*d + 8*C*a^5*b^11*c*d^12 - 5*C*a^6*b^10*c^12*d - 88*C*a^7*b^9*c*d^12 - 236*C*a^9*b^7*c*d^12 - 180*C*a^11*b^5*c*d^12 - 44*C*a^13*b^3*c*d^12 - 4*C*a^15*b*c^3*d^10 + 200*A*a^2*b^14*c^4*d^9 + 468*A*a^2*b^14*c^6*d^7 + 283*A*a^2*b^14*c^8*d^5 + 14*A*a^2*b^14*c^10*d^3 - 192*A*a^3*b^13*c^3*d^10 - 936*A*a^3*b^13*c^5*d^8 - 952*A*a^3*b^13*c^7*d^6 - 268*A*a^3*b^13*c^9*d^4 - 12*A*a^3*b^13*c^11*d^2 - 24*A*a^4*b^12*c^2*d^11 + 790*A*a^4*b^12*c^4*d^9 + 1768*A*a^4*b^12*c^6*d^7 + 1137*A*a^4*b^12*c^8*d^5 + 166*A*a^4*b^12*c^10*d^3 - 200*A*a^5*b^11*c^3*d^10 - 2016*A*a^5*b^11*c^5*d^8 - 2264*A*a^5*b^11*c^7*d^6 - 716*A*a^5*b^11*c^9*d^4 - 60*A*a^5*b^11*c^11*d^2 - 316*A*a^6*b^10*c^2*d^11 + 906*A*a^6*b^10*c^4*d^9 + 2524*A*a^6*b^10*c^6*d^7 + 1651*A*a^6*b^10*c^8*d^5 + 250*A*a^6*b^10*c^10*d^3 + 472*A*a^7*b^9*c^3*d^10 - 1512*A*a^7*b^9*c^5*d^8 - 2088*A*a^7*b^9*c^7*d^6 - 612*A*a^7*b^9*c^9*d^4 - 36*A*a^7*b^9*c^11*d^2 - 838*A*a^8*b^8*c^2*d^11 - 177*A*a^8*b^8*c^4*d^9 + 1252*A*a^8*b^8*c^6*d^7 + 898*A*a^8*b^8*c^8*d^5 + 108*A*a^8*b^8*c^10*d^3 + 1148*A*a^9*b^7*c^3*d^10 + 72*A*a^9*b^7*c^5*d^8 - 672*A*a^9*b^7*c^7*d^6 - 168*A*a^9*b^7*c^9*d^4 - 858*A*a^10*b^6*c^2*d^11 - 795*A*a^10*b^6*c^4*d^9 + 126*A*a^10*b^6*c^8*d^5 + 756*A*a^11*b^5*c^3*d^10 + 432*A*a^11*b^5*c^5*d^8 - 346*A*a^12*b^4*c^2*d^11 - 353*A*a^12*b^4*c^4*d^9 - 84*A*a^12*b^4*c^6*d^7 + 140*A*a^13*b^3*c^3*d^10 + 72*A*a^13*b^3*c^5*d^8 - 34*A*a^14*b^2*c^2*d^11 - 27*A*a^14*b^2*c^4*d^9 + 16*B*a^2*b^14*c^3*d^10 - 128*B*a^2*b^14*c^5*d^8 - 408*B*a^2*b^14*c^7*d^6 - 316*B*a^2*b^14*c^9*d^4 - 52*B*a^2*b^14*c^11*d^2 - 32*B*a^3*b^13*c^2*d^11 + 8*B*a^3*b^13*c^4*d^9 + 460*B*a^3*b^13*c^6*d^7 + 617*B*a^3*b^13*c^8*d^5 + 210*B*a^3*b^13*c^10*d^3 + 240*B*a^4*b^12*c^3*d^10 + 144*B*a^4*b^12*c^5*d^8 - 576*B*a^4*b^12*c^7*d^6 - 564*B*a^4*b^12*c^9*d^4 - 84*B*a^4*b^12*c^11*d^2 - 280*B*a^5*b^11*c^2*d^11 - 814*B*a^5*b^11*c^4*d^9 - 152*B*a^5*b^11*c^6*d^7 + 587*B*a^5*b^11*c^8*

d^5 + 218*B*a^5*b^11*c^10*d^3 + 968*B*a^6*b^10*c^3*d^10 + 1472*B*a^6*b^10*c
^5*d^8 + 328*B*a^6*b^10*c^7*d^6 - 268*B*a^6*b^10*c^9*d^4 - 28*B*a^6*b^10*c^
11*d^2 - 612*B*a^7*b^9*c^2*d^11 - 2034*B*a^7*b^9*c^4*d^9 - 1596*B*a^7*b^9*c
^6*d^7 - 159*B*a^7*b^9*c^8*d^5 + 38*B*a^7*b^9*c^10*d^3 + 1348*B*a^8*b^8*c^3
*d^10 + 2232*B*a^8*b^8*c^5*d^8 + 1048*B*a^8*b^8*c^7*d^6 + 72*B*a^8*b^8*c^9*
d^4 + 8*B*a^8*b^8*c^11*d^2 - 474*B*a^9*b^7*c^2*d^11 - 1731*B*a^9*b^7*c^4*d^
9 - 1524*B*a^9*b^7*c^6*d^7 - 346*B*a^9*b^7*c^8*d^5 - 28*B*a^9*b^7*c^10*d^3
+ 668*B*a^10*b^6*c^3*d^10 + 1176*B*a^10*b^6*c^5*d^8 + 560*B*a^10*b^6*c^7*d^
6 + 56*B*a^10*b^6*c^9*d^4 - 70*B*a^11*b^5*c^2*d^11 - 513*B*a^11*b^5*c^4*d^9
 - 448*B*a^11*b^5*c^6*d^7 - 70*B*a^11*b^5*c^8*d^5 + 60*B*a^12*b^4*c^3*d^10
+ 168*B*a^12*b^4*c^5*d^8 + 56*B*a^12*b^4*c^7*d^6 + 42*B*a^13*b^3*c^2*d^11 -
 19*B*a^13*b^3*c^4*d^9 - 28*B*a^13*b^3*c^6*d^7 - 4*B*a^14*b^2*c^3*d^10 + 8*
B*a^14*b^2*c^5*d^8 - 92*C*a^2*b^14*c^4*d^9 - 204*C*a^2*b^14*c^6*d^7 - 79*C*
a^2*b^14*c^8*d^5 + 34*C*a^2*b^14*c^10*d^3 + 96*C*a^3*b^13*c^3*d^10 + 504*C*
a^3*b^13*c^5*d^8 + 568*C*a^3*b^13*c^7*d^6 + 172*C*a^3*b^13*c^9*d^4 + 12*C*a
^3*b^13*c^11*d^2 - 36*C*a^4*b^12*c^2*d^11 - 646*C*a^4*b^12*c^4*d^9 - 1324*C
*a^4*b^12*c^6*d^7 - 801*C*a^4*b^12*c^8*d^5 - 94*C*a^4*b^12*c^10*d^3 + 344*C
*a^5*b^11*c^3*d^10 + 1512*C*a^5*b^11*c^5*d^8 + 1688*C*a^5*b^11*c^7*d^6 + 57
2*C*a^5*b^11*c^9*d^4 + 60*C*a^5*b^11*c^11*d^2 + 52*C*a^6*b^10*c^2*d^11 - 94
2*C*a^6*b^10*c^4*d^9 - 2188*C*a^6*b^10*c^6*d^7 - 1387*C*a^6*b^10*c^8*d^5 -
202*C*a^6*b^10*c^10*d^3 + 104*C*a^7*b^9*c^3*d^10 + 1416*C*a^7*b^9*c^5*d^8 +
 1704*C*a^7*b^9*c^7*d^6 + 516*C*a^7*b^9*c^9*d^4 + 36*C*a^7*b^9*c^11*d^2 + 3
82*C*a^8*b^8*c^2*d^11 - 87*C*a^8*b^8*c^4*d^9 - 1168*C*a^8*b^8*c^6*d^7 - 802
*C*a^8*b^8*c^8*d^5 - 96*C*a^8*b^8*c^10*d^3 - 524*C*a^9*b^7*c^3*d^10 + 144*C
*a^9*b^7*c^5*d^8 + 576*C*a^9*b^7*c^7*d^6 + 144*C*a^9*b^7*c^9*d^4 + 474*C*a^
10*b^6*c^2*d^11 + 543*C*a^10*b^6*c^4*d^9 - 24*C*a^10*b^6*c^6*d^7 - 114*C*a^
10*b^6*c^8*d^5 - 468*C*a^11*b^5*c^3*d^10 - 288*C*a^11*b^5*c^5*d^8 + 190*C*a
^12*b^4*c^2*d^11 + 257*C*a^12*b^4*c^4*d^9 + 72*C*a^12*b^4*c^6*d^7 - 92*C*a^
13*b^3*c^3*d^10 - 48*C*a^13*b^3*c^5*d^8 + 10*C*a^14*b^2*c^2*d^11 + 15*C*a^1
4*b^2*c^4*d^9 + 4*A*a^15*b*c*d^12 + 7*B*a*b^15*c^12*d - 4*C*a^15*b*c*d^12))
/(a^14*d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^1
0 + a^8*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12
*b^2*d^10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 -
 6*a*b^13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9
 - 36*a^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^
9 - 6*a^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*
d^5 + 15*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*
a^3*b^11*c^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*
c^2*d^8 + 90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2
- 92*a^5*b^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b
^8*c^2*d^8 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2
 - 168*a^7*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8
*b^6*c^2*d^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d
^2 - 152*a^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^
10*b^4*c^2*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c

^8*d^2 - 68*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 +
23*a^12*b^2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*
c^9*d - 6*a^13*b*c*d^9)) + (156*A^2*a^3*b^10*d^11 + 204*A^2*a^5*b^8*d^11 +
85*A^2*a^7*b^6*d^11 + 3*A^2*a^11*b^2*d^11 + 4*B^2*a^3*b^10*d^11 + 28*B^2*a^
5*b^8*d^11 + 45*B^2*a^7*b^6*d^11 + 24*B^2*a^9*b^4*d^11 - B^2*a^11*b^2*d^11
+ 36*A^2*b^13*c^3*d^8 - 4*A^2*b^13*c^5*d^6 - 3*A^2*b^13*c^7*d^4 + C^2*a^7*b
^6*d^11 + 3*C^2*a^11*b^2*d^11 + 16*B^2*b^13*c^3*d^8 + 16*B^2*b^13*c^5*d^6 +
B^2*b^13*c^7*d^4 + 2*B^2*b^13*c^9*d^2 + 8*C^2*b^13*c^5*d^6 + 9*C^2*b^13*c^
7*d^4 + 36*A^2*a*b^12*d^11 + 36*A^2*b^13*c*d^10 + 8*A^2*a^2*b^11*c^3*d^8 +
17*A^2*a^2*b^11*c^5*d^6 + 23*A^2*a^2*b^11*c^7*d^4 - 8*A^2*a^2*b^11*c^9*d^2
+ 168*A^2*a^3*b^10*c^2*d^9 + 87*A^2*a^3*b^10*c^4*d^7 + A^2*a^3*b^10*c^6*d^5
- 417*A^2*a^4*b^9*c^3*d^8 - 205*A^2*a^4*b^9*c^5*d^6 + 23*A^2*a^4*b^9*c^7*d
^4 + 16*A^2*a^4*b^9*c^9*d^2 + 393*A^2*a^5*b^8*c^2*d^9 + 359*A^2*a^5*b^8*c^4
*d^7 + 13*A^2*a^5*b^8*c^6*d^5 - 53*A^2*a^5*b^8*c^8*d^3 - 411*A^2*a^6*b^7*c^
3*d^8 - 13*A^2*a^6*b^7*c^5*d^6 + 93*A^2*a^6*b^7*c^7*d^4 + 43*A^2*a^7*b^6*c^
2*d^9 - 75*A^2*a^7*b^6*c^4*d^7 - 89*A^2*a^7*b^6*c^6*d^5 - 7*A^2*a^8*b^5*c^3
*d^8 + 37*A^2*a^8*b^5*c^5*d^6 + 5*A^2*a^9*b^4*c^2*d^9 + 9*A^2*a^9*b^4*c^4*d
^7 - 17*A^2*a^10*b^3*c^3*d^8 + 7*A^2*a^11*b^2*c^2*d^9 + 36*B^2*a^2*b^11*c^3
*d^8 - 11*B^2*a^2*b^11*c^5*d^6 - 13*B^2*a^2*b^11*c^7*d^4 + 12*B^2*a^2*b^11*
c^9*d^2 + 48*B^2*a^3*b^10*c^2*d^9 - 49*B^2*a^3*b^10*c^4*d^7 - 39*B^2*a^3*b^
10*c^6*d^5 - 20*B^2*a^3*b^10*c^8*d^3 + 163*B^2*a^4*b^9*c^3*d^8 + 91*B^2*a^4
*b^9*c^5*d^6 + 3*B^2*a^4*b^9*c^7*d^4 - 14*B^2*a^4*b^9*c^9*d^2 - 47*B^2*a^5*
b^8*c^2*d^9 - 209*B^2*a^5*b^8*c^4*d^7 + 13*B^2*a^5*b^8*c^6*d^5 + 43*B^2*a^5
*b^8*c^8*d^3 - 31*B^2*a^6*b^7*c^3*d^8 - 185*B^2*a^6*b^7*c^5*d^6 - 79*B^2*a^
6*b^7*c^7*d^4 + 131*B^2*a^7*b^6*c^2*d^9 + 149*B^2*a^7*b^6*c^4*d^7 + 119*B^2
*a^7*b^6*c^6*d^5 - 199*B^2*a^8*b^5*c^3*d^8 - 127*B^2*a^8*b^5*c^5*d^6 + 9*B^
2*a^9*b^4*c^2*d^9 - 19*B^2*a^9*b^4*c^4*d^7 + 7*B^2*a^10*b^3*c^3*d^8 - 5*B^2
*a^11*b^2*c^2*d^9 + 20*C^2*a^2*b^11*c^3*d^8 + 41*C^2*a^2*b^11*c^5*d^6 + 11*
C^2*a^2*b^11*c^7*d^4 - 8*C^2*a^2*b^11*c^9*d^2 + 36*C^2*a^3*b^10*c^2*d^9 + 9
9*C^2*a^3*b^10*c^4*d^7 - 11*C^2*a^3*b^10*c^6*d^5 - 69*C^2*a^4*b^9*c^3*d^8 -
97*C^2*a^4*b^9*c^5*d^6 - 37*C^2*a^4*b^9*c^7*d^4 + 16*C^2*a^4*b^9*c^9*d^2 +
141*C^2*a^5*b^8*c^2*d^9 + 179*C^2*a^5*b^8*c^4*d^7 - 119*C^2*a^5*b^8*c^6*d^
5 - 53*C^2*a^5*b^8*c^8*d^3 + 57*C^2*a^6*b^7*c^3*d^8 + 143*C^2*a^6*b^7*c^5*d
^6 + 57*C^2*a^6*b^7*c^7*d^4 - 65*C^2*a^7*b^6*c^2*d^9 - 231*C^2*a^7*b^6*c^4*
d^7 - 221*C^2*a^7*b^6*c^6*d^5 + 113*C^2*a^8*b^5*c^3*d^8 + 61*C^2*a^8*b^5*c^
5*d^6 + 17*C^2*a^9*b^4*c^2*d^9 - 15*C^2*a^9*b^4*c^4*d^7 - 36*C^2*a^9*b^4*c^
6*d^5 - 65*C^2*a^10*b^3*c^3*d^8 - 36*C^2*a^10*b^3*c^5*d^6 + 7*C^2*a^11*b^2*
c^2*d^9 - 24*A*B*a^2*b^11*d^11 - 136*A*B*a^4*b^9*d^11 - 200*A*B*a^6*b^7*d^1
1 - 89*A*B*a^8*b^5*d^11 + 6*A*B*a^10*b^3*d^11 - 12*A*C*a^3*b^10*d^11 + 12*A
*C*a^5*b^8*d^11 + 58*A*C*a^7*b^6*d^11 + 36*A*C*a^9*b^4*d^11 - 6*A*C*a^11*b^
2*d^11 - 48*A*B*b^13*c^2*d^9 - 48*A*B*b^13*c^4*d^7 - A*B*b^13*c^8*d^3 + 4*B
*C*a^4*b^9*d^11 - 4*B*C*a^6*b^7*d^11 - 19*B*C*a^8*b^5*d^11 - 18*B*C*a^10*b^
3*d^11 + 36*A*C*b^13*c^3*d^8 + 32*A*C*b^13*c^5*d^6 - 6*A*C*b^13*c^7*d^4 - 2
4*B*C*b^13*c^4*d^7 - 24*B*C*b^13*c^6*d^5 + B*C*b^13*c^8*d^3 + 2*A^2*a*b^12*
c^10*d - A^2*a^12*b*c*d^10 - 2*B^2*a*b^12*c^10*d + B^2*a^12*b*c*d^10 + 2*C^

2*a*b^12*c^10*d - C^2*a^12*b*c*d^10 - 44*A^2*a*b^12*c^4*d^7 - 29*A^2*a*b^12 *c^6*d^5 + A^2*a*b^12*c^8*d^3 + 24*A^2*a^2*b^11*c*d^10 - 2*A^2*a^3*b^10*c^1 0*d - 188*A^2*a^4*b^9*c*d^10 - 277*A^2*a^6*b^7*c*d^10 - 27*A^2*a^8*b^5*c*d^ 10 - 15*A^2*a^10*b^3*c*d^10 + 32*B^2*a*b^12*c^2*d^9 + 16*B^2*a*b^12*c^4*d^7 - 5*B^2*a*b^12*c^6*d^5 - 11*B^2*a*b^12*c^8*d^3 + 20*B^2*a^2*b^11*c*d^10 + 2*B^2*a^3*b^10*c^10*d + 72*B^2*a^4*b^9*c*d^10 + 47*B^2*a^6*b^7*c*d^10 - 89* B^2*a^8*b^5*c*d^10 + 5*B^2*a^10*b^3*c*d^10 + 16*C^2*a*b^12*c^4*d^7 - 5*C^2* a*b^12*c^6*d^5 + C^2*a*b^12*c^8*d^3 - 2*C^2*a^3*b^10*c^10*d - 8*C^2*a^4*b^9 *c*d^10 - C^2*a^6*b^7*c*d^10 + 69*C^2*a^8*b^5*c*d^10 - 27*C^2*a^10*b^3*c*d^ 10 - A*B*a^12*b*d^11 + A*B*b^13*c^10*d + B*C*a^12*b*d^11 - B*C*b^13*c^10*d - 72*A*B*a*b^12*c*d^10 - 4*A*C*a*b^12*c^10*d + 2*A*C*a^12*b*c*d^10 - 24*A*B *a*b^12*c^3*d^8 + 40*A*B*a*b^12*c^5*d^6 + 32*A*B*a*b^12*c^7*d^4 - 6*A*B*a^2 *b^11*c^10*d - 160*A*B*a^3*b^10*c*d^10 + A*B*a^4*b^9*c^10*d + 56*A*B*a^5*b^ 8*c*d^10 + 312*A*B*a^7*b^6*c*d^10 - 8*A*B*a^9*b^4*c*d^10 + A*B*a^12*b*c^2*d ^9 + 36*A*C*a*b^12*c^2*d^9 - 8*A*C*a*b^12*c^4*d^7 - 2*A*C*a*b^12*c^6*d^5 - 2*A*C*a*b^12*c^8*d^3 + 84*A*C*a^2*b^11*c*d^10 + 4*A*C*a^3*b^10*c^10*d + 268 *A*C*a^4*b^9*c*d^10 + 206*A*C*a^6*b^7*c*d^10 - 150*A*C*a^8*b^5*c*d^10 + 6*A *C*a^10*b^3*c*d^10 - 36*B*C*a*b^12*c^3*d^8 + 8*B*C*a*b^12*c^5*d^6 + 4*B*C*a *b^12*c^7*d^4 + 6*B*C*a^2*b^11*c^10*d - 20*B*C*a^3*b^10*c*d^10 - B*C*a^4*b^ 9*c^10*d - 116*B*C*a^5*b^8*c*d^10 - 180*B*C*a^7*b^6*c*d^10 + 92*B*C*a^9*b^4 *c*d^10 - B*C*a^12*b*c^2*d^9 - 64*A*B*a^2*b^11*c^2*d^9 + 40*A*B*a^2*b^11*c^ 4*d^7 + 52*A*B*a^2*b^11*c^6*d^5 - 30*A*B*a^2*b^11*c^8*d^3 - 112*A*B*a^3*b^1 0*c^3*d^8 - 104*A*B*a^3*b^10*c^5*d^6 + 40*A*B*a^3*b^10*c^7*d^4 + 40*A*B*a^3 *b^10*c^9*d^2 - 112*A*B*a^4*b^9*c^2*d^9 + 114*A*B*a^4*b^9*c^4*d^7 - 50*A*B* a^4*b^9*c^6*d^5 - 105*A*B*a^4*b^9*c^8*d^3 + 480*A*B*a^5*b^8*c^3*d^8 + 368*A *B*a^5*b^8*c^5*d^6 + 144*A*B*a^5*b^8*c^7*d^4 - 8*A*B*a^5*b^8*c^9*d^2 - 508* A*B*a^6*b^7*c^2*d^9 - 456*A*B*a^6*b^7*c^4*d^7 - 176*A*B*a^6*b^7*c^6*d^5 + 2 8*A*B*a^6*b^7*c^8*d^3 + 584*A*B*a^7*b^6*c^3*d^8 + 104*A*B*a^7*b^6*c^5*d^6 - 56*A*B*a^7*b^6*c^7*d^4 - 23*A*B*a^8*b^5*c^2*d^9 + 170*A*B*a^8*b^5*c^4*d^7 + 70*A*B*a^8*b^5*c^6*d^5 - 56*A*B*a^9*b^4*c^3*d^8 - 56*A*B*a^9*b^4*c^5*d^6 + 30*A*B*a^10*b^3*c^2*d^9 + 28*A*B*a^10*b^3*c^4*d^7 - 8*A*B*a^11*b^2*c^3*d^ 8 + 188*A*C*a^2*b^11*c^3*d^8 + 50*A*C*a^2*b^11*c^5*d^6 - 34*A*C*a^2*b^11*c^ 7*d^4 + 16*A*C*a^2*b^11*c^9*d^2 - 60*A*C*a^3*b^10*c^2*d^9 - 330*A*C*a^3*b^1 0*c^4*d^7 - 134*A*C*a^3*b^10*c^6*d^5 + 630*A*C*a^4*b^9*c^3*d^8 + 374*A*C*a^ 4*b^9*c^5*d^6 + 14*A*C*a^4*b^9*c^7*d^4 - 32*A*C*a^4*b^9*c^9*d^2 - 318*A*C*a ^5*b^8*c^2*d^9 - 754*A*C*a^5*b^8*c^4*d^7 - 110*A*C*a^5*b^8*c^6*d^5 + 106*A* C*a^5*b^8*c^8*d^3 + 210*A*C*a^6*b^7*c^3*d^8 - 202*A*C*a^6*b^7*c^5*d^6 - 150 *A*C*a^6*b^7*c^7*d^4 + 166*A*C*a^7*b^6*c^2*d^9 + 162*A*C*a^7*b^6*c^4*d^7 + 166*A*C*a^7*b^6*c^6*d^5 - 322*A*C*a^8*b^5*c^3*d^8 - 206*A*C*a^8*b^5*c^5*d^6 + 14*A*C*a^9*b^4*c^2*d^9 - 30*A*C*a^9*b^4*c^4*d^7 + 10*A*C*a^10*b^3*c^3*d^ 8 - 14*A*C*a^11*b^2*c^2*d^9 - 68*B*C*a^2*b^11*c^2*d^9 - 160*B*C*a^2*b^11*c^ 4*d^7 - 64*B*C*a^2*b^11*c^6*d^5 + 30*B*C*a^2*b^11*c^8*d^3 + 4*B*C*a^3*b^10* c^3*d^8 + 236*B*C*a^3*b^10*c^5*d^6 + 20*B*C*a^3*b^10*c^7*d^4 - 40*B*C*a^3*b ^10*c^9*d^2 - 140*B*C*a^4*b^9*c^2*d^9 - 174*B*C*a^4*b^9*c^4*d^7 + 110*B*C*a ^4*b^9*c^6*d^5 + 105*B*C*a^4*b^9*c^8*d^3 - 300*B*C*a^5*b^8*c^3*d^8 - 116*B*

C*a^5*b^8*c^5*d^6 - 132*B*C*a^5*b^8*c^7*d^4 + 8*B*C*a^5*b^8*c^9*d^2 + 208*B
*C*a^6*b^7*c^2*d^9 + 420*B*C*a^6*b^7*c^4*d^7 + 236*B*C*a^6*b^7*c^6*d^5 - 28
*B*C*a^6*b^7*c^8*d^3 - 140*B*C*a^7*b^6*c^3*d^8 + 196*B*C*a^7*b^6*c^5*d^6 +
44*B*C*a^7*b^6*c^7*d^4 - 109*B*C*a^8*b^5*c^2*d^9 - 182*B*C*a^8*b^5*c^4*d^7
- 58*B*C*a^8*b^5*c^6*d^5 + 272*B*C*a^9*b^4*c^3*d^8 + 188*B*C*a^9*b^4*c^5*d^
6 - 30*B*C*a^10*b^3*c^2*d^9 - 16*B*C*a^10*b^3*c^4*d^7 + 8*B*C*a^11*b^2*c^3*
d^8)/(a^14*d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8
*c^10 + a^8*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*
a^12*b^2*d^10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d
^2 - 6*a*b^13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c
*d^9 - 36*a^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*
c*d^9 - 6*a^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*
c^5*d^5 + 15*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 -
 20*a^3*b^11*c^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b
^10*c^2*d^8 + 90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*
d^2 - 92*a^5*b^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a
^6*b^8*c^2*d^8 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8
*d^2 - 168*a^7*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98
*a^8*b^6*c^2*d^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c
^8*d^2 - 152*a^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 7
2*a^10*b^4*c^2*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b
^4*c^8*d^2 - 68*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^
3 + 23*a^12*b^2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b
^13*c^9*d - 6*a^13*b*c*d^9) - (tan(e + f*x)*(20*A^2*a^6*b^7*d^11 - 54*A^2*a
^2*b^11*d^11 - 18*A^2*a^4*b^9*d^11 - 18*A^2*b^13*d^11 - 65*A^2*a^8*b^5*d^11
 - 2*B^2*a^2*b^11*d^11 - 6*B^2*a^4*b^9*d^11 + 12*B^2*a^6*b^7*d^11 + 66*B^2*
a^8*b^5*d^11 - 18*B^2*a^10*b^3*d^11 - 6*A^2*b^13*c^2*d^9 + 10*A^2*b^13*c^4*
d^7 + 12*A^2*b^13*c^6*d^5 - 3*A^2*b^13*c^8*d^3 + 2*C^2*a^6*b^7*d^11 - 29*C^
2*a^8*b^5*d^11 + 36*C^2*a^10*b^3*d^11 - 8*B^2*b^13*c^2*d^9 - 8*B^2*b^13*c^4
*d^7 - 18*B^2*b^13*c^6*d^5 - 2*B^2*b^13*c^8*d^3 - 2*C^2*b^13*c^4*d^7 + 6*C^
2*b^13*c^6*d^5 - 9*C^2*b^13*c^8*d^3 - A^2*a^12*b*d^11 - C^2*a^12*b*d^11 - B
^2*b^13*c^10*d - 158*A^2*a^2*b^11*c^2*d^9 - 232*A^2*a^2*b^11*c^4*d^7 - 96*A
^2*a^2*b^11*c^6*d^5 - 34*A^2*a^2*b^11*c^8*d^3 + 504*A^2*a^3*b^10*c^3*d^8 +
248*A^2*a^3*b^10*c^5*d^6 + 120*A^2*a^3*b^10*c^7*d^4 + 28*A^2*a^3*b^10*c^9*d
^2 - 224*A^2*a^4*b^9*c^2*d^9 - 446*A^2*a^4*b^9*c^4*d^7 - 244*A^2*a^4*b^9*c^
6*d^5 - 83*A^2*a^4*b^9*c^8*d^3 + 580*A^2*a^5*b^8*c^3*d^8 + 332*A^2*a^5*b^8*
c^5*d^6 + 132*A^2*a^5*b^8*c^7*d^4 - 252*A^2*a^6*b^7*c^2*d^9 - 452*A^2*a^6*b
^7*c^4*d^7 - 144*A^2*a^6*b^7*c^6*d^5 + 464*A^2*a^7*b^6*c^3*d^8 + 152*A^2*a^
7*b^6*c^5*d^6 - 194*A^2*a^8*b^5*c^2*d^9 - 128*A^2*a^8*b^5*c^4*d^7 + 28*A^2*
a^9*b^4*c^3*d^8 - 2*A^2*a^10*b^3*c^2*d^9 + 18*B^2*a^2*b^11*c^2*d^9 + 4*B^2*
a^2*b^11*c^4*d^7 - 84*B^2*a^2*b^11*c^6*d^5 - 4*B^2*a^2*b^11*c^8*d^3 + 128*B
^2*a^3*b^10*c^3*d^8 + 208*B^2*a^3*b^10*c^5*d^6 + 40*B^2*a^3*b^10*c^7*d^4 -
12*B^2*a^3*b^10*c^9*d^2 + 36*B^2*a^4*b^9*c^2*d^9 - 36*B^2*a^4*b^9*c^4*d^7 -
 134*B^2*a^4*b^9*c^6*d^5 + 22*B^2*a^4*b^9*c^8*d^3 + 180*B^2*a^5*b^8*c^3*d^8
 + 148*B^2*a^5*b^8*c^5*d^6 + 20*B^2*a^5*b^8*c^7*d^4 + 8*B^2*a^5*b^8*c^9*d^2

+ 208*B^2*a^6*b^7*c^2*d^9 + 164*B^2*a^6*b^7*c^4*d^7 - 96*B^2*a^6*b^7*c^6*d^5 - 28*B^2*a^6*b^7*c^8*d^3 - 96*B^2*a^7*b^6*c^3*d^8 + 16*B^2*a^7*b^6*c^5*d^6 + 48*B^2*a^7*b^6*c^7*d^4 + 179*B^2*a^8*b^5*c^2*d^9 + 76*B^2*a^8*b^5*c^4*d^7 - 36*B^2*a^8*b^5*c^6*d^5 + 36*B^2*a^9*b^4*c^3*d^8 - 32*B^2*a^10*b^3*c^2*d^9 - 16*B^2*a^10*b^3*c^4*d^7 + 8*B^2*a^11*b^2*c^3*d^8 - 8*C^2*a^2*b^11*c^2*d^9 + 44*C^2*a^2*b^11*c^4*d^7 + 90*C^2*a^2*b^11*c^6*d^5 - 28*C^2*a^2*b^11*c^8*d^3 - 4*C^2*a^3*b^10*c^5*d^6 + 36*C^2*a^3*b^10*c^7*d^4 + 28*C^2*a^3*b^10*c^9*d^2 + 16*C^2*a^4*b^9*c^2*d^9 + 178*C^2*a^4*b^9*c^4*d^7 + 188*C^2*a^4*b^9*c^6*d^5 - 53*C^2*a^4*b^9*c^8*d^3 + 64*C^2*a^5*b^8*c^3*d^8 + 80*C^2*a^5*b^8*c^5*d^6 - 132*C^2*a^6*b^7*c^2*d^9 - 68*C^2*a^6*b^7*c^4*d^7 + 120*C^2*a^6*b^7*c^6*d^5 + 18*C^2*a^6*b^7*c^8*d^3 + 356*C^2*a^7*b^6*c^3*d^8 + 164*C^2*a^7*b^6*c^5*d^6 - 60*C^2*a^7*b^6*c^7*d^4 - 104*C^2*a^8*b^5*c^2*d^9 - 68*C^2*a^8*b^5*c^4*d^7 + 6*C^2*a^8*b^5*c^6*d^5 + 64*C^2*a^9*b^4*c^3*d^8 + 72*C^2*a^9*b^4*c^5*d^6 + 64*C^2*a^10*b^3*c^2*d^9 + 12*C^2*a^10*b^3*c^4*d^7 - 18*C^2*a^10*b^3*c^6*d^5 - 12*C^2*a^11*b^2*c^3*d^8 + 36*A*B*a^3*b^10*d^11 - 36*A*B*a^5*b^8*d^11 - 132*A*B*a^7*b^6*d^11 + 60*A*B*a^9*b^4*d^11 - 4*A*B*a^11*b^2*d^11 - 18*A*C*a^4*b^9*d^11 + 14*A*C*a^6*b^7*d^11 + 148*A*C*a^8*b^5*d^11 - 18*A*C*a^10*b^3*d^11 + 16*A*B*b^13*c^3*d^8 + 16*A*B*b^13*c^5*d^6 - 8*A*B*b^13*c^7*d^4 + 2*A*B*b^13*c^9*d^2 + 6*B*C*a^5*b^8*d^11 + 18*B*C*a^7*b^6*d^11 - 114*B*C*a^9*b^4*d^11 + 10*B*C*a^11*b^2*d^11 - 12*A*C*b^13*c^2*d^9 + 10*A*C*b^13*c^4*d^7 + 12*A*C*b^13*c^8*d^3 + 8*B*C*b^13*c^3*d^8 - 4*B*C*b^13*c^5*d^6 + 20*B*C*b^13*c^7*d^4 - 2*B*C*b^13*c^9*d^2 + 96*A^2*a*b^12*c*d^10 - 8*B^2*a*b^12*c*d^10 + 136*A^2*a*b^12*c^3*d^8 + 52*A^2*a*b^12*c^5*d^6 + 20*A^2*a*b^12*c^7*d^4 + 4*A^2*a*b^12*c^9*d^2 - 4*A^2*a^2*b^11*c^10*d + 336*A^2*a^3*b^10*c*d^10 + 372*A^2*a^5*b^8*c*d^10 + 320*A^2*a^7*b^6*c*d^10 + 40*A^2*a^9*b^4*c*d^10 + 4*A^2*a^11*b^2*c*d^10 + 48*B^2*a*b^12*c^3*d^8 + 92*B^2*a*b^12*c^5*d^6 + 36*B^2*a*b^12*c^7*d^4 + 4*B^2*a*b^12*c^9*d^2 + 2*B^2*a^2*b^11*c^10*d - 16*B^2*a^3*b^10*c*d^10 - B^2*a^4*b^9*c^10*d + 52*B^2*a^5*b^8*c*d^10 - 72*B^2*a^7*b^6*c*d^10 + 24*B^2*a^9*b^4*c*d^10 + 4*B^2*a^11*b^2*c*d^10 - B^2*a^12*b*c^2*d^9 - 8*C^2*a*b^12*c^3*d^8 - 8*C^2*a*b^12*c^5*d^6 + 8*C^2*a*b^12*c^7*d^4 + 4*C^2*a*b^12*c^9*d^2 - 4*C^2*a^2*b^11*c^10*d - 24*C^2*a^5*b^8*c*d^10 + 140*C^2*a^7*b^6*c*d^10 + 4*C^2*a^9*b^4*c*d^10 - 8*C^2*a^11*b^2*c*d^10 + 12*A*B*a*b^12*d^11 + 2*A*C*a^12*b*d^11 + 24*A*B*b^13*c*d^10 - 4*A*B*a*b^12*c^10*d + 2*A*B*a^12*b*c*d^10 - 24*A*C*a*b^12*c*d^10 + 4*B*C*a*b^12*c^10*d - 2*B*C*a^12*b*c*d^10 - 140*A*B*a*b^12*c^2*d^9 - 220*A*B*a*b^12*c^4*d^7 - 68*A*B*a*b^12*c^6*d^5 - 12*A*B*a*b^12*c^8*d^3 + 16*A*B*a^2*b^11*c*d^10 + 4*A*B*a^3*b^10*c^10*d - 136*A*B*a^4*b^9*c*d^10 + 8*A*B*a^6*b^7*c*d^10 - 174*A*B*a^8*b^5*c*d^10 - 4*A*B*a^10*b^3*c*d^10 + 16*A*C*a*b^12*c^3*d^8 + 28*A*C*a*b^12*c^5*d^6 - 28*A*C*a*b^12*c^7*d^4 - 8*A*C*a*b^12*c^9*d^2 + 8*A*C*a^2*b^11*c^10*d - 48*A*C*a^3*b^10*c*d^10 + 84*A*C*a^5*b^8*c*d^10 - 172*A*C*a^7*b^6*c*d^10 + 28*A*C*a^9*b^4*c*d^10 + 4*A*C*a^11*b^2*c*d^10 + 20*B*C*a*b^12*c^2*d^9 - 14*B*C*a*b^12*c^4*d^7 - 52*B*C*a*b^12*c^6*d^5 - 6*B*C*a*b^12*c^8*d^3 + 8*B*C*a^2*b^11*c*d^10 - 4*B*C*a^3*b^10*c^10*d + 28*B*C*a^4*b^9*c*d^10 - 188*B*C*a^6*b^7*c*d^10 + 114*B*C*a^8*b^5*c*d^10 + 16*B*C*a^10*b^3*c*d^10 + 64*A*B*a^2*b^11*c^3*d^8 + 184*A*B*a^2*b^11*c^5*d^6 + 32*A*B*a^2*b^

11*c^7*d^4 + 20*A*B*a^2*b^11*c^9*d^2 - 300*A*B*a^3*b^10*c^2*d^9 - 420*A*B*a
^3*b^10*c^4*d^7 - 84*A*B*a^3*b^10*c^6*d^5 - 20*A*B*a^3*b^10*c^8*d^3 + 8*A*B
*a^4*b^9*c^3*d^8 + 292*A*B*a^4*b^9*c^5*d^6 - 40*A*B*a^4*b^9*c^7*d^4 - 30*A*
B*a^4*b^9*c^9*d^2 - 580*A*B*a^5*b^8*c^2*d^9 - 596*A*B*a^5*b^8*c^4*d^7 + 60*
A*B*a^5*b^8*c^6*d^5 + 96*A*B*a^5*b^8*c^8*d^3 + 208*A*B*a^6*b^7*c^3*d^8 + 12
8*A*B*a^6*b^7*c^5*d^6 - 144*A*B*a^6*b^7*c^7*d^4 - 340*A*B*a^7*b^6*c^2*d^9 -
100*A*B*a^7*b^6*c^4*d^7 + 92*A*B*a^7*b^6*c^6*d^5 - 200*A*B*a^8*b^5*c^3*d^8
- 28*A*B*a^8*b^5*c^5*d^6 + 92*A*B*a^9*b^4*c^2*d^9 + 56*A*B*a^9*b^4*c^4*d^7
- 12*A*B*a^11*b^2*c^2*d^9 + 112*A*C*a^2*b^11*c^2*d^9 + 242*A*C*a^2*b^11*c^
4*d^7 + 60*A*C*a^2*b^11*c^6*d^5 + 62*A*C*a^2*b^11*c^8*d^3 + 72*A*C*a^3*b^10
*c^3*d^8 + 44*A*C*a^3*b^10*c^5*d^6 - 156*A*C*a^3*b^10*c^7*d^4 - 56*A*C*a^3*
b^10*c^9*d^2 + 172*A*C*a^4*b^9*c^2*d^9 + 304*A*C*a^4*b^9*c^4*d^7 + 92*A*C*a
^4*b^9*c^6*d^5 + 136*A*C*a^4*b^9*c^8*d^3 + 220*A*C*a^5*b^8*c^3*d^8 + 20*A*C
*a^5*b^8*c^5*d^6 - 132*A*C*a^5*b^8*c^7*d^4 + 420*A*C*a^6*b^7*c^2*d^9 + 484*
A*C*a^6*b^7*c^4*d^7 - 12*A*C*a^6*b^7*c^6*d^5 - 18*A*C*a^6*b^7*c^8*d^3 - 244
*A*C*a^7*b^6*c^3*d^8 - 28*A*C*a^7*b^6*c^5*d^6 + 60*A*C*a^7*b^6*c^7*d^4 + 35
2*A*C*a^8*b^5*c^2*d^9 + 142*A*C*a^8*b^5*c^4*d^7 - 60*A*C*a^8*b^5*c^6*d^5 +
52*A*C*a^9*b^4*c^3*d^8 - 44*A*C*a^10*b^3*c^2*d^9 - 30*A*C*a^10*b^3*c^4*d^7
+ 12*A*C*a^11*b^2*c^3*d^8 - 88*B*C*a^2*b^11*c^3*d^8 - 172*B*C*a^2*b^11*c^5*
d^6 + 28*B*C*a^2*b^11*c^7*d^4 - 20*B*C*a^2*b^11*c^9*d^2 - 66*B*C*a^3*b^10*c
^4*d^7 - 96*B*C*a^3*b^10*c^6*d^5 - 10*B*C*a^3*b^10*c^8*d^3 - 332*B*C*a^4*b^
9*c^3*d^8 - 448*B*C*a^4*b^9*c^5*d^6 + 100*B*C*a^4*b^9*c^7*d^4 + 30*B*C*a^4*
b^9*c^9*d^2 + 160*B*C*a^5*b^8*c^2*d^9 + 248*B*C*a^5*b^8*c^4*d^7 - 24*B*C*a^
5*b^8*c^6*d^5 - 102*B*C*a^5*b^8*c^8*d^3 - 652*B*C*a^6*b^7*c^3*d^8 - 404*B*C
*a^6*b^7*c^5*d^6 + 132*B*C*a^6*b^7*c^7*d^4 - 80*B*C*a^7*b^6*c^2*d^9 - 80*B*
C*a^7*b^6*c^4*d^7 + 40*B*C*a^7*b^6*c^6*d^5 + 6*B*C*a^7*b^6*c^8*d^3 + 68*B*C
*a^8*b^5*c^3*d^8 - 68*B*C*a^8*b^5*c^5*d^6 - 24*B*C*a^8*b^5*c^7*d^4 - 272*B*
C*a^9*b^4*c^2*d^9 - 146*B*C*a^9*b^4*c^4*d^7 + 36*B*C*a^9*b^4*c^6*d^5 + 36*B
*C*a^10*b^3*c^3*d^8 + 24*B*C*a^10*b^3*c^5*d^6 + 12*B*C*a^11*b^2*c^2*d^9 - 6
*B*C*a^11*b^2*c^4*d^7))/(a^14*d^10 + b^14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^
10*c^10 + 4*a^6*b^8*c^10 + a^8*b^6*c^10 + a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6
*a^10*b^4*d^10 + 4*a^12*b^2*d^10 + 2*a^14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6
*d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*c^5*d^5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*
c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*b^9*c^9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*
c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*b^5*c^9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*
c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a^2*b^12*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23
*a^2*b^12*c^8*d^2 - 20*a^3*b^11*c^3*d^7 - 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11
*c^7*d^3 + 15*a^4*b^10*c^2*d^8 + 90*a^4*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4
+ 72*a^4*b^10*c^8*d^2 - 92*a^5*b^9*c^3*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5
*b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8 + 211*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d
^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7*b^7*c^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a
^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d^8 + 244*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6
*d^4 + 62*a^8*b^6*c^8*d^2 - 152*a^9*b^5*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*
a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2*d^8 + 141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*
c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 68*a^11*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 -

20*a^11*b^3*c^7*d^3 + 23*a^12*b^2*c^2*d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*
b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*a^13*b*c*d^9)) + (tan(e + f*x)*(10*A^3*a^6
*b^4*d^9 - 27*A^3*a^2*b^8*d^9 - 24*A^3*a^4*b^6*d^9 - 9*A^3*b^10*d^9 + B^3*a
^3*b^7*d^9 + B^3*a^5*b^5*d^9 - 12*A^3*b^10*c^2*d^7 - A^3*b^10*c^4*d^5 - C^3
*a^6*b^4*d^9 + 3*C^3*a^8*b^2*d^9 + 4*B^3*b^10*c^5*d^4 + C^3*b^10*c^4*d^5 +
9*A^2*C*b^10*d^9 - 58*A^3*a^2*b^8*c^2*d^7 - 17*A^3*a^2*b^8*c^4*d^5 + 52*A^3
*a^3*b^7*c^3*d^6 - 46*A^3*a^4*b^6*c^2*d^7 - 8*B^3*a^2*b^8*c^3*d^6 - 8*B^3*a
^2*b^8*c^5*d^4 + 16*B^3*a^3*b^7*c^2*d^7 + 17*B^3*a^3*b^7*c^4*d^5 + 20*B^3*a
^4*b^6*c^3*d^6 + 4*B^3*a^4*b^6*c^5*d^4 - 26*B^3*a^5*b^5*c^2*d^7 - 17*B^3*a^
5*b^5*c^4*d^5 + 28*B^3*a^6*b^4*c^3*d^6 - 6*B^3*a^7*b^3*c^2*d^7 + 4*C^3*a^2*
b^8*c^2*d^7 - 10*C^3*a^2*b^8*c^4*d^5 - 12*C^3*a^2*b^8*c^6*d^3 + 20*C^3*a^3*
b^7*c^3*d^6 + 36*C^3*a^3*b^7*c^5*d^4 - 2*C^3*a^4*b^6*c^2*d^7 - 6*C^3*a^4*b^
6*c^4*d^5 + 6*C^3*a^6*b^4*c^2*d^7 + 9*C^3*a^6*b^4*c^4*d^5 + 15*A^2*B*a*b^9*
d^9 + 12*A^2*B*b^10*c*d^8 + 12*A^3*a*b^9*c*d^8 - 7*A*B^2*a^2*b^8*d^9 - 15*A
*B^2*a^4*b^6*d^9 - 24*A*B^2*a^6*b^4*d^9 + 45*A^2*B*a^3*b^7*d^9 + 56*A^2*B*a
^5*b^5*d^9 - 6*A^2*B*a^7*b^3*d^9 + 3*A*C^2*a^4*b^6*d^9 + 21*A*C^2*a^6*b^4*d
^9 - 6*A*C^2*a^8*b^2*d^9 + 27*A^2*C*a^2*b^8*d^9 + 21*A^2*C*a^4*b^6*d^9 - 30
*A^2*C*a^6*b^4*d^9 + 3*A^2*C*a^8*b^2*d^9 - 4*A*B^2*b^10*c^2*d^7 - 14*A*B^2*
b^10*c^4*d^5 - B*C^2*a^5*b^5*d^9 - 9*B*C^2*a^7*b^3*d^9 + 20*A^2*B*b^10*c^3*
d^6 + B^2*C*a^2*b^8*d^9 + 3*B^2*C*a^4*b^6*d^9 + 6*B^2*C*a^6*b^4*d^9 + 6*A*C
^2*b^10*c^2*d^7 + 6*A*C^2*b^10*c^4*d^5 + 6*A^2*C*b^10*c^2*d^7 - 6*A^2*C*b^1
0*c^4*d^5 - 4*B*C^2*b^10*c^3*d^6 - 6*B*C^2*b^10*c^5*d^4 + 4*B^2*C*b^10*c^2*
d^7 + 8*B^2*C*b^10*c^4*d^5 - 3*B^2*C*b^10*c^6*d^3 + 20*A^3*a*b^9*c^3*d^6 +
36*A^3*a^3*b^7*c*d^8 - 8*A^3*a^5*b^5*c*d^8 + 4*B^3*a*b^9*c^2*d^7 + 2*B^3*a*
b^9*c^4*d^5 + 4*B^3*a^2*b^8*c*d^8 + 12*B^3*a^4*b^6*c*d^8 + 24*B^3*a^6*b^4*c
*d^8 + 4*C^3*a*b^9*c^3*d^6 + 12*C^3*a*b^9*c^5*d^4 + 8*C^3*a^5*b^5*c*d^8 - 6
*A*B*C*a*b^9*d^9 - 12*A*B*C*b^10*c*d^8 + 8*A*B^2*a^2*b^8*c^2*d^7 - 7*A*B^2*
a^2*b^8*c^4*d^5 - 92*A*B^2*a^3*b^7*c^3*d^6 - 16*A*B^2*a^3*b^7*c^5*d^4 + 54*
A*B^2*a^4*b^6*c^2*d^7 + 55*A*B^2*a^4*b^6*c^4*d^5 - 56*A*B^2*a^5*b^5*c^3*d^6
 - 22*A*B^2*a^6*b^4*c^2*d^7 + 68*A^2*B*a^2*b^8*c^3*d^6 + 16*A^2*B*a^2*b^8*c
^5*d^4 + 46*A^2*B*a^3*b^7*c^2*d^7 - 33*A^2*B*a^3*b^7*c^4*d^5 - 16*A^2*B*a^4
*b^6*c^3*d^6 + 82*A^2*B*a^5*b^5*c^2*d^7 - 12*A*C^2*a^2*b^8*c^2*d^7 + 30*A*C
^2*a^2*b^8*c^4*d^5 + 24*A*C^2*a^2*b^8*c^6*d^3 + 12*A*C^2*a^3*b^7*c^3*d^6 -
72*A*C^2*a^3*b^7*c^5*d^4 + 12*A*C^2*a^4*b^6*c^2*d^7 + 39*A*C^2*a^4*b^6*c^4*
d^5 + 6*A*C^2*a^6*b^4*c^2*d^7 - 9*A*C^2*a^6*b^4*c^4*d^5 + 66*A^2*C*a^2*b^8*
c^2*d^7 - 3*A^2*C*a^2*b^8*c^4*d^5 - 12*A^2*C*a^2*b^8*c^6*d^3 - 84*A^2*C*a^3
*b^7*c^3*d^6 + 36*A^2*C*a^3*b^7*c^5*d^4 + 36*A^2*C*a^4*b^6*c^2*d^7 - 33*A^2
*C*a^4*b^6*c^4*d^5 - 12*A^2*C*a^6*b^4*c^2*d^7 + 8*B*C^2*a^2*b^8*c^3*d^6 + 4
*B*C^2*a^2*b^8*c^5*d^4 - 20*B*C^2*a^3*b^7*c^2*d^7 - 66*B*C^2*a^3*b^7*c^4*d^
5 - 12*B*C^2*a^3*b^7*c^6*d^3 + 32*B*C^2*a^4*b^6*c^3*d^6 + 42*B*C^2*a^4*b^6*
c^5*d^4 + 4*B*C^2*a^5*b^5*c^2*d^7 - 21*B*C^2*a^5*b^5*c^4*d^5 - 12*B*C^2*a^6
*b^4*c^3*d^6 + 6*B*C^2*a^7*b^3*c^2*d^7 + 9*B*C^2*a^7*b^3*c^4*d^5 - 2*B^2*C*
a^2*b^8*c^2*d^7 + 13*B^2*C*a^2*b^8*c^4*d^5 + 6*B^2*C*a^2*b^8*c^6*d^3 + 32*B
^2*C*a^3*b^7*c^3*d^6 + 4*B^2*C*a^3*b^7*c^5*d^4 - 63*B^2*C*a^4*b^6*c^2*d^7 -
 73*B^2*C*a^4*b^6*c^4*d^5 - 3*B^2*C*a^4*b^6*c^6*d^3 + 44*B^2*C*a^5*b^5*c^3*

d^6 + 12*B^2*C*a^5*b^5*c^5*d^4 - 2*B^2*C*a^6*b^4*c^2*d^7 - 18*B^2*C*a^6*b^4
*c^4*d^5 - 12*B^2*C*a^7*b^3*c^3*d^6 + 3*B^2*C*a^8*b^2*c^2*d^7 - 18*A*B*C*a^
3*b^7*d^9 - 28*A*B*C*a^5*b^5*d^9 + 24*A*B*C*a^7*b^3*d^9 - 16*A*B*C*b^10*c^3
*d^6 + 6*A*B*C*b^10*c^5*d^4 - 16*A*B^2*a*b^9*c*d^8 + 12*A*C^2*a*b^9*c*d^8 -
24*A^2*C*a*b^9*c*d^8 + 4*B^2*C*a*b^9*c*d^8 - 4*A*B^2*a*b^9*c^3*d^6 + 16*A*
B^2*a*b^9*c^5*d^4 - 56*A*B^2*a^3*b^7*c*d^8 - 28*A*B^2*a^5*b^5*c*d^8 + 12*A*
B^2*a^7*b^3*c*d^8 - 4*A^2*B*a*b^9*c^2*d^7 - 33*A^2*B*a*b^9*c^4*d^5 + 20*A^2
*B*a^2*b^8*c*d^8 - 56*A^2*B*a^4*b^6*c*d^8 - 16*A^2*B*a^6*b^4*c*d^8 + 12*A*C
^2*a*b^9*c^3*d^6 - 24*A*C^2*a*b^9*c^5*d^4 + 36*A*C^2*a^3*b^7*c*d^8 - 24*A*C
^2*a^5*b^5*c*d^8 - 36*A^2*C*a*b^9*c^3*d^6 + 12*A^2*C*a*b^9*c^5*d^4 - 72*A^2
*C*a^3*b^7*c*d^8 + 24*A^2*C*a^5*b^5*c*d^8 - 10*B*C^2*a*b^9*c^2*d^7 - 12*B*C
^2*a*b^9*c^4*d^5 + 12*B*C^2*a*b^9*c^6*d^3 - 4*B*C^2*a^2*b^8*c*d^8 - 14*B*C^
2*a^4*b^6*c*d^8 - 4*B*C^2*a^6*b^4*c*d^8 + 6*B*C^2*a^8*b^2*c*d^8 - 8*B^2*C*a
*b^9*c^3*d^6 - 16*B^2*C*a*b^9*c^5*d^4 + 8*B^2*C*a^3*b^7*c*d^8 + 4*B^2*C*a^5
*b^5*c*d^8 - 24*B^2*C*a^7*b^3*c*d^8 - 76*A*B*C*a^2*b^8*c^3*d^6 - 20*A*B*C*a
^2*b^8*c^5*d^4 + 28*A*B*C*a^3*b^7*c^2*d^7 + 126*A*B*C*a^3*b^7*c^4*d^5 + 12*
A*B*C*a^3*b^7*c^6*d^3 - 16*A*B*C*a^4*b^6*c^3*d^6 - 42*A*B*C*a^4*b^6*c^5*d^4
- 32*A*B*C*a^5*b^5*c^2*d^7 + 48*A*B*C*a^5*b^5*c^4*d^5 + 12*A*B*C*a^6*b^4*c
^3*d^6 + 12*A*B*C*a^7*b^3*c^2*d^7 + 32*A*B*C*a*b^9*c^2*d^7 + 54*A*B*C*a*b^9
*c^4*d^5 - 12*A*B*C*a*b^9*c^6*d^3 - 16*A*B*C*a^2*b^8*c*d^8 + 70*A*B*C*a^4*b
^6*c*d^8 + 20*A*B*C*a^6*b^4*c*d^8 - 6*A*B*C*a^8*b^2*c*d^8))/(a^14*d^10 + b^
14*c^10 + 4*a^2*b^12*c^10 + 6*a^4*b^10*c^10 + 4*a^6*b^8*c^10 + a^8*b^6*c^10
+ a^6*b^8*d^10 + 4*a^8*b^6*d^10 + 6*a^10*b^4*d^10 + 4*a^12*b^2*d^10 + 2*a^
14*c^2*d^8 + a^14*c^4*d^6 + b^14*c^6*d^4 + 2*b^14*c^8*d^2 - 6*a*b^13*c^5*d^
5 - 12*a*b^13*c^7*d^3 - 24*a^3*b^11*c^9*d - 6*a^5*b^9*c*d^9 - 36*a^5*b^9*c^
9*d - 24*a^7*b^7*c*d^9 - 24*a^7*b^7*c^9*d - 36*a^9*b^5*c*d^9 - 6*a^9*b^5*c^
9*d - 24*a^11*b^3*c*d^9 - 12*a^13*b*c^3*d^7 - 6*a^13*b*c^5*d^5 + 15*a^2*b^1
2*c^4*d^6 + 34*a^2*b^12*c^6*d^4 + 23*a^2*b^12*c^8*d^2 - 20*a^3*b^11*c^3*d^7
- 64*a^3*b^11*c^5*d^5 - 68*a^3*b^11*c^7*d^3 + 15*a^4*b^10*c^2*d^8 + 90*a^4
*b^10*c^4*d^6 + 141*a^4*b^10*c^6*d^4 + 72*a^4*b^10*c^8*d^2 - 92*a^5*b^9*c^3
*d^7 - 202*a^5*b^9*c^5*d^5 - 152*a^5*b^9*c^7*d^3 + 62*a^6*b^8*c^2*d^8 + 211
*a^6*b^8*c^4*d^6 + 244*a^6*b^8*c^6*d^4 + 98*a^6*b^8*c^8*d^2 - 168*a^7*b^7*c
^3*d^7 - 288*a^7*b^7*c^5*d^5 - 168*a^7*b^7*c^7*d^3 + 98*a^8*b^6*c^2*d^8 + 2
44*a^8*b^6*c^4*d^6 + 211*a^8*b^6*c^6*d^4 + 62*a^8*b^6*c^8*d^2 - 152*a^9*b^5
*c^3*d^7 - 202*a^9*b^5*c^5*d^5 - 92*a^9*b^5*c^7*d^3 + 72*a^10*b^4*c^2*d^8 +
141*a^10*b^4*c^4*d^6 + 90*a^10*b^4*c^6*d^4 + 15*a^10*b^4*c^8*d^2 - 68*a^11
*b^3*c^3*d^7 - 64*a^11*b^3*c^5*d^5 - 20*a^11*b^3*c^7*d^3 + 23*a^12*b^2*c^2*
d^8 + 34*a^12*b^2*c^4*d^6 + 15*a^12*b^2*c^6*d^4 - 6*a*b^13*c^9*d - 6*a^13*b
*c*d^9))*root(640*a^13*b^7*c*d^15*f^4 + 640*a^7*b^13*c^15*d*f^4 + 480*a^15*
b^5*c*d^15*f^4 + 480*a^11*b^9*c*d^15*f^4 + 480*a^9*b^11*c^15*d*f^4 + 480*a^
5*b^15*c^15*d*f^4 + 192*a^19*b*c^5*d^11*f^4 + 192*a^17*b^3*c*d^15*f^4 + 192
*a^11*b^9*c^15*d*f^4 + 192*a^9*b^11*c*d^15*f^4 + 192*a^3*b^17*c^15*d*f^4 +
192*a*b^19*c^11*d^5*f^4 + 128*a^19*b*c^7*d^9*f^4 + 128*a^19*b*c^3*d^13*f^4
+ 128*a*b^19*c^13*d^3*f^4 + 128*a*b^19*c^9*d^7*f^4 + 32*a^19*b*c^9*d^7*f^4
+ 32*a^13*b^7*c^15*d*f^4 + 32*a^7*b^13*c*d^15*f^4 + 32*a*b^19*c^7*d^9*f^4 +

32*a^19*b*c*d^15*f^4 + 32*a*b^19*c^15*d*f^4 - 47088*a^10*b^10*c^8*d^8*f^4
+ 42432*a^11*b^9*c^7*d^9*f^4 + 42432*a^9*b^11*c^9*d^7*f^4 + 39328*a^11*b^9*
c^9*d^7*f^4 + 39328*a^9*b^11*c^7*d^9*f^4 - 36912*a^12*b^8*c^8*d^8*f^4 - 369
12*a^8*b^12*c^8*d^8*f^4 - 34256*a^10*b^10*c^10*d^6*f^4 - 34256*a^10*b^10*c^
6*d^10*f^4 - 31152*a^12*b^8*c^6*d^10*f^4 - 31152*a^8*b^12*c^10*d^6*f^4 + 28
128*a^13*b^7*c^7*d^9*f^4 + 28128*a^7*b^13*c^9*d^7*f^4 + 24160*a^11*b^9*c^5*
d^11*f^4 + 24160*a^9*b^11*c^11*d^5*f^4 - 23088*a^12*b^8*c^10*d^6*f^4 - 2308
8*a^8*b^12*c^6*d^10*f^4 + 22272*a^13*b^7*c^9*d^7*f^4 + 22272*a^7*b^13*c^7*d
^9*f^4 + 19072*a^11*b^9*c^11*d^5*f^4 + 19072*a^9*b^11*c^5*d^11*f^4 + 18624*
a^13*b^7*c^5*d^11*f^4 + 18624*a^7*b^13*c^11*d^5*f^4 - 17328*a^14*b^6*c^8*d^
8*f^4 - 17328*a^6*b^14*c^8*d^8*f^4 - 17232*a^14*b^6*c^6*d^10*f^4 - 17232*a^
6*b^14*c^10*d^6*f^4 - 13520*a^12*b^8*c^4*d^12*f^4 - 13520*a^8*b^12*c^12*d^4
*f^4 - 12464*a^10*b^10*c^12*d^4*f^4 - 12464*a^10*b^10*c^4*d^12*f^4 + 10880*
a^15*b^5*c^7*d^9*f^4 + 10880*a^5*b^15*c^9*d^7*f^4 - 9072*a^14*b^6*c^10*d^6*
f^4 - 9072*a^6*b^14*c^6*d^10*f^4 + 8928*a^13*b^7*c^11*d^5*f^4 + 8928*a^7*b^
13*c^5*d^11*f^4 - 8880*a^14*b^6*c^4*d^12*f^4 - 8880*a^6*b^14*c^12*d^4*f^4 +
8480*a^15*b^5*c^5*d^11*f^4 + 8480*a^5*b^15*c^11*d^5*f^4 + 7200*a^15*b^5*c^
9*d^7*f^4 + 7200*a^5*b^15*c^7*d^9*f^4 - 6912*a^12*b^8*c^12*d^4*f^4 - 6912*a
^8*b^12*c^4*d^12*f^4 + 6400*a^11*b^9*c^3*d^13*f^4 + 6400*a^9*b^11*c^13*d^3*
f^4 + 5920*a^13*b^7*c^3*d^13*f^4 + 5920*a^7*b^13*c^13*d^3*f^4 - 5392*a^16*b
^4*c^6*d^10*f^4 - 5392*a^4*b^16*c^10*d^6*f^4 - 4428*a^16*b^4*c^8*d^8*f^4 -
4428*a^4*b^16*c^8*d^8*f^4 + 4128*a^11*b^9*c^13*d^3*f^4 + 4128*a^9*b^11*c^3*
d^13*f^4 - 3328*a^16*b^4*c^4*d^12*f^4 - 3328*a^4*b^16*c^12*d^4*f^4 + 3264*a
^15*b^5*c^3*d^13*f^4 + 3264*a^5*b^15*c^13*d^3*f^4 - 2480*a^12*b^8*c^2*d^14*
f^4 - 2480*a^8*b^12*c^14*d^2*f^4 + 2240*a^15*b^5*c^11*d^5*f^4 + 2240*a^5*b^
15*c^5*d^11*f^4 - 2128*a^14*b^6*c^12*d^4*f^4 - 2128*a^6*b^14*c^4*d^12*f^4 +
2112*a^17*b^3*c^7*d^9*f^4 + 2112*a^3*b^17*c^9*d^7*f^4 + 2048*a^17*b^3*c^5*
d^11*f^4 + 2048*a^3*b^17*c^11*d^5*f^4 - 2000*a^14*b^6*c^2*d^14*f^4 - 2000*a
^6*b^14*c^14*d^2*f^4 - 1792*a^16*b^4*c^10*d^6*f^4 - 1792*a^4*b^16*c^6*d^10*
f^4 - 1776*a^10*b^10*c^14*d^2*f^4 - 1776*a^10*b^10*c^2*d^14*f^4 + 1472*a^13
*b^7*c^13*d^3*f^4 + 1472*a^7*b^13*c^3*d^13*f^4 + 1088*a^17*b^3*c^9*d^7*f^4
+ 1088*a^3*b^17*c^7*d^9*f^4 + 992*a^17*b^3*c^3*d^13*f^4 + 992*a^3*b^17*c^13
*d^3*f^4 - 912*a^16*b^4*c^2*d^14*f^4 - 912*a^4*b^16*c^14*d^2*f^4 - 768*a^18
*b^2*c^6*d^10*f^4 - 768*a^2*b^18*c^10*d^6*f^4 - 688*a^12*b^8*c^14*d^2*f^4 -
688*a^8*b^12*c^2*d^14*f^4 - 592*a^18*b^2*c^4*d^12*f^4 - 592*a^2*b^18*c^12*
d^4*f^4 - 472*a^18*b^2*c^8*d^8*f^4 - 472*a^2*b^18*c^8*d^8*f^4 - 280*a^16*b^
4*c^12*d^4*f^4 - 280*a^4*b^16*c^4*d^12*f^4 + 224*a^17*b^3*c^11*d^5*f^4 + 22
4*a^15*b^5*c^13*d^3*f^4 + 224*a^5*b^15*c^3*d^13*f^4 + 224*a^3*b^17*c^5*d^11
*f^4 - 208*a^18*b^2*c^2*d^14*f^4 - 208*a^2*b^18*c^14*d^2*f^4 - 112*a^18*b^2
*c^10*d^6*f^4 - 112*a^14*b^6*c^14*d^2*f^4 - 112*a^6*b^14*c^2*d^14*f^4 - 112
*a^2*b^18*c^6*d^10*f^4 - 24*b^20*c^12*d^4*f^4 - 16*b^20*c^14*d^2*f^4 - 16*b
^20*c^10*d^6*f^4 - 4*b^20*c^8*d^8*f^4 - 24*a^20*c^4*d^12*f^4 - 16*a^20*c^6*
d^10*f^4 - 16*a^20*c^2*d^14*f^4 - 4*a^20*c^8*d^8*f^4 - 80*a^14*b^6*d^16*f^4
- 60*a^16*b^4*d^16*f^4 - 60*a^12*b^8*d^16*f^4 - 24*a^18*b^2*d^16*f^4 - 24*
a^10*b^10*d^16*f^4 - 4*a^8*b^12*d^16*f^4 - 80*a^6*b^14*c^16*f^4 - 60*a^8*b^

12*c^16*f^4 - 60*a^4*b^16*c^16*f^4 - 24*a^10*b^10*c^16*f^4 - 24*a^2*b^18*c^
16*f^4 - 4*a^12*b^8*c^16*f^4 - 4*b^20*c^16*f^4 - 4*a^20*d^16*f^4 + 56*A*C*a
^13*b*c*d^11*f^2 - 48*A*C*a*b^13*c^11*d*f^2 + 48*A*C*a*b^13*c*d^11*f^2 + 59
04*B*C*a^7*b^7*c^6*d^6*f^2 - 5016*B*C*a^8*b^6*c^5*d^7*f^2 - 4608*B*C*a^6*b^
8*c^7*d^5*f^2 - 4512*B*C*a^6*b^8*c^5*d^7*f^2 - 4384*B*C*a^8*b^6*c^7*d^5*f^2
 + 3056*B*C*a^7*b^7*c^8*d^4*f^2 + 2256*B*C*a^7*b^7*c^4*d^8*f^2 - 1824*B*C*a
^8*b^6*c^3*d^9*f^2 + 1632*B*C*a^4*b^10*c^9*d^3*f^2 - 1400*B*C*a^3*b^11*c^8*
d^4*f^2 - 1320*B*C*a^11*b^3*c^4*d^8*f^2 - 1248*B*C*a^6*b^8*c^3*d^9*f^2 + 11
52*B*C*a^10*b^4*c^3*d^9*f^2 - 1072*B*C*a^6*b^8*c^9*d^3*f^2 + 1068*B*C*a^9*b
^5*c^6*d^6*f^2 - 1004*B*C*a^5*b^9*c^4*d^8*f^2 - 968*B*C*a^3*b^11*c^6*d^6*f^
2 - 864*B*C*a^5*b^9*c^8*d^4*f^2 - 828*B*C*a^9*b^5*c^4*d^8*f^2 - 792*B*C*a^1
1*b^3*c^2*d^10*f^2 - 792*B*C*a^3*b^11*c^4*d^8*f^2 - 776*B*C*a^8*b^6*c^9*d^3
*f^2 + 688*B*C*a^4*b^10*c^7*d^5*f^2 - 672*B*C*a^3*b^11*c^10*d^2*f^2 - 592*B
*C*a^9*b^5*c^2*d^10*f^2 + 544*B*C*a^7*b^7*c^10*d^2*f^2 - 492*B*C*a^5*b^9*c^
2*d^10*f^2 + 480*B*C*a^10*b^4*c^5*d^7*f^2 - 392*B*C*a^5*b^9*c^10*d^2*f^2 +
332*B*C*a^9*b^5*c^8*d^4*f^2 - 328*B*C*a^11*b^3*c^6*d^6*f^2 + 320*B*C*a^2*b^
12*c^9*d^3*f^2 + 272*B*C*a^12*b^2*c^3*d^9*f^2 - 248*B*C*a^4*b^10*c^5*d^7*f^
2 - 248*B*C*a^3*b^11*c^2*d^10*f^2 - 208*B*C*a^10*b^4*c^7*d^5*f^2 - 192*B*C*
a^2*b^12*c^5*d^7*f^2 + 144*B*C*a^7*b^7*c^2*d^10*f^2 - 96*B*C*a^4*b^10*c^3*d
^9*f^2 + 88*B*C*a^12*b^2*c^5*d^7*f^2 - 72*B*C*a^11*b^3*c^8*d^4*f^2 - 48*B*C
*a^12*b^2*c^7*d^5*f^2 + 48*B*C*a^10*b^4*c^9*d^3*f^2 - 48*B*C*a^2*b^12*c^7*d
^5*f^2 - 48*B*C*a^2*b^12*c^3*d^9*f^2 - 12*B*C*a^9*b^5*c^10*d^2*f^2 + 4*B*C*
a^5*b^9*c^6*d^6*f^2 + 5824*A*C*a^5*b^9*c^7*d^5*f^2 - 4378*A*C*a^6*b^8*c^8*d
^4*f^2 + 4296*A*C*a^5*b^9*c^5*d^7*f^2 - 3912*A*C*a^6*b^8*c^6*d^6*f^2 - 3672
*A*C*a^9*b^5*c^5*d^7*f^2 + 3594*A*C*a^8*b^6*c^4*d^8*f^2 + 3236*A*C*a^8*b^6*
c^6*d^6*f^2 + 2816*A*C*a^5*b^9*c^9*d^3*f^2 + 2624*A*C*a^5*b^9*c^3*d^9*f^2 +
 2432*A*C*a^7*b^7*c^7*d^5*f^2 - 2366*A*C*a^4*b^10*c^8*d^4*f^2 + 2298*A*C*a^
10*b^4*c^4*d^8*f^2 + 1872*A*C*a^7*b^7*c^3*d^9*f^2 + 1848*A*C*a^10*b^4*c^6*d
^6*f^2 - 1644*A*C*a^4*b^10*c^6*d^6*f^2 - 1488*A*C*a^9*b^5*c^7*d^5*f^2 - 140
8*A*C*a^9*b^5*c^3*d^9*f^2 - 1308*A*C*a^6*b^8*c^4*d^8*f^2 + 1248*A*C*a^7*b^7
*c^5*d^7*f^2 - 1012*A*C*a^6*b^8*c^10*d^2*f^2 + 1008*A*C*a^3*b^11*c^7*d^5*f^
2 + 992*A*C*a^3*b^11*c^5*d^7*f^2 + 928*A*C*a^3*b^11*c^3*d^9*f^2 + 848*A*C*a
^7*b^7*c^9*d^3*f^2 + 636*A*C*a^8*b^6*c^2*d^10*f^2 - 628*A*C*a^4*b^10*c^10*d
^2*f^2 - 600*A*C*a^6*b^8*c^2*d^10*f^2 - 576*A*C*a^11*b^3*c^5*d^7*f^2 + 572*
A*C*a^10*b^4*c^2*d^10*f^2 + 464*A*C*a^8*b^6*c^8*d^4*f^2 - 304*A*C*a^4*b^10*
c^4*d^8*f^2 + 304*A*C*a^2*b^12*c^6*d^6*f^2 + 296*A*C*a^2*b^12*c^4*d^8*f^2 +
 260*A*C*a^10*b^4*c^8*d^4*f^2 - 232*A*C*a^12*b^2*c^2*d^10*f^2 - 232*A*C*a^9
*b^5*c^9*d^3*f^2 + 228*A*C*a^2*b^12*c^10*d^2*f^2 - 188*A*C*a^4*b^10*c^2*d^1
0*f^2 + 144*A*C*a^11*b^3*c^3*d^9*f^2 + 116*A*C*a^12*b^2*c^6*d^6*f^2 - 112*A
*C*a^11*b^3*c^7*d^5*f^2 + 112*A*C*a^3*b^11*c^9*d^3*f^2 + 92*A*C*a^8*b^6*c^1
0*d^2*f^2 + 74*A*C*a^12*b^2*c^4*d^8*f^2 + 62*A*C*a^2*b^12*c^8*d^4*f^2 + 40*
A*C*a^2*b^12*c^2*d^10*f^2 - 7008*A*B*a^7*b^7*c^6*d^6*f^2 - 4032*A*B*a^7*b^7
*c^4*d^8*f^2 + 3952*A*B*a^8*b^6*c^7*d^5*f^2 + 3648*A*B*a^8*b^6*c^5*d^7*f^2
- 3392*A*B*a^7*b^7*c^8*d^4*f^2 + 3264*A*B*a^6*b^8*c^7*d^5*f^2 - 2992*A*B*a^
4*b^10*c^5*d^7*f^2 - 2368*A*B*a^4*b^10*c^7*d^5*f^2 - 2304*A*B*a^4*b^10*c^3*

d^9*f^2 - 1968*A*B*a^9*b^5*c^6*d^6*f^2 - 1872*A*B*a^4*b^10*c^9*d^3*f^2 - 17
28*A*B*a^7*b^7*c^2*d^10*f^2 + 1712*A*B*a^3*b^11*c^8*d^4*f^2 - 1536*A*B*a^10
*b^4*c^3*d^9*f^2 + 1536*A*B*a^6*b^8*c^5*d^7*f^2 - 1392*A*B*a^2*b^12*c^5*d^7
*f^2 + 1328*A*B*a^3*b^11*c^6*d^6*f^2 - 1104*A*B*a^2*b^12*c^3*d^9*f^2 - 1056
*A*B*a^6*b^8*c^3*d^9*f^2 + 976*A*B*a^6*b^8*c^9*d^3*f^2 + 960*A*B*a^11*b^3*c
^4*d^8*f^2 + 936*A*B*a^5*b^9*c^8*d^4*f^2 - 912*A*B*a^10*b^4*c^5*d^7*f^2 + 8
48*A*B*a^8*b^6*c^9*d^3*f^2 + 816*A*B*a^3*b^11*c^4*d^8*f^2 - 816*A*B*a^2*b^1
2*c^7*d^5*f^2 + 768*A*B*a^3*b^11*c^10*d^2*f^2 + 672*A*B*a^8*b^6*c^3*d^9*f^2
 - 632*A*B*a^9*b^5*c^8*d^4*f^2 - 608*A*B*a^9*b^5*c^2*d^10*f^2 - 552*A*B*a^9
*b^5*c^4*d^8*f^2 - 544*A*B*a^7*b^7*c^10*d^2*f^2 - 480*A*B*a^5*b^9*c^2*d^10*
f^2 + 464*A*B*a^5*b^9*c^10*d^2*f^2 - 464*A*B*a^2*b^12*c^9*d^3*f^2 + 432*A*B
*a^11*b^3*c^2*d^10*f^2 - 368*A*B*a^12*b^2*c^3*d^9*f^2 - 256*A*B*a^5*b^9*c^6
*d^6*f^2 - 208*A*B*a^12*b^2*c^5*d^7*f^2 + 176*A*B*a^5*b^9*c^4*d^8*f^2 + 112
*A*B*a^11*b^3*c^6*d^6*f^2 + 112*A*B*a^10*b^4*c^7*d^5*f^2 - 16*A*B*a^3*b^11*
c^2*d^10*f^2 - 576*B*C*a^8*b^6*c*d^11*f^2 + 400*B*C*a^4*b^10*c^11*d*f^2 - 2
88*B*C*a^6*b^8*c*d^11*f^2 - 176*B*C*a^6*b^8*c^11*d*f^2 + 128*B*C*a^10*b^4*c
*d^11*f^2 - 108*B*C*a*b^13*c^4*d^8*f^2 - 104*B*C*a^4*b^10*c*d^11*f^2 - 92*B
*C*a^13*b*c^4*d^8*f^2 - 60*B*C*a*b^13*c^8*d^4*f^2 - 60*B*C*a*b^13*c^6*d^6*f
^2 + 48*B*C*a^2*b^12*c^11*d*f^2 - 40*B*C*a*b^13*c^2*d^10*f^2 - 28*B*C*a^13*
b*c^2*d^10*f^2 - 24*B*C*a^12*b^2*c*d^11*f^2 + 20*B*C*a*b^13*c^10*d^2*f^2 -
16*B*C*a^2*b^12*c*d^11*f^2 + 12*B*C*a^13*b*c^6*d^6*f^2 + 912*A*C*a^7*b^7*c*
d^11*f^2 + 808*A*C*a^5*b^9*c*d^11*f^2 + 432*A*C*a^5*b^9*c^11*d*f^2 + 336*A*
C*a^3*b^11*c*d^11*f^2 + 224*A*C*a^11*b^3*c*d^11*f^2 - 112*A*C*a^3*b^11*c^11
*d*f^2 + 112*A*C*a*b^13*c^3*d^9*f^2 - 88*A*C*a*b^13*c^9*d^3*f^2 + 80*A*C*a^
13*b*c^3*d^9*f^2 + 56*A*C*a*b^13*c^5*d^7*f^2 + 48*A*C*a^9*b^5*c*d^11*f^2 -
40*A*C*a^13*b*c^5*d^7*f^2 - 16*A*C*a^7*b^7*c^11*d*f^2 + 16*A*C*a*b^13*c^7*d
^5*f^2 - 496*A*B*a^4*b^10*c*d^11*f^2 - 400*A*B*a^4*b^10*c^11*d*f^2 + 288*A*
B*a^8*b^6*c*d^11*f^2 - 288*A*B*a^6*b^8*c*d^11*f^2 - 272*A*B*a^2*b^12*c*d^11
*f^2 + 240*A*B*a*b^13*c^6*d^6*f^2 - 224*A*B*a^10*b^4*c*d^11*f^2 + 192*A*B*a
*b^13*c^8*d^4*f^2 + 192*A*B*a*b^13*c^4*d^8*f^2 + 176*A*B*a^6*b^8*c^11*d*f^2
 + 104*A*B*a^13*b*c^4*d^8*f^2 - 48*A*B*a^2*b^12*c^11*d*f^2 + 16*A*B*a^13*b*
c^2*d^10*f^2 + 16*A*B*a*b^13*c^10*d^2*f^2 + 16*A*B*a*b^13*c^2*d^10*f^2 - 96
*B*C*b^14*c^7*d^5*f^2 - 72*B*C*b^14*c^5*d^7*f^2 - 24*B*C*b^14*c^9*d^3*f^2 -
 16*B*C*b^14*c^3*d^9*f^2 + 116*A*C*b^14*c^6*d^6*f^2 + 100*A*C*b^14*c^4*d^8*
f^2 + 24*A*C*b^14*c^2*d^10*f^2 + 22*A*C*b^14*c^8*d^4*f^2 + 16*B*C*a^14*c^3*
d^9*f^2 + 8*A*C*b^14*c^10*d^2*f^2 - 192*A*B*b^14*c^5*d^7*f^2 - 176*A*B*b^14
*c^3*d^9*f^2 - 112*B*C*a^11*b^3*d^12*f^2 - 48*A*B*b^14*c^7*d^5*f^2 - 28*A*C
*a^14*c^2*d^10*f^2 + 4*B*C*a^5*b^9*d^12*f^2 + 2*A*C*a^14*c^4*d^8*f^2 + 150*
A*C*a^10*b^4*d^12*f^2 - 80*B*C*a^3*b^11*c^12*f^2 + 66*A*C*a^8*b^6*d^12*f^2
 - 30*A*C*a^12*b^2*d^12*f^2 + 24*B*C*a^5*b^9*c^12*f^2 - 16*A*B*a^14*c^3*d^9*
f^2 - 12*A*C*a^4*b^10*d^12*f^2 - 576*A*B*a^7*b^7*d^12*f^2 - 432*A*B*a^9*b^5
*d^12*f^2 - 400*A*B*a^5*b^9*d^12*f^2 - 144*A*B*a^3*b^11*d^12*f^2 - 66*A*C*a
^4*b^10*c^12*f^2 + 54*A*C*a^2*b^12*c^12*f^2 - 32*A*B*a^11*b^3*d^12*f^2 + 2*
A*C*a^6*b^8*c^12*f^2 + 80*A*B*a^3*b^11*c^12*f^2 - 24*A*B*a^5*b^9*c^12*f^2 +
 2508*C^2*a^6*b^8*c^6*d^6*f^2 + 2376*C^2*a^9*b^5*c^5*d^7*f^2 + 2357*C^2*a^6

*b^8*c^8*d^4*f^2 - 2048*C^2*a^5*b^9*c^7*d^5*f^2 + 1304*C^2*a^9*b^5*c^3*d^9*f^2 + 1303*C^2*a^4*b^10*c^8*d^4*f^2 + 1212*C^2*a^4*b^10*c^6*d^6*f^2 - 1203*C^2*a^8*b^6*c^4*d^8*f^2 - 1192*C^2*a^5*b^9*c^9*d^3*f^2 + 1062*C^2*a^6*b^8*c^4*d^8*f^2 + 984*C^2*a^9*b^5*c^7*d^5*f^2 - 952*C^2*a^8*b^6*c^6*d^6*f^2 + 768*C^2*a^7*b^7*c^5*d^7*f^2 - 681*C^2*a^10*b^4*c^4*d^8*f^2 - 672*C^2*a^5*b^9*c^5*d^7*f^2 - 480*C^2*a^10*b^4*c^6*d^6*f^2 + 458*C^2*a^6*b^8*c^10*d^2*f^2 - 448*C^2*a^7*b^7*c^7*d^5*f^2 + 422*C^2*a^4*b^10*c^4*d^8*f^2 + 372*C^2*a^6*b^8*c^2*d^10*f^2 + 360*C^2*a^11*b^3*c^5*d^7*f^2 + 312*C^2*a^7*b^7*c^3*d^9*f^2 + 278*C^2*a^4*b^10*c^10*d^2*f^2 - 232*C^2*a^7*b^7*c^9*d^3*f^2 + 194*C^2*a^12*b^2*c^2*d^10*f^2 + 176*C^2*a^9*b^5*c^9*d^3*f^2 + 152*C^2*a^3*b^11*c^5*d^7*f^2 + 124*C^2*a^4*b^10*c^2*d^10*f^2 - 120*C^2*a^3*b^11*c^7*d^5*f^2 - 114*C^2*a^2*b^12*c^10*d^2*f^2 - 102*C^2*a^8*b^6*c^2*d^10*f^2 + 101*C^2*a^12*b^2*c^4*d^8*f^2 + 100*C^2*a^2*b^12*c^6*d^6*f^2 - 88*C^2*a^5*b^9*c^3*d^9*f^2 + 77*C^2*a^2*b^12*c^8*d^4*f^2 + 72*C^2*a^11*b^3*c^3*d^9*f^2 - 64*C^2*a^8*b^6*c^10*d^2*f^2 + 64*C^2*a^3*b^11*c^3*d^9*f^2 - 58*C^2*a^10*b^4*c^2*d^10*f^2 + 56*C^2*a^12*b^2*c^6*d^6*f^2 + 56*C^2*a^11*b^3*c^7*d^5*f^2 + 40*C^2*a^3*b^11*c^9*d^3*f^2 + 36*C^2*a^12*b^2*c^8*d^4*f^2 + 32*C^2*a^2*b^12*c^4*d^8*f^2 + 26*C^2*a^10*b^4*c^8*d^4*f^2 + 16*C^2*a^2*b^12*c^2*d^10*f^2 + 2*C^2*a^8*b^6*c^8*d^4*f^2 + 2277*B^2*a^8*b^6*c^4*d^8*f^2 + 2144*B^2*a^5*b^9*c^7*d^5*f^2 - 2112*B^2*a^9*b^5*c^5*d^7*f^2 + 2028*B^2*a^8*b^6*c^6*d^6*f^2 - 1671*B^2*a^6*b^8*c^8*d^4*f^2 + 1275*B^2*a^10*b^4*c^4*d^8*f^2 + 1176*B^2*a^5*b^9*c^5*d^7*f^2 + 1096*B^2*a^5*b^9*c^9*d^3*f^2 - 1044*B^2*a^6*b^8*c^6*d^6*f^2 + 984*B^2*a^10*b^4*c^6*d^6*f^2 - 968*B^2*a^9*b^5*c^3*d^9*f^2 - 888*B^2*a^9*b^5*c^7*d^5*f^2 + 672*B^2*a^7*b^7*c^7*d^5*f^2 + 664*B^2*a^5*b^9*c^3*d^9*f^2 - 649*B^2*a^4*b^10*c^8*d^4*f^2 + 618*B^2*a^8*b^6*c^2*d^10*f^2 + 514*B^2*a^4*b^10*c^4*d^8*f^2 + 460*B^2*a^2*b^12*c^6*d^6*f^2 + 422*B^2*a^8*b^6*c^8*d^4*f^2 + 406*B^2*a^10*b^4*c^2*d^10*f^2 - 382*B^2*a^6*b^8*c^10*d^2*f^2 + 368*B^2*a^2*b^12*c^4*d^8*f^2 - 312*B^2*a^11*b^3*c^5*d^7*f^2 + 312*B^2*a^7*b^7*c^3*d^9*f^2 + 248*B^2*a^7*b^7*c^9*d^3*f^2 + 245*B^2*a^2*b^12*c^8*d^4*f^2 - 192*B^2*a^7*b^7*c^5*d^7*f^2 - 184*B^2*a^3*b^11*c^9*d^3*f^2 + 182*B^2*a^2*b^12*c^10*d^2*f^2 + 176*B^2*a^3*b^11*c^3*d^9*f^2 + 174*B^2*a^6*b^8*c^4*d^8*f^2 - 170*B^2*a^4*b^10*c^10*d^2*f^2 - 152*B^2*a^9*b^5*c^9*d^3*f^2 + 152*B^2*a^4*b^10*c^2*d^10*f^2 + 142*B^2*a^10*b^4*c^8*d^4*f^2 - 90*B^2*a^12*b^2*c^2*d^10*f^2 + 88*B^2*a^2*b^12*c^2*d^10*f^2 + 84*B^2*a^8*b^6*c^10*d^2*f^2 + 84*B^2*a^6*b^8*c^2*d^10*f^2 + 60*B^2*a^12*b^2*c^6*d^6*f^2 - 56*B^2*a^11*b^3*c^7*d^5*f^2 + 53*B^2*a^12*b^2*c^4*d^8*f^2 + 24*B^2*a^11*b^3*c^3*d^9*f^2 + 24*B^2*a^4*b^10*c^6*d^6*f^2 + 24*B^2*a^3*b^11*c^7*d^5*f^2 - 8*B^2*a^3*b^11*c^5*d^7*f^2 + 4566*A^2*a^6*b^8*c^4*d^8*f^2 + 4284*A^2*a^6*b^8*c^6*d^6*f^2 - 3776*A^2*a^5*b^9*c^7*d^5*f^2 - 3624*A^2*a^5*b^9*c^5*d^7*f^2 + 3122*A^2*a^4*b^10*c^4*d^8*f^2 + 3108*A^2*a^6*b^8*c^2*d^10*f^2 + 2741*A^2*a^6*b^8*c^8*d^4*f^2 + 2592*A^2*a^4*b^10*c^6*d^6*f^2 - 2536*A^2*a^5*b^9*c^3*d^9*f^2 + 2224*A^2*a^4*b^10*c^2*d^10*f^2 - 2184*A^2*a^7*b^7*c^3*d^9*f^2 - 2016*A^2*a^7*b^7*c^5*d^7*f^2 - 1984*A^2*a^7*b^7*c^7*d^5*f^2 + 1626*A^2*a^8*b^6*c^2*d^10*f^2 - 1624*A^2*a^5*b^9*c^9*d^3*f^2 + 1603*A^2*a^4*b^10*c^8*d^4*f^2 + 1296*A^2*a^9*b^5*c^5*d^7*f^2 - 1144*A^2*a^3*b^11*c^5*d^7*f^2 - 992*A^2*a^3*b^11*c^3*d^9*f^2 + 9

68*A^2*a^2*b^12*c^4*d^8*f^2 - 888*A^2*a^3*b^11*c^7*d^5*f^2 + 849*A^2*a^8*b^6*c^4*d^8*f^2 + 808*A^2*a^2*b^12*c^2*d^10*f^2 - 616*A^2*a^7*b^7*c^9*d^3*f^2 + 554*A^2*a^6*b^8*c^10*d^2*f^2 - 504*A^2*a^10*b^4*c^6*d^6*f^2 + 504*A^2*a^9*b^5*c^7*d^5*f^2 + 460*A^2*a^2*b^12*c^6*d^6*f^2 + 350*A^2*a^10*b^4*c^2*d^10*f^2 + 350*A^2*a^4*b^10*c^10*d^2*f^2 - 321*A^2*a^10*b^4*c^4*d^8*f^2 + 216*A^2*a^11*b^3*c^5*d^7*f^2 - 216*A^2*a^11*b^3*c^3*d^9*f^2 + 182*A^2*a^12*b^2*c^2*d^10*f^2 - 152*A^2*a^3*b^11*c^9*d^3*f^2 - 124*A^2*a^8*b^6*c^6*d^6*f^2 - 114*A^2*a^2*b^12*c^10*d^2*f^2 + 104*A^2*a^9*b^5*c^3*d^9*f^2 + 77*A^2*a^2*b^12*c^8*d^4*f^2 + 74*A^2*a^8*b^6*c^8*d^4*f^2 - 70*A^2*a^10*b^4*c^8*d^4*f^2 + 56*A^2*a^11*b^3*c^7*d^5*f^2 + 56*A^2*a^9*b^5*c^9*d^3*f^2 + 41*A^2*a^12*b^2*c^4*d^8*f^2 - 28*A^2*a^12*b^2*c^6*d^6*f^2 - 28*A^2*a^8*b^6*c^10*d^2*f^2 - 16*B*C*b^14*c^11*d*f^2 - 16*B*C*a^14*c*d^11*f^2 - 48*A*B*b^14*c*d^11*f^2 + 16*A*B*b^14*c^11*d*f^2 + 12*B*C*a^13*b*d^12*f^2 + 24*B*C*a*b^13*c^12*f^2 + 16*A*B*a^14*c*d^11*f^2 - 24*A*B*a^13*b*d^12*f^2 - 24*A*B*a*b^13*d^12*f^2 - 24*A*B*a*b^13*c^12*f^2 + 216*C^2*a^9*b^5*c*d^11*f^2 - 216*C^2*a^5*b^9*c^11*d*f^2 + 56*C^2*a^3*b^11*c^11*d*f^2 + 56*C^2*a*b^13*c^9*d^3*f^2 + 56*C^2*a*b^13*c^5*d^7*f^2 - 40*C^2*a^11*b^3*c*d^11*f^2 + 40*C^2*a*b^13*c^7*d^5*f^2 + 32*C^2*a^13*b*c^5*d^7*f^2 - 24*C^2*a^7*b^7*c*d^11*f^2 - 16*C^2*a^13*b*c^3*d^9*f^2 + 16*C^2*a*b^13*c^3*d^9*f^2 + 8*C^2*a^7*b^7*c^11*d*f^2 - 8*C^2*a^5*b^9*c*d^11*f^2 + 264*B^2*a^7*b^7*c*d^11*f^2 + 224*B^2*a^5*b^9*c*d^11*f^2 + 168*B^2*a^5*b^9*c^11*d*f^2 - 112*B^2*a*b^13*c^9*d^3*f^2 - 104*B^2*a^3*b^11*c^11*d*f^2 - 104*B^2*a*b^13*c^7*d^5*f^2 + 96*B^2*a^3*b^11*c*d^11*f^2 + 88*B^2*a^11*b^3*c*d^11*f^2 - 72*B^2*a^9*b^5*c*d^11*f^2 - 64*B^2*a*b^13*c^5*d^7*f^2 + 32*B^2*a^13*b*c^3*d^9*f^2 - 24*B^2*a^13*b*c^5*d^7*f^2 - 24*B^2*a^7*b^7*c^11*d*f^2 + 16*B^2*a*b^13*c^3*d^9*f^2 - 888*A^2*a^7*b^7*c*d^11*f^2 - 800*A^2*a^5*b^9*c*d^11*f^2 - 336*A^2*a^3*b^11*c*d^11*f^2 - 264*A^2*a^9*b^5*c*d^11*f^2 - 216*A^2*a^5*b^9*c^11*d*f^2 - 184*A^2*a^11*b^3*c*d^11*f^2 - 128*A^2*a*b^13*c^3*d^9*f^2 - 112*A^2*a*b^13*c^5*d^7*f^2 - 64*A^2*a^13*b*c^3*d^9*f^2 + 56*A^2*a^3*b^11*c^11*d*f^2 - 56*A^2*a*b^13*c^7*d^5*f^2 + 32*A^2*a*b^13*c^9*d^3*f^2 + 8*A^2*a^13*b*c^5*d^7*f^2 + 8*A^2*a^7*b^7*c^11*d*f^2 + 24*C^2*a*b^13*c^11*d*f^2 - 16*C^2*a^13*b*c*d^11*f^2 - 40*B^2*a*b^13*c^11*d*f^2 + 24*B^2*a^13*b*c*d^11*f^2 + 16*B^2*a*b^13*c*d^11*f^2 - 48*A^2*a*b^13*c*d^11*f^2 - 40*A^2*a^13*b*c*d^11*f^2 + 24*A^2*a*b^13*c^11*d*f^2 - 6*A*C*b^14*c^12*f^2 + 2*A*C*a^14*d^12*f^2 + 31*C^2*b^14*c^8*d^4*f^2 + 20*C^2*b^14*c^6*d^6*f^2 + 4*C^2*b^14*c^4*d^8*f^2 + 2*C^2*b^14*c^10*d^2*f^2 + 80*B^2*b^14*c^6*d^6*f^2 + 64*B^2*b^14*c^4*d^8*f^2 + 31*B^2*b^14*c^8*d^4*f^2 + 16*B^2*b^14*c^2*d^10*f^2 + 14*C^2*a^14*c^2*d^10*f^2 + 14*B^2*b^14*c^10*d^2*f^2 - C^2*a^14*c^4*d^8*f^2 + 120*A^2*b^14*c^2*d^10*f^2 + 112*A^2*b^14*c^4*d^8*f^2 + 33*C^2*a^12*b^2*d^12*f^2 - 27*C^2*a^10*b^4*d^12*f^2 - 17*A^2*b^14*c^8*d^4*f^2 - 10*B^2*a^14*c^2*d^10*f^2 - 10*A^2*b^14*c^10*d^2*f^2 + 8*A^2*b^14*c^6*d^6*f^2 + 3*C^2*a^8*b^6*d^12*f^2 + 3*B^2*a^14*c^4*d^8*f^2 + 117*B^2*a^10*b^4*d^12*f^2 + 111*B^2*a^8*b^6*d^12*f^2 + 72*B^2*a^6*b^8*d^12*f^2 + 33*C^2*a^4*b^10*c^12*f^2 - 27*C^2*a^2*b^12*c^12*f^2 + 24*B^2*a^4*b^10*d^12*f^2 + 14*A^2*a^14*c^2*d^10*f^2 + 4*B^2*a^2*b^12*d^12*f^2 - 3*B^2*a^12*b^2*d^12*f^2 - C^2*a^6*b^8*c^12*f^2 - A^2*a^14*c^4*d^8*f^2 + 720*A^2*a^6*b^8*d^12*f^2 + 552*A^2* a^4*b^10*d^12*f^2 + 471*A^2*a^8*b^6*d^12*f^2 + 216*A^2*a^2*b^12*d^12*f^2 + 93*A^2*a^10*b^4*d^12*f^2 + 33*B^2*a^2*b^12*c^12*f^2 + 33*A^2*a^12*b^2*d^12*f^2 - 27*B^2*a^4*b^10*c^12*f^2 + 3*B^2*a^6*b^8*c^12*f^2 + 33*A^2*a^4*b^10*c^12*f^2 - 27*A^2*a^2*b^12*c^12*f^2 - A^2*a^6*b^8*c^12*f^2 + 3*C^2*b^14*c^12*f^2 - C^2*a^14*d^12*f^2 + 36*A^2*b^14*d^12*f^2 + 3*B^2*a^14*d^12*f^2 - B^2*b^14*c^12*f^2 + 3*A^2*b^14*c^12*f^2 - A^2*a^14*d^12*f^2 - 44*A*B*C*a^10*b*c*d^9*f + 3816*A*B*C*a^4*b^7*c^5*d^5*f + 2920*A*B*C*a^5*b^6*c^2*d^8*f - 2736*A*B*C*a^6*b^5*c^3*d^7*f - 2672*A*B*C*a^3*b^8*c^4*d^6*f + 1996*A*B*C*a^7*b^4*c^4*d^6*f - 1412*A*B*C*a^5*b^6*c^6*d^4*f + 1120*A*B*C*a^2*b^9*c^3*d^7*f + 1080*A*B*C*a^7*b^4*c^2*d^8*f + 1040*A*B*C*a^2*b^9*c^5*d^5*f + 684*A*B*C*a^5*b^6*c^4*d^6*f + 592*A*B*C*a^4*b^7*c^3*d^7*f - 560*A*B*C*a^2*b^9*c^7*d^3*f - 448*A*B*C*a^3*b^8*c^2*d^8*f - 400*A*B*C*a^8*b^3*c^5*d^5*f - 398*A*B*C*a^9*b^2*c^2*d^8*f - 312*A*B*C*a^3*b^8*c^6*d^4*f + 166*A*B*C*a^3*b^8*c^8*d^2*f + 136*A*B*C*a^6*b^5*c^5*d^5*f + 128*A*B*C*a^6*b^5*c^7*d^3*f - 100*A*B*C*a^7*b^4*c^6*d^4*f - 64*A*B*C*a^9*b^2*c^4*d^6*f + 64*A*B*C*a^4*b^7*c^7*d^3*f - 32*A*B*C*a^8*b^3*c^3*d^7*f - 16*A*B*C*a^5*b^6*c^8*d^2*f - 1312*A*B*C*a^4*b^7*c*d^9*f + 996*A*B*C*a^8*b^3*c*d^9*f + 728*A*B*C*a*b^10*c^6*d^4*f - 624*A*B*C*a^6*b^5*c*d^9*f - 584*A*B*C*a*b^10*c^2*d^8*f - 512*A*B*C*a*b^10*c^4*d^6*f - 320*A*B*C*a^2*b^9*c*d^9*f - 98*A*B*C*a*b^10*c^8*d^2*f + 36*A*B*C*a^2*b^9*c^9*d*f + 32*A*B*C*a^10*b*c^3*d^7*f - 16*A*B*C*a^4*b^7*c^9*d*f + 46*B*C^2*a^10*b*c*d^9*f - 16*B^2*C*a*b^10*c*d^9*f - 2*B^2*C*a*b^10*c^9*d*f + 312*A^2*C*a*b^10*c*d^9*f - 48*A*C^2*a*b^10*c*d^9*f - 6*A^2*C*a*b^10*c^9*d*f + 6*A*C^2*a*b^10*c^9*d*f + 208*A*B^2*a*b^10*c*d^9*f - 2*A^2*B*a^10*b*c*d^9*f + 2*A*B^2*a*b^10*c^9*d*f - 224*A*B*C*b^11*c^5*d^5*f + 80*A*B*C*b^11*c^7*d^3*f - 32*A*B*C*b^11*c^3*d^7*f + 2*A*B*C*a^11*c^2*d^8*f - 480*A*B*C*a^7*b^4*d^10*f + 78*A*B*C*a^9*b^2*d^10*f - 64*A*B*C*a^5*b^6*d^10*f + 2*A*B*C*a^3*b^8*c^10*f - 1692*B*C^2*a^4*b^7*c^5*d^5*f - 1500*B^2*C*a^5*b^6*c^5*d^5*f - 1464*B^2*C*a^5*b^6*c^3*d^7*f + 1426*B*C^2*a^5*b^6*c^6*d^4*f - 1158*B^2*C*a^4*b^7*c^6*d^4*f + 1152*B*C^2*a^6*b^5*c^3*d^7*f + 1026*B^2*C*a^6*b^5*c^4*d^6*f - 974*B*C^2*a^7*b^4*c^4*d^6*f + 960*B^2*C*a^3*b^8*c^5*d^5*f - 884*B*C^2*a^5*b^6*c^2*d^8*f - 764*B^2*C*a^7*b^4*c^5*d^5*f + 752*B^2*C*a^4*b^7*c^2*d^8*f - 752*B*C^2*a^4*b^7*c^3*d^7*f + 738*B^2*C*a^4*b^7*c^4*d^6*f - 688*B^2*C*a^2*b^9*c^6*d^4*f - 675*B^2*C*a^8*b^3*c^2*d^8*f + 560*B*C^2*a^8*b^3*c^5*d^5*f + 496*B*C^2*a^3*b^8*c^4*d^6*f + 496*B*C^2*a^2*b^9*c^7*d^3*f - 468*B*C^2*a^7*b^4*c^2*d^8*f + 456*B^2*C*a^3*b^8*c^7*d^3*f - 452*B^2*C*a^8*b^3*c^4*d^6*f - 416*B*C^2*a^2*b^9*c^3*d^7*f + 378*B*C^2*a^5*b^6*c^4*d^6*f + 376*B*C^2*a^8*b^3*c^3*d^7*f - 360*B^2*C*a^6*b^5*c^2*d^8*f + 355*B*C^2*a^9*b^2*c^2*d^8*f + 346*B^2*C*a^6*b^5*c^6*d^4*f - 320*B^2*C*a^2*b^9*c^4*d^6*f + 268*B^2*C*a^2*b^9*c^2*d^8*f + 216*B^2*C*a^7*b^4*c^3*d^7*f - 203*B*C^2*a^3*b^8*c^8*d^2*f - 184*B*C^2*a^6*b^5*c^7*d^3*f + 170*B*C^2*a^7*b^4*c^6*d^4*f + 160*B^2*C*a^5*b^6*c^7*d^3*f - 160*B*C^2*a^2*b^9*c^5*d^5*f - 140*B^2*C*a^4*b^7*c^8*d^2*f - 136*B*C^2*a^3*b^8*c^2*d^8*f + 112*B^2*C*a^9*b^2*c^3*d^7*f + 91*B^2*C*a^2*b^9*c^8*d^2*f + 88*B*C^2*a^4*b^7*c^7*d^3*f + 72*B^2*C*a^8*b^3*c^6*d^4*f - 64*B^2*C*a^3*b^8*c^3*d^7*f - 60*B*C^2*a^3*b^8*c^6*d^4*f + 56*B*C^2*a^9*b^2*c^4*d^6*f + 52*B*C^2*a^6*b^5*c^5*d^5*f + 48*B^2*C*a^9*b^2*c^5*d^5*f - 48*B^2

*C*a^7*b^4*c^7*d^3*f + 44*B*C^2*a^5*b^6*c^8*d^2*f - 36*B*C^2*a^9*b^2*c^6*d^
4*f + 12*B^2*C*a^6*b^5*c^8*d^2*f - 2958*A^2*C*a^4*b^7*c^4*d^6*f - 1932*A^2*
C*a^4*b^7*c^2*d^8*f + 1848*A^2*C*a^5*b^6*c^3*d^7*f + 1728*A^2*C*a^3*b^8*c^3
*d^7*f + 1524*A^2*C*a^5*b^6*c^5*d^5*f + 1374*A*C^2*a^4*b^7*c^4*d^6*f - 1272
*A*C^2*a^5*b^6*c^3*d^7*f - 1236*A*C^2*a^5*b^6*c^5*d^5*f + 1116*A*C^2*a^4*b^
7*c^2*d^8*f - 1110*A^2*C*a^6*b^5*c^4*d^6*f + 1038*A*C^2*a^6*b^5*c^4*d^6*f -
768*A^2*C*a^2*b^9*c^2*d^8*f - 696*A^2*C*a^7*b^4*c^3*d^7*f - 666*A*C^2*a^4*
b^7*c^6*d^4*f + 564*A^2*C*a^6*b^5*c^2*d^8*f - 564*A*C^2*a^7*b^4*c^5*d^5*f -
555*A*C^2*a^8*b^3*c^2*d^8*f + 519*A^2*C*a^8*b^3*c^2*d^8*f - 480*A*C^2*a^3*
b^8*c^3*d^7*f + 456*A*C^2*a^3*b^8*c^5*d^5*f - 420*A*C^2*a^2*b^9*c^6*d^4*f +
408*A*C^2*a^7*b^4*c^3*d^7*f + 408*A*C^2*a^2*b^9*c^2*d^8*f + 348*A^2*C*a^2*
b^9*c^6*d^4*f - 348*A*C^2*a^6*b^5*c^2*d^8*f + 342*A*C^2*a^6*b^5*c^6*d^4*f -
336*A*C^2*a^8*b^3*c^4*d^6*f + 324*A^2*C*a^7*b^4*c^5*d^5*f - 312*A^2*C*a^2*
b^9*c^4*d^6*f + 264*A^2*C*a^8*b^3*c^4*d^6*f + 240*A*C^2*a^5*b^6*c^7*d^3*f +
195*A*C^2*a^2*b^9*c^8*d^2*f - 174*A^2*C*a^6*b^5*c^6*d^4*f + 144*A*C^2*a^9*
b^2*c^3*d^7*f - 123*A^2*C*a^2*b^9*c^8*d^2*f + 120*A*C^2*a^3*b^8*c^7*d^3*f +
108*A*C^2*a^8*b^3*c^6*d^4*f - 102*A^2*C*a^4*b^7*c^6*d^4*f - 96*A^2*C*a^4*b
^7*c^8*d^2*f + 72*A^2*C*a^3*b^8*c^7*d^3*f + 72*A*C^2*a^9*b^2*c^5*d^5*f - 48
*A^2*C*a^9*b^2*c^3*d^7*f + 48*A^2*C*a^5*b^6*c^7*d^3*f - 48*A*C^2*a^2*b^9*c^
4*d^6*f - 24*A^2*C*a^3*b^8*c^5*d^5*f - 12*A*C^2*a^4*b^7*c^8*d^2*f + 2736*A^
2*B*a^6*b^5*c^3*d^7*f + 2464*A^2*B*a^3*b^8*c^4*d^6*f - 2298*A*B^2*a^4*b^7*c
^4*d^6*f - 2252*A^2*B*a^5*b^6*c^2*d^8*f - 1692*A^2*B*a^4*b^7*c^5*d^5*f - 15
92*A*B^2*a^4*b^7*c^2*d^8*f - 1338*A*B^2*a^6*b^5*c^4*d^6*f + 1320*A*B^2*a^5*
b^6*c^3*d^7*f + 1212*A*B^2*a^5*b^6*c^5*d^5*f - 1056*A*B^2*a^3*b^8*c^5*d^5*f
+ 1024*A^2*B*a^4*b^7*c^3*d^7*f - 1022*A^2*B*a^7*b^4*c^4*d^6*f - 880*A^2*B*
a^2*b^9*c^5*d^5*f - 846*A^2*B*a^5*b^6*c^4*d^6*f - 840*A*B^2*a^7*b^4*c^3*d^7
*f + 760*A*B^2*a^2*b^9*c^6*d^4*f - 704*A^2*B*a^2*b^9*c^3*d^7*f + 688*A*B^2*
a^3*b^8*c^3*d^7*f + 660*A^2*B*a^3*b^8*c^6*d^4*f - 612*A^2*B*a^7*b^4*c^2*d^8
*f + 462*A*B^2*a^4*b^7*c^6*d^4*f + 459*A*B^2*a^8*b^3*c^2*d^8*f - 412*A*B^2*
a^2*b^9*c^2*d^8*f - 408*A*B^2*a^3*b^8*c^7*d^3*f + 388*A^2*B*a^6*b^5*c^5*d^5
*f + 296*A^2*B*a^3*b^8*c^2*d^8*f + 288*A*B^2*a^6*b^5*c^2*d^8*f + 284*A*B^2*
a^7*b^4*c^5*d^5*f + 236*A*B^2*a^8*b^3*c^4*d^6*f - 226*A*B^2*a^6*b^5*c^6*d^4
*f + 212*A*B^2*a^2*b^9*c^4*d^6*f + 202*A^2*B*a^5*b^6*c^6*d^4*f - 152*A^2*B*
a^4*b^7*c^7*d^3*f + 88*A^2*B*a^8*b^3*c^3*d^7*f + 79*A^2*B*a^9*b^2*c^2*d^8*f
- 70*A^2*B*a^7*b^4*c^6*d^4*f + 68*A*B^2*a^4*b^7*c^8*d^2*f + 64*A^2*B*a^2*b
^9*c^7*d^3*f - 64*A*B^2*a^9*b^2*c^3*d^7*f + 56*A^2*B*a^8*b^3*c^5*d^5*f + 56
*A^2*B*a^6*b^5*c^7*d^3*f + 37*A^2*B*a^3*b^8*c^8*d^2*f - 28*A^2*B*a^9*b^2*c^
4*d^6*f - 28*A^2*B*a^5*b^6*c^8*d^2*f + 17*A*B^2*a^2*b^9*c^8*d^2*f - 16*A*B^
2*a^5*b^6*c^7*d^3*f + 48*A*B*C*b^11*c*d^9*f + 4*A*B*C*b^11*c^9*d*f + 24*A*B
*C*a*b^10*d^10*f - 6*A*B*C*a*b^10*c^10*f + 432*B^2*C*a^7*b^4*c*d^9*f - 376*
B*C^2*a*b^10*c^6*d^4*f - 354*B*C^2*a^8*b^3*c*d^9*f + 352*B^2*C*a*b^10*c^5*d
^5*f + 320*B^2*C*a^5*b^6*c*d^9*f + 256*B^2*C*a*b^10*c^3*d^7*f - 232*B^2*C*a
*b^10*c^7*d^3*f - 210*B^2*C*a^9*b^2*c*d^9*f - 152*B*C^2*a*b^10*c^4*d^6*f +
85*B*C^2*a*b^10*c^8*d^2*f + 72*B^2*C*a^3*b^8*c*d^9*f - 48*B*C^2*a^6*b^5*c*d
^9*f - 40*B*C^2*a^10*b*c^3*d^7*f + 40*B*C^2*a*b^10*c^2*d^8*f + 37*B^2*C*a^1

0*b*c^2*d^8*f + 22*B^2*C*a^3*b^8*c^9*d*f - 18*B*C^2*a^2*b^9*c^9*d*f + 16*B*
C^2*a^2*b^9*c*d^9*f - 12*B^2*C*a^10*b*c^4*d^6*f + 8*B*C^2*a^4*b^7*c^9*d*f +
8*B*C^2*a^4*b^7*c*d^9*f - 984*A^2*C*a^7*b^4*c*d^9*f + 672*A^2*C*a^3*b^8*c*
d^9*f + 552*A*C^2*a^7*b^4*c*d^9*f - 504*A^2*C*a*b^10*c^5*d^5*f - 408*A^2*C*
a^5*b^6*c*d^9*f + 408*A*C^2*a^5*b^6*c*d^9*f + 336*A*C^2*a*b^10*c^5*d^5*f -
216*A*C^2*a*b^10*c^7*d^3*f + 192*A*C^2*a*b^10*c^3*d^7*f - 162*A*C^2*a^9*b^2
*c*d^9*f + 120*A^2*C*a*b^10*c^7*d^3*f + 96*A^2*C*a*b^10*c^3*d^7*f + 90*A^2*
C*a^9*b^2*c*d^9*f + 66*A^2*C*a^3*b^8*c^9*d*f - 66*A*C^2*a^3*b^8*c^9*d*f + 5
7*A*C^2*a^10*b*c^2*d^8*f - 48*A*C^2*a^3*b^8*c*d^9*f - 9*A^2*C*a^10*b*c^2*d^
8*f + 1736*A^2*B*a^4*b^7*c*d^9*f + 1248*A^2*B*a^6*b^5*c*d^9*f - 1008*A*B^2*
a^7*b^4*c*d^9*f + 772*A^2*B*a*b^10*c^4*d^6*f - 688*A*B^2*a*b^10*c^5*d^5*f -
608*A*B^2*a^5*b^6*c*d^9*f + 436*A^2*B*a*b^10*c^2*d^8*f - 426*A^2*B*a^8*b^3
*c*d^9*f + 312*A*B^2*a^3*b^8*c*d^9*f + 304*A^2*B*a^2*b^9*c*d^9*f - 244*A^2*
B*a*b^10*c^6*d^4*f - 160*A*B^2*a*b^10*c^3*d^7*f + 114*A*B^2*a^9*b^2*c*d^9*f
+ 88*A*B^2*a*b^10*c^7*d^3*f - 22*A*B^2*a^3*b^8*c^9*d*f - 18*A^2*B*a^2*b^9*
c^9*d*f + 13*A^2*B*a*b^10*c^8*d^2*f - 13*A*B^2*a^10*b*c^2*d^8*f + 8*A^2*B*a
^10*b*c^3*d^7*f + 8*A^2*B*a^4*b^7*c^9*d*f + 112*B^2*C*b^11*c^6*d^4*f - 64*B
*C^2*b^11*c^7*d^3*f + 16*B^2*C*b^11*c^4*d^6*f - 16*B^2*C*b^11*c^2*d^8*f + 1
6*B*C^2*b^11*c^5*d^5*f + 16*B*C^2*b^11*c^3*d^7*f - B^2*C*b^11*c^8*d^2*f + 9
6*A^2*C*b^11*c^4*d^6*f - 84*A^2*C*b^11*c^6*d^4*f + 72*A*C^2*b^11*c^6*d^4*f
- 24*A*C^2*b^11*c^4*d^6*f - 24*A*C^2*b^11*c^2*d^8*f - 21*A*C^2*b^11*c^8*d^2
*f + 12*A^2*C*b^11*c^2*d^8*f + 9*A^2*C*b^11*c^8*d^2*f - B*C^2*a^11*c^2*d^8*
f + 176*A*B^2*b^11*c^4*d^6*f + 136*A^2*B*b^11*c^5*d^5*f - 128*A^2*B*b^11*c^
3*d^7*f + 112*A*B^2*b^11*c^2*d^8*f + 111*B^2*C*a^8*b^3*d^10*f - 64*A*B^2*b^
11*c^6*d^4*f - 39*B*C^2*a^9*b^2*d^10*f + 24*B*C^2*a^7*b^4*d^10*f - 16*A^2*B
*b^11*c^7*d^3*f - 4*B^2*C*a^2*b^9*d^10*f - 4*B*C^2*a^5*b^6*d^10*f + 432*A^2
*C*a^6*b^5*d^10*f + 192*A^2*C*a^4*b^7*d^10*f - 111*A^2*C*a^8*b^3*d^10*f + 1
11*A*C^2*a^8*b^3*d^10*f - 72*A*C^2*a^6*b^5*d^10*f + 12*A*C^2*a^4*b^7*d^10*f
- 3*B^2*C*a^2*b^9*c^10*f - A^2*B*a^11*c^2*d^8*f - B*C^2*a^3*b^8*c^10*f + 4
56*A^2*B*a^7*b^4*d^10*f - 288*A^2*B*a^3*b^8*d^10*f + 252*A*B^2*a^6*b^5*d^10
*f + 192*A*B^2*a^4*b^7*d^10*f - 183*A*B^2*a^8*b^3*d^10*f - 148*A^2*B*a^5*b^
6*d^10*f + 76*A*B^2*a^2*b^9*d^10*f - 9*A^2*C*a^2*b^9*c^10*f + 9*A*C^2*a^2*b
^9*c^10*f - 3*A^2*B*a^9*b^2*d^10*f + 3*A*B^2*a^2*b^9*c^10*f - A^2*B*a^3*b^8
*c^10*f - 2*C^3*a*b^10*c^9*d*f - 2*B^3*a^10*b*c*d^9*f - 264*A^3*a*b^10*c*d^
9*f + 2*A^3*a*b^10*c^9*d*f - 2*B*C^2*b^11*c^9*d*f - 2*B^2*C*a^11*c*d^9*f -
120*A^2*B*b^11*c*d^9*f - 9*B^2*C*a^10*b*d^10*f - 6*A^2*C*a^11*c*d^9*f + 6*A
*C^2*a^11*c*d^9*f - 2*A^2*B*b^11*c^9*d*f + 9*A^2*C*a^10*b*d^10*f - 9*A*C^2*
a^10*b*d^10*f + 3*B*C^2*a*b^10*c^10*f + 2*A*B^2*a^11*c*d^9*f - 132*A^2*B*a*
b^10*d^10*f - 3*A*B^2*a^10*b*d^10*f + 3*A^2*B*a*b^10*c^10*f + 520*C^3*a^5*b
^6*c^3*d^7*f + 460*C^3*a^5*b^6*c^5*d^5*f - 418*C^3*a^6*b^5*c^4*d^6*f + 406*
C^3*a^4*b^7*c^6*d^4*f + 268*C^3*a^7*b^4*c^5*d^5*f - 266*C^3*a^6*b^5*c^6*d^4
*f + 233*C^3*a^8*b^3*c^2*d^8*f - 176*C^3*a^5*b^6*c^7*d^3*f + 164*C^3*a^2*b^
9*c^6*d^4*f + 140*C^3*a^6*b^5*c^2*d^8*f + 136*C^3*a^2*b^9*c^4*d^6*f - 128*C
^3*a^9*b^2*c^3*d^7*f + 128*C^3*a^3*b^8*c^3*d^7*f - 108*C^3*a^8*b^3*c^6*d^4*
f - 104*C^3*a^3*b^8*c^7*d^3*f - 104*C^3*a^3*b^8*c^5*d^5*f + 100*C^3*a^8*b^3

*c^4*d^6*f - 89*C^3*a^2*b^9*c^8*d^2*f - 72*C^3*a^9*b^2*c^5*d^5*f - 40*C^3*a
^7*b^4*c^3*d^7*f + 40*C^3*a^4*b^7*c^8*d^2*f - 28*C^3*a^4*b^7*c^2*d^8*f - 16
*C^3*a^2*b^9*c^2*d^8*f - 2*C^3*a^4*b^7*c^4*d^6*f + 828*B^3*a^4*b^7*c^5*d^5*
f + 408*B^3*a^5*b^6*c^2*d^8*f + 390*B^3*a^7*b^4*c^4*d^6*f - 372*B^3*a^3*b^8
*c^4*d^6*f - 336*B^3*a^6*b^5*c^3*d^7*f - 314*B^3*a^5*b^6*c^6*d^4*f + 288*B^
3*a^4*b^7*c^3*d^7*f + 216*B^3*a^7*b^4*c^2*d^8*f - 176*B^3*a^2*b^9*c^7*d^3*f
 + 128*B^3*a^2*b^9*c^3*d^7*f + 108*B^3*a^6*b^5*c^5*d^5*f + 88*B^3*a^4*b^7*c
^7*d^3*f + 72*B^3*a^2*b^9*c^5*d^5*f - 68*B^3*a^3*b^8*c^2*d^8*f - 65*B^3*a^9
*b^2*c^2*d^8*f - 56*B^3*a^8*b^3*c^5*d^5*f + 40*B^3*a^6*b^5*c^7*d^3*f + 37*B
^3*a^3*b^8*c^8*d^2*f + 30*B^3*a^5*b^6*c^4*d^6*f - 28*B^3*a^5*b^6*c^8*d^2*f
+ 24*B^3*a^8*b^3*c^3*d^7*f - 4*B^3*a^9*b^2*c^4*d^6*f - 2*B^3*a^7*b^4*c^6*d^
4*f + 1586*A^3*a^4*b^7*c^4*d^6*f - 1376*A^3*a^3*b^8*c^3*d^7*f - 1096*A^3*a^
5*b^6*c^3*d^7*f + 844*A^3*a^4*b^7*c^2*d^8*f - 748*A^3*a^5*b^6*c^5*d^5*f + 4
90*A^3*a^6*b^5*c^4*d^6*f + 376*A^3*a^2*b^9*c^2*d^8*f + 362*A^3*a^4*b^7*c^6*
d^4*f - 356*A^3*a^6*b^5*c^2*d^8*f + 328*A^3*a^7*b^4*c^3*d^7*f - 328*A^3*a^3
*b^8*c^5*d^5*f + 224*A^3*a^2*b^9*c^4*d^6*f - 197*A^3*a^8*b^3*c^2*d^8*f - 11
2*A^3*a^5*b^6*c^7*d^3*f + 98*A^3*a^6*b^5*c^6*d^4*f - 92*A^3*a^2*b^9*c^6*d^4
*f - 88*A^3*a^3*b^8*c^7*d^3*f + 68*A^3*a^4*b^7*c^8*d^2*f + 32*A^3*a^9*b^2*c
^3*d^7*f - 28*A^3*a^8*b^3*c^4*d^6*f - 28*A^3*a^7*b^4*c^5*d^5*f + 17*A^3*a^2
*b^9*c^8*d^2*f + 104*C^3*a*b^10*c^7*d^3*f + 54*C^3*a^9*b^2*c*d^9*f - 40*C^3
*a^7*b^4*c*d^9*f - 35*C^3*a^10*b*c^2*d^8*f + 22*C^3*a^3*b^8*c^9*d*f + 16*C^
3*a*b^10*c^5*d^5*f - 16*C^3*a*b^10*c^3*d^7*f + 8*C^3*a^5*b^6*c*d^9*f - 2*A*
B*C*a^11*d^10*f + 198*B^3*a^8*b^3*c*d^9*f + 192*B^3*a*b^10*c^6*d^4*f - 128*
B^3*a^4*b^7*c*d^9*f - 80*B^3*a*b^10*c^2*d^8*f - 56*B^3*a^2*b^9*c*d^9*f - 24
*B^3*a^6*b^5*c*d^9*f - 18*B^3*a^2*b^9*c^9*d*f - 16*B^3*a*b^10*c^4*d^6*f + 1
3*B^3*a*b^10*c^8*d^2*f + 8*B^3*a^10*b*c^3*d^7*f + 8*B^3*a^4*b^7*c^9*d*f - 6
24*A^3*a^3*b^8*c*d^9*f + 472*A^3*a^7*b^4*c*d^9*f - 272*A^3*a*b^10*c^3*d^7*f
 + 152*A^3*a*b^10*c^5*d^5*f - 22*A^3*a^3*b^8*c^9*d*f + 18*A^3*a^9*b^2*c*d^9
*f - 13*A^3*a^10*b*c^2*d^8*f - 8*A^3*a^5*b^6*c*d^9*f - 8*A^3*a*b^10*c^7*d^3
*f + A*B^2*b^11*c^8*d^2*f + 11*C^3*b^11*c^8*d^2*f - 8*C^3*b^11*c^6*d^4*f -
4*C^3*b^11*c^4*d^6*f - 64*B^3*b^11*c^5*d^5*f - 32*B^3*b^11*c^3*d^7*f - 68*A
^3*b^11*c^4*d^6*f + 20*A^3*b^11*c^6*d^4*f + 12*A^3*b^11*c^2*d^8*f - C^3*a^8
*b^3*d^10*f - B^3*a^11*c^2*d^8*f - 60*B^3*a^7*b^4*d^10*f - 32*B^3*a^5*b^6*d
^10*f + 21*B^3*a^9*b^2*d^10*f - 12*B^3*a^3*b^8*d^10*f - 3*C^3*a^2*b^9*c^10*
f - 360*A^3*a^6*b^5*d^10*f - 204*A^3*a^4*b^7*d^10*f - B^3*a^3*b^8*c^10*f +
3*A^3*a^2*b^9*c^10*f - 2*C^3*a^11*c*d^9*f - 2*B^3*b^11*c^9*d*f + 3*C^3*a^10
*b*d^10*f + 2*A^3*a^11*c*d^9*f + 3*B^3*a*b^10*c^10*f - 3*A^3*a^10*b*d^10*f
- 36*A^2*C*b^11*d^10*f + 3*A^2*C*b^11*c^10*f - 3*A*C^2*b^11*c^10*f - A*B^2*
b^11*c^10*f + 36*A^3*b^11*d^10*f - A^3*b^11*c^10*f + A^3*b^11*c^8*d^2*f + A
^3*a^8*b^3*d^10*f + B^2*C*b^11*c^10*f + B*C^2*a^11*d^10*f + A^2*B*a^11*d^10
*f + C^3*b^11*c^10*f + B^3*a^11*d^10*f - 6*A*B^2*C*a^7*b*c*d^7 + 4*A*B^2*C*
a*b^7*c*d^7 + 168*A^2*B*C*a^2*b^6*c^3*d^5 + 144*A*B*C^2*a^3*b^5*c^4*d^4 - 1
29*A^2*B*C*a^3*b^5*c^4*d^4 - 96*A*B*C^2*a^2*b^6*c^3*d^5 + 84*A*B*C^2*a^3*b^
5*c^2*d^6 + 72*A^2*B*C*a^4*b^4*c^3*d^5 - 72*A^2*B*C*a^3*b^5*c^2*d^6 + 64*A*
B^2*C*a^4*b^4*c^4*d^4 - 60*A*B*C^2*a^4*b^4*c^3*d^5 + 57*A^2*B*C*a^5*b^3*c^2

*d^6 - 56*A*B^2*C*a^5*b^3*c^3*d^5 - 39*A*B^2*C*a^2*b^6*c^4*d^4 - 38*A*B^2*C
*a^3*b^5*c^5*d^3 + 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^5*b^3*c^4*d^4
- 30*A*B*C^2*a^5*b^3*c^2*d^6 + 27*A*B^2*C*a^6*b^2*c^2*d^6 - 24*A*B^2*C*a^2*
b^6*c^2*d^6 + 24*A*B*C^2*a^6*b^2*c^3*d^5 - 24*A*B*C^2*a^4*b^4*c^5*d^3 - 18*
A^2*B*C*a^5*b^3*c^4*d^4 + 18*A^2*B*C*a^2*b^6*c^5*d^3 - 15*A*B^2*C*a^4*b^4*c
^2*d^6 - 12*A^2*B*C*a^6*b^2*c^3*d^5 + 12*A^2*B*C*a^4*b^4*c^5*d^3 + 9*A*B^2*
C*a^2*b^6*c^6*d^2 + 6*A*B*C^2*a^3*b^5*c^6*d^2 - 3*A^2*B*C*a^3*b^5*c^6*d^2 +
60*A^2*B*C*a^2*b^6*c*d^7 - 51*A^2*B*C*a*b^7*c^4*d^4 + 48*A*B*C^2*a^6*b^2*c
*d^7 - 42*A^2*B*C*a^6*b^2*c*d^7 - 42*A^2*B*C*a*b^7*c^2*d^6 + 36*A*B*C^2*a^4
*b^4*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4 + 36*A*B*C^2*a*b^7*c^2*d^6 - 30*A^2*B
*C*a^4*b^4*c*d^7 + 24*A*B^2*C*a^3*b^5*c*d^7 - 24*A*B*C^2*a^2*b^6*c*d^7 + 18
*A*B^2*C*a*b^7*c^5*d^3 - 18*A*B*C^2*a*b^7*c^6*d^2 + 12*A*B^2*C*a*b^7*c^3*d^
5 + 9*A^2*B*C*a*b^7*c^6*d^2 + 6*A*B^2*C*a^5*b^3*c*d^7 - 6*A*B*C^2*a^7*b*c^2
*d^6 + 3*A^2*B*C*a^7*b*c^2*d^6 - 18*B^3*C*a^6*b^2*c*d^7 - 18*B*C^3*a^6*b^2*
c*d^7 - 14*B^3*C*a^4*b^4*c*d^7 - 14*B*C^3*a^4*b^4*c*d^7 - 10*B^3*C*a*b^7*c^
2*d^6 - 10*B*C^3*a*b^7*c^2*d^6 + 9*B^3*C*a*b^7*c^6*d^2 + 9*B*C^3*a*b^7*c^6*
d^2 - 7*B^3*C*a*b^7*c^4*d^4 - 7*B*C^3*a*b^7*c^4*d^4 + 6*B^2*C^2*a^7*b*c*d^7
- 4*B^3*C*a^2*b^6*c*d^7 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a^2*b^6*c*d^7 +
3*B^3*C*a^7*b*c^2*d^6 + 3*B*C^3*a^7*b*c^2*d^6 + 144*A^3*C*a^3*b^5*c*d^7 + 6
2*A^3*C*a^5*b^3*c*d^7 + 48*A*C^3*a^3*b^5*c*d^7 - 36*A^2*C^2*a*b^7*c*d^7 + 2
6*A*C^3*a^5*b^3*c*d^7 + 20*A^3*C*a*b^7*c^3*d^5 + 18*A^2*C^2*a^7*b*c*d^7 - 1
8*A*C^3*a*b^7*c^5*d^3 - 6*A^3*C*a*b^7*c^5*d^3 - 4*A*C^3*a*b^7*c^3*d^5 - 32*
A^3*B*a^2*b^6*c*d^7 - 32*A*B^3*a^2*b^6*c*d^7 + 22*A^3*B*a*b^7*c^4*d^4 + 22*
A*B^3*a*b^7*c^4*d^4 + 16*A^3*B*a*b^7*c^2*d^6 + 16*A*B^3*a*b^7*c^2*d^6 + 12*
A^3*B*a^6*b^2*c*d^7 + 12*A*B^3*a^6*b^2*c*d^7 + 8*A^3*B*a^4*b^4*c*d^7 - 8*A^
2*B^2*a*b^7*c*d^7 + 8*A*B^3*a^4*b^4*c*d^7 + 36*A^2*B*C*b^8*c^3*d^5 + 24*A*B
*C^2*b^8*c^5*d^3 - 18*A^2*B*C*b^8*c^5*d^3 - 12*A*B*C^2*b^8*c^3*d^5 - 3*A*B^
2*C*b^8*c^6*d^2 - 3*A*B^2*C*b^8*c^4*d^4 - 2*A*B^2*C*b^8*c^2*d^6 + 57*A^2*B*
C*a^5*b^3*d^8 + 36*A^2*B*C*a^3*b^5*d^8 - 30*A*B*C^2*a^5*b^3*d^8 - 18*A*B*C^
2*a^3*b^5*d^8 - 9*A*B^2*C*a^4*b^4*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 2*A*B^2*C*a
^2*b^6*d^8 + 34*B^2*C^2*a^3*b^5*c^5*d^3 + 28*B^2*C^2*a^5*b^3*c^3*d^5 + 24*B
^2*C^2*a^2*b^6*c^4*d^4 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^
3*d^5 + 12*B^2*C^2*a^2*b^6*c^2*d^6 + 9*B^2*C^2*a^6*b^2*c^4*d^4 + 9*B^2*C^2*
a^4*b^4*c^2*d^6 - 9*B^2*C^2*a^2*b^6*c^6*d^2 - 3*B^2*C^2*a^6*b^2*c^2*d^6 + 1
59*A^2*C^2*a^4*b^4*c^2*d^6 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^3*b
^5*c^5*d^3 + 78*A^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A
^2*C^2*a^6*b^2*c^2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^2*b^6*c^
4*d^4 + 9*A^2*C^2*a^6*b^2*c^4*d^4 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2
*a^4*b^4*c^2*d^6 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^2*b^6*c^4*d^4
+ 28*A^2*B^2*a^5*b^3*c^3*d^5 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^6*b
^2*c^2*d^6 + 4*A^2*B^2*a^3*b^5*c^5*d^3 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a^
7*b*c*d^7 + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a^7*b*c*d^7 + 24*A^2*B*C*b^8*c*d
^7 - 12*A*B*C^2*b^8*c*d^7 + 12*A^2*B*C*a*b^7*d^8 + 6*A*B*C^2*a^7*b*d^8 - 6*
A*B*C^2*a*b^7*d^8 - 3*A^2*B*C*a^7*b*d^8 - 53*B^3*C*a^3*b^5*c^4*d^4 - 53*B*C
^3*a^3*b^5*c^4*d^4 - 32*B^3*C*a^3*b^5*c^2*d^6 - 32*B*C^3*a^3*b^5*c^2*d^6 -

18*B^3*C*a^5*b^3*c^4*d^4 - 18*B*C^3*a^5*b^3*c^4*d^4 + 16*B^3*C*a^4*b^4*c^3*
d^5 + 16*B*C^3*a^4*b^4*c^3*d^5 - 12*B^3*C*a^6*b^2*c^3*d^5 + 12*B^3*C*a^4*b^
4*c^5*d^3 + 12*B^2*C^2*a^3*b^5*c*d^7 - 12*B*C^3*a^6*b^2*c^3*d^5 + 12*B*C^3*
a^4*b^4*c^5*d^3 + 8*B^3*C*a^2*b^6*c^3*d^5 + 8*B*C^3*a^2*b^6*c^3*d^5 - 6*B^3
*C*a^2*b^6*c^5*d^3 + 6*B^2*C^2*a^5*b^3*c*d^7 - 6*B^2*C^2*a*b^7*c^5*d^3 - 6*
B*C^3*a^2*b^6*c^5*d^3 - 3*B^3*C*a^3*b^5*c^6*d^2 - 3*B*C^3*a^3*b^5*c^6*d^2 -
175*A^3*C*a^4*b^4*c^2*d^6 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a^3*b^
5*c*d^7 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^3*b^5*c^5*d^3 - 73*A*C^3*a
^4*b^4*c^2*d^6 - 66*A^2*C^2*a^5*b^3*c*d^7 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A
*C^3*a^4*b^4*c^4*d^4 + 30*A^3*C*a^4*b^4*c^4*d^4 - 30*A^3*C*a^3*b^5*c^5*d^3
+ 27*A*C^3*a^2*b^6*c^6*d^2 + 21*A*C^3*a^2*b^6*c^4*d^4 + 18*A^2*C^2*a*b^7*c^
5*d^3 - 18*A*C^3*a^6*b^2*c^4*d^4 - 16*A*C^3*a^2*b^6*c^2*d^6 + 15*A^3*C*a^6*
b^2*c^2*d^6 - 15*A^3*C*a^2*b^6*c^4*d^4 - 12*A^2*C^2*a*b^7*c^3*d^5 + 9*A^3*C
*a^2*b^6*c^6*d^2 + 9*A*C^3*a^6*b^2*c^2*d^6 - 80*A^3*B*a^2*b^6*c^3*d^5 - 80*
A*B^3*a^2*b^6*c^3*d^5 + 38*A^3*B*a^3*b^5*c^4*d^4 + 38*A*B^3*a^3*b^5*c^4*d^4
- 36*A^2*B^2*a^3*b^5*c*d^7 - 28*A^3*B*a^5*b^3*c^2*d^6 - 28*A^3*B*a^4*b^4*c
^3*d^5 - 28*A*B^3*a^5*b^3*c^2*d^6 - 28*A*B^3*a^4*b^4*c^3*d^5 + 20*A^3*B*a^3
*b^5*c^2*d^6 + 20*A*B^3*a^3*b^5*c^2*d^6 - 12*A^3*B*a^2*b^6*c^5*d^3 - 12*A^2
*B^2*a^5*b^3*c*d^7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A^2*B^2*a*b^7*c^3*d^5 -
12*A*B^3*a^2*b^6*c^5*d^3 + 9*B^2*C^2*b^8*c^4*d^4 + 4*B^2*C^2*b^8*c^2*d^6 +
3*B^2*C^2*b^8*c^6*d^2 - 30*A^2*C^2*b^8*c^4*d^4 + 9*A^2*C^2*b^8*c^6*d^2 + 16
*A^2*B^2*b^8*c^2*d^6 + 6*B^2*C^2*a^6*b^2*d^8 + 3*B^2*C^2*a^4*b^4*d^8 + 3*A^
2*B^2*b^8*c^4*d^4 + 36*A^2*C^2*a^4*b^4*d^8 + 27*A^2*C^2*a^2*b^6*d^8 - 18*A^
2*C^2*a^6*b^2*d^8 + 33*A^2*B^2*a^4*b^4*d^8 + 28*A^2*B^2*a^2*b^6*d^8 + 6*A^2
*B^2*a^6*b^2*d^8 + 6*C^4*a*b^7*c^5*d^3 + 4*C^4*a*b^7*c^3*d^5 - 2*C^4*a^5*b^
3*c*d^7 + 12*B^4*a^3*b^5*c*d^7 - 12*B^4*a*b^7*c^5*d^3 + 8*B^4*a^5*b^3*c*d^7
- 4*B^4*a*b^7*c^3*d^5 - 48*A^4*a^3*b^5*c*d^7 - 20*A^4*a^5*b^3*c*d^7 - 8*A^
4*a*b^7*c^3*d^5 - 10*B^3*C*b^8*c^5*d^3 - 10*B*C^3*b^8*c^5*d^3 - 4*B^3*C*b^8
*c^3*d^5 - 4*B*C^3*b^8*c^3*d^5 + 23*A^3*C*b^8*c^4*d^4 - 18*A^3*C*b^8*c^2*d^
6 + 11*A*C^3*b^8*c^4*d^4 - 9*A*C^3*b^8*c^6*d^2 + 6*A*C^3*b^8*c^2*d^6 - 3*A^
3*C*b^8*c^6*d^2 - 20*A^3*B*b^8*c^3*d^5 - 20*A*B^3*b^8*c^3*d^5 + 4*A^3*B*b^8
*c^5*d^3 + 4*A*B^3*b^8*c^5*d^3 - 63*A^3*C*a^4*b^4*d^8 - 54*A^3*C*a^2*b^6*d^
8 + 9*A^3*C*a^6*b^2*d^8 + 9*A*C^3*a^6*b^2*d^8 - 3*A*C^3*a^4*b^4*d^8 - 28*A^
3*B*a^5*b^3*d^8 - 28*A*B^3*a^5*b^3*d^8 - 18*A^3*B*a^3*b^5*d^8 - 18*A*B^3*a^
3*b^5*d^8 + B^3*C*a^5*b^3*c^2*d^6 + B*C^3*a^5*b^3*c^2*d^6 + 6*C^4*a^7*b*c*d
^7 + 4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 12*A^3*B*b^8*c*d^7 - 12*A*B^3
*b^8*c*d^7 - 3*B^3*C*a^7*b*d^8 - 3*B*C^3*a^7*b*d^8 - 6*A^3*B*a*b^7*d^8 - 6*
A*B^3*a*b^7*d^8 + 30*C^4*a^3*b^5*c^5*d^3 + 19*C^4*a^4*b^4*c^2*d^6 + 9*C^4*a
^6*b^2*c^4*d^4 - 9*C^4*a^2*b^6*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*
b^6*c^2*d^6 + 3*C^4*a^6*b^2*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^2*b^6
*c^4*d^4 + 28*B^4*a^5*b^3*c^3*d^5 + 27*B^4*a^2*b^6*c^4*d^4 - 17*B^4*a^4*b^4
*c^4*d^4 - 10*B^4*a^4*b^4*c^2*d^6 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c
^2*d^6 - 6*B^4*a^6*b^2*c^2*d^6 + 4*B^4*a^3*b^5*c^5*d^3 + 70*A^4*a^4*b^4*c^2
*d^6 + 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^2*b^6*c^4
*d^4 + B^2*C^2*a^2*b^6*d^8 - 18*A^3*C*b^8*d^8 + B^3*C*a^5*b^3*d^8 + B*C^3*a

```
^5*b^3*d^8 + 3*C^4*b^8*c^6*d^2 + 8*B^4*b^8*c^4*d^4 + 4*B^4*b^8*c^2*d^6 + 12
*A^4*b^8*c^2*d^6 - 5*A^4*b^8*c^4*d^4 + 6*B^4*a^6*b^2*d^8 + 3*B^4*a^4*b^4*d^
8 + 30*A^4*a^4*b^4*d^8 + 27*A^4*a^2*b^6*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2
*b^8*d^8 + 9*A^4*b^8*d^8 + C^4*b^8*c^4*d^4 + B^4*a^2*b^6*d^8, f, k), k, 1,
4) - ((2*A*a^6*d^4 - A*b^6*c^4 - B*a*b^5*c^4 - 2*B*a^6*c*d^3 - 5*A*a^2*b^4*
c^4 + 2*A*a^2*b^4*d^4 + 4*A*a^4*b^2*d^4 + 3*B*a^3*b^3*c^4 + 3*C*a^2*b^4*c^4
 - C*a^4*b^2*c^4 - A*b^6*c^2*d^2 + 2*C*a^6*c^2*d^2 + 9*A*a^3*b^3*c*d^3 + 9*
A*a^3*b^3*c^3*d - B*a*b^5*c^2*d^2 - 5*B*a^2*b^4*c*d^3 - 3*B*a^2*b^4*c^3*d -
 11*B*a^4*b^2*c*d^3 - 7*B*a^4*b^2*c^3*d + C*a^3*b^3*c*d^3 + C*a^3*b^3*c^3*d
 - 5*A*a^2*b^4*c^2*d^2 + 3*B*a^3*b^3*c^2*d^2 + 5*C*a^2*b^4*c^2*d^2 + 3*C*a^
4*b^2*c^2*d^2 + 5*A*a*b^5*c*d^3 + 5*A*a*b^5*c^3*d + 5*C*a^5*b*c*d^3 + 5*C*a
^5*b*c^3*d)/(2*(a^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^4*c^2
 + a^4*d^2 + b^4*c^2 + b^4*d^2 + 2*a^2*b^2*c^2 + 2*a^2*b^2*d^2)) + (tan(e +
 f*x)*(9*A*a*b^5*d^4 - 4*A*a*b^5*c^4 - 2*B*b^6*c^4 + 4*A*a^5*b*d^4 + 4*C*a*
b^5*c^4 + 3*A*b^6*c*d^3 + 3*A*b^6*c^3*d + 5*C*a^5*b*d^4 + 17*A*a^3*b^3*d^4
+ 2*B*a^2*b^4*c^4 - 3*B*a^2*b^4*d^4 - 7*B*a^4*b^2*d^4 + C*a^3*b^3*d^4 - 2*B
*b^6*c^2*d^2 + A*a*b^5*c^2*d^2 + 3*A*a^2*b^4*c*d^3 + 3*A*a^2*b^4*c^3*d - 11
*B*a^3*b^3*c*d^3 - 3*B*a^3*b^3*c^3*d + 8*C*a*b^5*c^2*d^2 + 3*C*a^2*b^4*c*d^
3 + 3*C*a^2*b^4*c^3*d + 3*C*a^4*b^2*c*d^3 + 3*C*a^4*b^2*c^3*d + 9*C*a^5*b*c
^2*d^2 + 9*A*a^3*b^3*c^2*d^2 - B*a^2*b^4*c^2*d^2 - 7*B*a^4*b^2*c^2*d^2 + 9*
C*a^3*b^3*c^2*d^2 - 7*B*a*b^5*c*d^3 - 3*B*a*b^5*c^3*d - 4*B*a^5*b*c*d^3))/(
2*(a^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^4*c^2 + a^4*d^2 +
b^4*c^2 + b^4*d^2 + 2*a^2*b^2*c^2 + 2*a^2*b^2*d^2)) + (tan(e + f*x)^2*(3*A*
b^6*d^4 - B*a*b^5*d^4 - 2*B*b^6*c*d^3 - B*b^6*c^3*d + 6*A*a^2*b^4*d^4 + A*a
^4*b^2*d^4 - 3*B*a^3*b^3*d^4 + 2*A*b^6*c^2*d^2 + 2*C*a^4*b^2*d^4 + C*b^6*c^
2*d^2 - B*a*b^5*c^2*d^2 - B*a^2*b^4*c*d^3 + B*a^2*b^4*c^3*d - B*a^4*b^2*c*d
^3 + 4*A*a^2*b^4*c^2*d^2 - 3*B*a^3*b^3*c^2*d^2 + 2*C*a^2*b^4*c^2*d^2 + 3*C*
a^4*b^2*c^2*d^2 - 2*A*a*b^5*c*d^3 - 2*A*a*b^5*c^3*d + 2*C*a*b^5*c*d^3 + 2*C
*a*b^5*c^3*d))/((a^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^4*c^
2 + a^4*d^2 + b^4*c^2 + b^4*d^2 + 2*a^2*b^2*c^2 + 2*a^2*b^2*d^2)))/(tan(e +
 f*x)*(a^2*d + 2*a*b*c) + a^2*c + tan(e + f*x)^2*(b^2*c + 2*a*b*d) + b^2*d*
tan(e + f*x)^3))/f
```

sympy [**F(-2)**] time = 0.00, size = 0, normalized size = 0.00

Exception raised: NotImplementedError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**3/(c+d*tan(f*x
+e))**2,x)
```

```
[Out] Exception raised: NotImplementedError
```

3.84 $\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$

Optimal. Leaf size=804

$$-\frac{\left(Cc^2-Bdc+Ad^2\right)(a+b\tan(e+fx))^3}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}-\frac{\left(2a\left(2c(A-C)d-B\left(c^2-d^2\right)\right)d^2+b\left(3Cc^4-Bdc^3-(A-7C)d^2c^2-\right.\right.}{2d^2\left(c^2+d^2\right)^2f(c+d\tan(e+fx))}$$

```
[Out] -(3*a*b^2*(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)+a^3*(c^3*C-3*B*
c^2*d-3*c*C*d^2+B*d^3-A*(c^3-3*c*d^2))-3*a^2*b*((A-C)*d*(3*c^2-d^2)-B*(c^3-
3*c*d^2))+b^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2)))*x/(c^2+d^2)^3-(3*a^2*b
*(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)-b^3*(A*c^3-3*A*c*d^2+3*B
*c^2*d-B*d^3-C*c^3+3*C*c*d^2)-a^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2))+3*a
*b^2*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2)))*ln(cos(f*x+e))/(c^2+d^2)^3/f-(-
a*d+b*c)*(b^2*(3*c^6*C-B*c^5*d+9*c^4*C*d^2-3*B*c^3*d^3-c^2*(A-10*C)*d^4-6*B
*c*d^5+3*A*d^6)+a^2*d^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2))+a*b*d^2*(8*c*
(A-C)*d^3-B*(c^4+6*c^2*d^2-3*d^4)))*ln(c+d*tan(f*x+e))/d^4/(c^2+d^2)^3/f+b^
2*(b*(3*c^4*C-B*c^3*d+6*c^2*C*d^2-3*B*c*d^3+(2*A+C)*d^4)+a*d^2*(2*c*(A-C)*d
-B*(c^2-d^2)))*tan(f*x+e)/d^3/(c^2+d^2)^2/f-1/2*(A*d^2-B*c*d+C*c^2)*(a+b*ta
n(f*x+e))^3/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^2-1/2*(b*(3*c^4*C-B*c^3*d-c^2*(A
-7*C)*d^2-5*B*c*d^3+3*A*d^4)+2*a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*
x+e))^2/d^2/(c^2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] time = 2.75, antiderivative size = 804, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3645, 3637, 3626, 3617, 31, 3475}

$$-\frac{\left(Cc^2-Bdc+Ad^2\right)(a+b\tan(e+fx))^3}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}-\frac{\left(2a\left(2c(A-C)d-B\left(c^2-d^2\right)\right)d^2+b\left(3Cc^4-Bdc^3-(A-7C)d^2c^2-\right.\right.}{2d^2\left(c^2+d^2\right)^2f(c+d\tan(e+fx))}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^3,x]
```

```
[Out] -(((3*a*b^2*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) + a
^3*(c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)) - 3*a^2*b*((
A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) + b^3*((A - C)*d*(3*c^2 - d^2)
- B*(c^3 - 3*c*d^2)))*x)/(c^2 + d^2)^3) - ((3*a^2*b*(A*c^3 - c^3*C + 3*B*c^
2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) - b^3*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A
*c*d^2 + 3*c*C*d^2 - B*d^3) - a^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d
^2)) + 3*a*b^2*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)))*Log[Cos[e + f
*x]])/((c^2 + d^2)^3*f) - ((b*c - a*d)*(b^2*(3*c^6*C - B*c^5*d + 9*c^4*C*d^
```

```
2 - 3*B*c^3*d^3 - c^2*(A - 10*C)*d^4 - 6*B*c*d^5 + 3*A*d^6) + a^2*d^3*((A -
 C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) + a*b*d^2*(8*c*(A - C)*d^3 - B*(c^
4 + 6*c^2*d^2 - 3*d^4)))*Log[c + d*Tan[e + f*x]])/(d^4*(c^2 + d^2)^3*f) + (
b^2*(b*(3*c^4*C - B*c^3*d + 6*c^2*C*d^2 - 3*B*c*d^3 + (2*A + C)*d^4) + a*d^
2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Tan[e + f*x])/(d^3*(c^2 + d^2)^2*f) - ((
c^2*C - B*c*d + A*d^2)*(a + b*Tan[e + f*x])^3)/(2*d*(c^2 + d^2)*f*(c + d*Ta
n[e + f*x])^2) - ((b*(3*c^4*C - B*c^3*d - c^2*(A - 7*C)*d^2 - 5*B*c*d^3 + 3
*A*d^4) + 2*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*(a + b*Tan[e + f*x])^2)/
(2*d^2*(c^2 + d^2)^2*f*(c + d*Tan[e + f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
```

```
!LtQ[n, -1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx = -\frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} + \frac{\int}{}$$

$$= -\frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} - \frac{(}{}$$

$$= \frac{b^2\left(b\left(3c^4C-Bc^3d+6c^2Cd^2-3Bcd^3+(2A\right.\right.}{d^3(}$$

$$= -\frac{\left(3ab^2\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2\right.\right.}{}$$

$$= -\frac{\left(3ab^2\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2\right.\right.}{}$$

$$= -\frac{\left(3ab^2\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2\right.\right.}{}$$

Mathematica [A] time = 15.60, size = 1445, normalized size = 1.80

result too large to display

Warning: Unable to verify antiderivative.

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^3,x]
```

```
[Out] ((3*a*b^2*(-(A*c^3) + c^3*C - 3*B*c^2*d + 3*A*c*d^2 - 3*c*C*d^2 + B*d^3) +
a^3*(-(c^3*C) + 3*B*c^2*d + 3*c*C*d^2 - B*d^3 + A*(c^3 - 3*c*d^2)) - 3*a^2*
b*((A - C)*d*(-3*c^2 + d^2) + B*(c^3 - 3*c*d^2)) + b^3*((A - C)*d*(-3*c^2 +
 d^2) + B*(c^3 - 3*c*d^2)))*(e + f*x)*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(
a + b*Tan[e + f*x])^3)/((c^2 + d^2)^3*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^3
*(c + d*Tan[e + f*x])^3) - (b^2*(-3*b*c*C + b*B*d + 3*a*C*d)*Log[1 - Tan[(e
 + f*x)/2]^2]*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^3)/(
d^4*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^3*(c + d*Tan[e + f*x])^3) + ((-3*a^
2*b*(-(A*c^3) + c^3*C - 3*B*c^2*d + 3*A*c*d^2 - 3*c*C*d^2 + B*d^3) + b^3*(-
(A*c^3) + c^3*C - 3*B*c^2*d + 3*A*c*d^2 - 3*c*C*d^2 + B*d^3) + a^3*((A - C)
*d*(-3*c^2 + d^2) + B*(c^3 - 3*c*d^2)) - 3*a*b^2*((A - C)*d*(-3*c^2 + d^2)
+ B*(c^3 - 3*c*d^2)))*Log[1 + Tan[(e + f*x)/2]^2]*(c*Cos[e + f*x] + d*Sin[e
 + f*x])^3*(a + b*Tan[e + f*x])^3)/((c^2 + d^2)^3*f*(a*Cos[e + f*x] + b*Sin
[e + f*x])^3*(c + d*Tan[e + f*x])^3) + ((-(b*c) + a*d)*(b^2*(3*c^6*C - B*c^
5*d + 9*c^4*C*d^2 - 3*B*c^3*d^3 - c^2*(A - 10*C)*d^4 - 6*B*c*d^5 + 3*A*d^6)
 + a^2*d^3*(-((A - C)*d*(-3*c^2 + d^2)) - B*(c^3 - 3*c*d^2)) - a*b*d^2*(8*c
*(-A + C)*d^3 + B*(c^4 + 6*c^2*d^2 - 3*d^4)))*Log[-2*d*Tan[(e + f*x)/2] + c
*(-1 + Tan[(e + f*x)/2]^2)]*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[
e + f*x])^3)/(d^4*(c^2 + d^2)^3*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^3*(c +
d*Tan[e + f*x])^3) - (2*b^3*C*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*Tan[(e +
f*x)/2]*(a + b*Tan[e + f*x])^3)/(d^3*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^3*
(-1 + Tan[(e + f*x)/2]^2)*(c + d*Tan[e + f*x])^3) + (2*(b*c - a*d)^3*(c^2*C
 - B*c*d + A*d^2)*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(c + 2*d*Tan[(e + f*x
)/2])*(a + b*Tan[e + f*x])^3)/(c^3*d^2*(c^2 + d^2)*f*(a*Cos[e + f*x] + b*Si
n[e + f*x])^3*(c + 2*d*Tan[(e + f*x)/2] - c*Tan[(e + f*x)/2]^2)^2*(c + d*Ta
n[e + f*x])^3) - (2*(b*c - a*d)^2*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a*d*
(c^2*(A + C)*d^3 + A*d^5 + c^5*C*Tan[(e + f*x)/2] + c*d^4*(-B + A*Tan[(e +
f*x)/2]) + c^4*d*(C - 2*B*Tan[(e + f*x)/2]) - c^3*d^2*(B - 3*A*Tan[(e + f*x
)/2] + C*Tan[(e + f*x)/2])) + b*c*(-(A*d^5) + 2*c^5*C*Tan[(e + f*x)/2] + c*
d^4*(B + 2*A*Tan[(e + f*x)/2]) - c^4*d*(C + B*Tan[(e + f*x)/2]) - c^2*d^3*(
A + C + 3*B*Tan[(e + f*x)/2]) + c^3*d^2*(B + 4*C*Tan[(e + f*x)/2])))*(a + b
*Tan[e + f*x])^3)/(c^3*d^3*(c^2 + d^2)^2*f*(a*Cos[e + f*x] + b*Sin[e + f*x]
)^3*(-2*d*Tan[(e + f*x)/2] + c*(-1 + Tan[(e + f*x)/2]^2))*(c + d*Tan[e + f*
x])^3)
```

fricas [B] time = 2.65, size = 2490, normalized size = 3.10

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="fricas")

[Out] -1/2*(3*C*b^3*c^7*d^2 + A*a^3*d^9 - (3*C*a*b^2 + B*b^3)*c^6*d^3 - (3*C*a^2*
b + 3*B*a*b^2 + (A - 9*C)*b^3)*c^5*d^4 + (3*C*a^3 + 9*B*a^2*b + 3*(3*A - 7*
C)*a*b^2 - 7*B*b^3)*c^4*d^5 - 5*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - A*b^
3)*c^3*d^6 + ((7*A - 3*C)*a^3 - 9*B*a^2*b - 9*A*a*b^2)*c^2*d^7 + (B*a^3 + 3
*A*a^2*b)*c*d^8 - 2*(C*b^3*c^6*d^3 + 3*C*b^3*c^4*d^5 + 3*C*b^3*c^2*d^7 + C*
b^3*d^9)*tan(f*x + e)^3 - 2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B
*b^3)*c^5*d^4 + 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c^4*d
^5 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c^3*d^6 - (B*a^3
 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c^2*d^7)*f*x - (9*C*b^3*c^7*d
^2 - A*a^3*d^9 - 3*(3*C*a*b^2 + B*b^3)*c^6*d^3 + (3*C*a^2*b + 3*B*a*b^2 + (
A + 23*C)*b^3)*c^5*d^4 + (C*a^3 + 3*B*a^2*b + 3*(A - 9*C)*a*b^2 - 9*B*b^3)*
c^4*d^5 - (3*B*a^3 + 3*(3*A - 7*C)*a^2*b - 21*B*a*b^2 - (7*A + 12*C)*b^3)*c
^3*d^6 + 5*((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*c^2*d^7 + (3*B*a^3 + 9*A*a
^2*b + 4*C*b^3)*c*d^8 + 2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b
^3)*c^3*d^6 + 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c^2*d^7
 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d^8 - (B*a^3 + 3
*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d^9)*f*x)*tan(f*x + e)^2 + (3*C*b
^3*c^9 + 9*C*b^3*c^7*d^2 - (3*C*a*b^2 + B*b^3)*c^8*d - 3*(3*C*a*b^2 + B*b^3
)*c^6*d^3 + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - 10*C)*b^3)*c^5*d^4
- 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - 2*C)*a*b^2 + 2*B*b^3)*c^4*d^5 - 3*(B*
a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - A*b^3)*c^3*d^6 + ((A - C)*a^3 - 3*B*a^2
*b - 3*A*a*b^2)*c^2*d^7 + (3*C*b^3*c^7*d^2 + 9*C*b^3*c^5*d^4 - (3*C*a*b^2 +
 B*b^3)*c^6*d^3 - 3*(3*C*a*b^2 + B*b^3)*c^4*d^5 + (B*a^3 + 3*(A - C)*a^2*b
- 3*B*a*b^2 - (A - 10*C)*b^3)*c^3*d^6 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A -
 2*C)*a*b^2 + 2*B*b^3)*c^2*d^7 - 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - A
*b^3)*c*d^8 + ((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*d^9)*tan(f*x + e)^2 + 2
*(3*C*b^3*c^8*d + 9*C*b^3*c^6*d^3 - (3*C*a*b^2 + B*b^3)*c^7*d^2 - 3*(3*C*a*
b^2 + B*b^3)*c^5*d^4 + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - 10*C)*b^
3)*c^4*d^5 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - 2*C)*a*b^2 + 2*B*b^3)*c^3*
d^6 - 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - A*b^3)*c^2*d^7 + ((A - C)*a^
3 - 3*B*a^2*b - 3*A*a*b^2)*c*d^8)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2
*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) - (3*C*b^3*c^9 + 9*C*b^3*c^7
*d^2 + 9*C*b^3*c^5*d^4 + 3*C*b^3*c^3*d^6 - (3*C*a*b^2 + B*b^3)*c^8*d - 3*(3
*C*a*b^2 + B*b^3)*c^6*d^3 - 3*(3*C*a*b^2 + B*b^3)*c^4*d^5 - (3*C*a*b^2 + B*
b^3)*c^2*d^7 + (3*C*b^3*c^7*d^2 + 9*C*b^3*c^5*d^4 + 9*C*b^3*c^3*d^6 + 3*C*b
^3*c*d^8 - (3*C*a*b^2 + B*b^3)*c^6*d^3 - 3*(3*C*a*b^2 + B*b^3)*c^4*d^5 - 3*
(3*C*a*b^2 + B*b^3)*c^2*d^7 - (3*C*a*b^2 + B*b^3)*d^9)*tan(f*x + e)^2 + 2*(
3*C*b^3*c^8*d + 9*C*b^3*c^6*d^3 + 9*C*b^3*c^4*d^5 + 3*C*b^3*c^2*d^7 - (3*C*
a*b^2 + B*b^3)*c^7*d^2 - 3*(3*C*a*b^2 + B*b^3)*c^5*d^4 - 3*(3*C*a*b^2 + B*b
^3)*c^3*d^6 - (3*C*a*b^2 + B*b^3)*c*d^8)*tan(f*x + e))*log(1/(tan(f*x + e)^
2 + 1)) - 2*(3*C*b^3*c^8*d + 6*C*b^3*c^6*d^3 - (3*C*a*b^2 + B*b^3)*c^7*d^2
+ (C*a^3 + 3*B*a^2*b + 3*(A - 3*C)*a*b^2 - 3*B*b^3)*c^5*d^4 - (2*B*a^3 + 3*
```

```
(2*A - 3*C)*a^2*b - 9*B*a*b^2 - (3*A - 2*C)*b^3)*c^4*d^5 + (3*(A - C)*a^3 -
 9*B*a^2*b - 3*(3*A - 4*C)*a*b^2 + 4*B*b^3)*c^3*d^6 + (3*B*a^3 + 9*(A - C)*
a^2*b - 9*B*a*b^2 - (3*A - C)*b^3)*c^2*d^7 - ((3*A - 2*C)*a^3 - 6*B*a^2*b -
 6*A*a*b^2)*c*d^8 - (B*a^3 + 3*A*a^2*b)*d^9 + 2*(((A - C)*a^3 - 3*B*a^2*b -
 3*(A - C)*a*b^2 + B*b^3)*c^4*d^5 + 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2
- (A - C)*b^3)*c^3*d^6 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b
^3)*c^2*d^7 - (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d^8)*f*
x)*tan(f*x + e))/((c^6*d^6 + 3*c^4*d^8 + 3*c^2*d^10 + d^12)*f*tan(f*x + e)^
2 + 2*(c^7*d^5 + 3*c^5*d^7 + 3*c^3*d^9 + c*d^11)*f*tan(f*x + e) + (c^8*d^4
+ 3*c^6*d^6 + 3*c^4*d^8 + c^2*d^10)*f)
```

giac [B] time = 5.09, size = 2505, normalized size = 3.12

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*C*b^3*tan(f*x + e)/d^3 + 2*(A*a^3*c^3 - C*a^3*c^3 - 3*B*a^2*b*c^3 -
3*A*a*b^2*c^3 + 3*C*a*b^2*c^3 + B*b^3*c^3 + 3*B*a^3*c^2*d + 9*A*a^2*b*c^2*d
 - 9*C*a^2*b*c^2*d - 9*B*a*b^2*c^2*d - 3*A*b^3*c^2*d + 3*C*b^3*c^2*d - 3*A*
a^3*c*d^2 + 3*C*a^3*c*d^2 + 9*B*a^2*b*c*d^2 + 9*A*a*b^2*c*d^2 - 9*C*a*b^2*c
*d^2 - 3*B*b^3*c*d^2 - B*a^3*d^3 - 3*A*a^2*b*d^3 + 3*C*a^2*b*d^3 + 3*B*a*b^
2*d^3 + A*b^3*d^3 - C*b^3*d^3)*(f*x + e)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6
) + (B*a^3*c^3 + 3*A*a^2*b*c^3 - 3*C*a^2*b*c^3 - 3*B*a*b^2*c^3 - A*b^3*c^3
+ C*b^3*c^3 - 3*A*a^3*c^2*d + 3*C*a^3*c^2*d + 9*B*a^2*b*c^2*d + 9*A*a*b^2*c
^2*d - 9*C*a*b^2*c^2*d - 3*B*b^3*c^2*d - 3*B*a^3*c*d^2 - 9*A*a^2*b*c*d^2 +
9*C*a^2*b*c*d^2 + 9*B*a*b^2*c*d^2 + 3*A*b^3*c*d^2 - 3*C*b^3*c*d^2 + A*a^3*d
^3 - C*a^3*d^3 - 3*B*a^2*b*d^3 - 3*A*a*b^2*d^3 + 3*C*a*b^2*d^3 + B*b^3*d^3)
*log(tan(f*x + e)^2 + 1)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) - 2*(3*C*b^3*c
^7 - 3*C*a*b^2*c^6*d - B*b^3*c^6*d + 9*C*b^3*c^5*d^2 - 9*C*a*b^2*c^4*d^3 -
3*B*b^3*c^4*d^3 + B*a^3*c^3*d^4 + 3*A*a^2*b*c^3*d^4 - 3*C*a^2*b*c^3*d^4 - 3
*B*a*b^2*c^3*d^4 - A*b^3*c^3*d^4 + 10*C*b^3*c^3*d^4 - 3*A*a^3*c^2*d^5 + 3*C
*a^3*c^2*d^5 + 9*B*a^2*b*c^2*d^5 + 9*A*a*b^2*c^2*d^5 - 18*C*a*b^2*c^2*d^5 -
 6*B*b^3*c^2*d^5 - 3*B*a^3*c*d^6 - 9*A*a^2*b*c*d^6 + 9*C*a^2*b*c*d^6 + 9*B*
a*b^2*c*d^6 + 3*A*b^3*c*d^6 + A*a^3*d^7 - C*a^3*d^7 - 3*B*a^2*b*d^7 - 3*A*a
*b^2*d^7)*log(abs(d*tan(f*x + e) + c))/(c^6*d^4 + 3*c^4*d^6 + 3*c^2*d^8 + d
^10) + (9*C*b^3*c^7*d^2*tan(f*x + e)^2 - 9*C*a*b^2*c^6*d^3*tan(f*x + e)^2 -
 3*B*b^3*c^6*d^3*tan(f*x + e)^2 + 27*C*b^3*c^5*d^4*tan(f*x + e)^2 - 27*C*a*
b^2*c^4*d^5*tan(f*x + e)^2 - 9*B*b^3*c^4*d^5*tan(f*x + e)^2 + 3*B*a^3*c^3*d
^6*tan(f*x + e)^2 + 9*A*a^2*b*c^3*d^6*tan(f*x + e)^2 - 9*C*a^2*b*c^3*d^6*ta
n(f*x + e)^2 - 9*B*a*b^2*c^3*d^6*tan(f*x + e)^2 - 3*A*b^3*c^3*d^6*tan(f*x +
 e)^2 + 30*C*b^3*c^3*d^6*tan(f*x + e)^2 - 9*A*a^3*c^2*d^7*tan(f*x + e)^2 +
9*C*a^3*c^2*d^7*tan(f*x + e)^2 + 27*B*a^2*b*c^2*d^7*tan(f*x + e)^2 + 27*A*a
```

```
*b^2*c^2*d^7*tan(f*x + e)^2 - 54*C*a*b^2*c^2*d^7*tan(f*x + e)^2 - 18*B*b^3*
c^2*d^7*tan(f*x + e)^2 - 9*B*a^3*c*d^8*tan(f*x + e)^2 - 27*A*a^2*b*c*d^8*ta
n(f*x + e)^2 + 27*C*a^2*b*c*d^8*tan(f*x + e)^2 + 27*B*a*b^2*c*d^8*tan(f*x +
 e)^2 + 9*A*b^3*c*d^8*tan(f*x + e)^2 + 3*A*a^3*d^9*tan(f*x + e)^2 - 3*C*a^3
*d^9*tan(f*x + e)^2 - 9*B*a^2*b*d^9*tan(f*x + e)^2 - 9*A*a*b^2*d^9*tan(f*x
+ e)^2 + 12*C*b^3*c^8*d*tan(f*x + e) - 6*C*a*b^2*c^7*d^2*tan(f*x + e) - 2*B
*b^3*c^7*d^2*tan(f*x + e) - 6*C*a^2*b*c^6*d^3*tan(f*x + e) - 6*B*a*b^2*c^6*
d^3*tan(f*x + e) - 2*A*b^3*c^6*d^3*tan(f*x + e) + 38*C*b^3*c^6*d^3*tan(f*x
+ e) - 18*C*a*b^2*c^5*d^4*tan(f*x + e) - 6*B*b^3*c^5*d^4*tan(f*x + e) + 8*B
*a^3*c^4*d^5*tan(f*x + e) + 24*A*a^2*b*c^4*d^5*tan(f*x + e) - 42*C*a^2*b*c^
4*d^5*tan(f*x + e) - 42*B*a*b^2*c^4*d^5*tan(f*x + e) - 14*A*b^3*c^4*d^5*tan
(f*x + e) + 50*C*b^3*c^4*d^5*tan(f*x + e) - 22*A*a^3*c^3*d^6*tan(f*x + e) +
 22*C*a^3*c^3*d^6*tan(f*x + e) + 66*B*a^2*b*c^3*d^6*tan(f*x + e) + 66*A*a*b
^2*c^3*d^6*tan(f*x + e) - 84*C*a*b^2*c^3*d^6*tan(f*x + e) - 28*B*b^3*c^3*d^
6*tan(f*x + e) - 18*B*a^3*c^2*d^7*tan(f*x + e) - 54*A*a^2*b*c^2*d^7*tan(f*x
 + e) + 36*C*a^2*b*c^2*d^7*tan(f*x + e) + 36*B*a*b^2*c^2*d^7*tan(f*x + e) +
 12*A*b^3*c^2*d^7*tan(f*x + e) + 2*A*a^3*c*d^8*tan(f*x + e) - 2*C*a^3*c*d^8
*tan(f*x + e) - 6*B*a^2*b*c*d^8*tan(f*x + e) - 6*A*a*b^2*c*d^8*tan(f*x + e)
 - 2*B*a^3*d^9*tan(f*x + e) - 6*A*a^2*b*d^9*tan(f*x + e) + 4*C*b^3*c^9 - 3*
C*a^2*b*c^7*d^2 - 3*B*a*b^2*c^7*d^2 - A*b^3*c^7*d^2 + 13*C*b^3*c^7*d^2 - C*
a^3*c^6*d^3 - 3*B*a^2*b*c^6*d^3 - 3*A*a*b^2*c^6*d^3 + 3*C*a*b^2*c^6*d^3 + B
*b^3*c^6*d^3 + 6*B*a^3*c^5*d^4 + 18*A*a^2*b*c^5*d^4 - 27*C*a^2*b*c^5*d^4 -
27*B*a*b^2*c^5*d^4 - 9*A*b^3*c^5*d^4 + 21*C*b^3*c^5*d^4 - 14*A*a^3*c^4*d^5
+ 11*C*a^3*c^4*d^5 + 33*B*a^2*b*c^4*d^5 + 33*A*a*b^2*c^4*d^5 - 33*C*a*b^2*c
^4*d^5 - 11*B*b^3*c^4*d^5 - 7*B*a^3*c^3*d^6 - 21*A*a^2*b*c^3*d^6 + 12*C*a^2
*b*c^3*d^6 + 12*B*a*b^2*c^3*d^6 + 4*A*b^3*c^3*d^6 - 3*A*a^3*c^2*d^7 - B*a^3
*c*d^8 - 3*A*a^2*b*c*d^8 - A*a^3*d^9)/((c^6*d^4 + 3*c^4*d^6 + 3*c^2*d^8 + d
^10)*(d*tan(f*x + e) + c)^2))/f
```

maple [B] time = 0.29, size = 3522, normalized size = 4.38

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x
)
```

```
[Out] 3/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*a^3*c^2*d-3/f/(c^2+d^2)^3*B*arctan(tan
(f*x+e))*a^2*b*c^3+3/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*a*b^2*d^3-1/f/d^2/(
c^2+d^2)^2/(c+d*tan(f*x+e))*A*b^3*c^4-1/2/f/d^3/(c^2+d^2)/(c+d*tan(f*x+e))^
2*B*c^4*b^3-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*a*b^2*d^3+3/2/f/(c^2+d^2
)^3*ln(1+tan(f*x+e)^2)*A*b^3*c*d^2+1/2/f/d^4/(c^2+d^2)/(c+d*tan(f*x+e))^2*C
*c^5*b^3+3/f*d^2/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*a^3*c-3/2/f/(c^2+d^2)^3*l
n(1+tan(f*x+e)^2)*C*b^3*c*d^2+6/f*d/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*a^2*b*c-
3/f/d^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*a*b^2*c^4-3/f/d^2/(c^2+d^2)^2/(c+d*t
```

an(f*x+e))*C*a^2*b*c^4+6/f/d^3/(c^2+d^2)^2/(c+d*tan(f*x+e))*C*a*b^2*c^5+12/
f/d/(c^2+d^2)^2/(c+d*tan(f*x+e))*C*a*b^2*c^3+9/f*d^2/(c^2+d^2)^3*ln(c+d*tan
(f*x+e))*A*a^2*b*c+18/f*d/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*a*b^2*c^2-9/2/f/
(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*a^2*b*c*d^2+9/2/f/(c^2+d^2)^3*ln(1+tan(f*x
+e)^2)*A*a*b^2*c^2*d+9/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a^2*b*c^2*d-9/f
*d/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*a*b^2*c^2-9/f*d/(c^2+d^2)^3*ln(c+d*tan(
f*x+e))*B*a^2*b*c^2+9/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a^2*b*c*d^2-9/2/
f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a*b^2*c^2*d+9/f/(c^2+d^2)^3*A*arctan(tan
(f*x+e))*a^2*b*c^2*d+9/f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*a*b^2*c*d^2+9/f/(
c^2+d^2)^3*B*arctan(tan(f*x+e))*a^2*b*c*d^2-9/f/(c^2+d^2)^3*B*arctan(tan(f*
x+e))*a*b^2*c^2*d-9/f/(c^2+d^2)^3*C*arctan(tan(f*x+e))*a^2*b*c^2*d-9/f/(c^2
+d^2)^3*C*arctan(tan(f*x+e))*a*b^2*c*d^2-3/2/f/d/(c^2+d^2)/(c+d*tan(f*x+e))
^2*A*a*b^2*c^2+9/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a*b^2*c*d^2-9/f*d^2/(
c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*a*b^2*c-3/2/f/d/(c^2+d^2)/(c+d*tan(f*x+e))^
2*B*a^2*b*c^2+3/2/f/d^2/(c^2+d^2)/(c+d*tan(f*x+e))^2*B*a*b^2*c^3+3/2/f/d^2/
(c^2+d^2)/(c+d*tan(f*x+e))^2*C*a^2*b*c^3-3/2/f/d^3/(c^2+d^2)/(c+d*tan(f*x+e
))^2*C*c^4*a*b^2+6/f*d/(c^2+d^2)^2/(c+d*tan(f*x+e))*A*a*b^2*c-9/f*d^2/(c^2+
d^2)^3*ln(c+d*tan(f*x+e))*C*a^2*b*c+3/f/d^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*
C*a*b^2*c^6+9/f/d/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*a*b^2*c^4+3/2/f/(c^2+d^2
)^3*ln(1+tan(f*x+e)^2)*C*a^3*c^2*d-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a
^2*b*c^3+3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a*b^2*d^3+1/2/f/d^2/(c^2+d^
2)/(c+d*tan(f*x+e))^2*A*b^3*c^3+1/f*b^3*C/d^3*tan(f*x+e)+3/f/(c^2+d^2)^3*ln
(c+d*tan(f*x+e))*C*a^2*b*c^3+3/2/f/(c^2+d^2)/(c+d*tan(f*x+e))^2*A*a^2*b*c+3
/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*A*a^2*b*c^2+3/f/(c^2+d^2)^3*ln(c+d*tan(f*x+
e))*B*a*b^2*c^3-3/f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*a^3*c*d^2-3/f/(c^2+d^2
)^3*A*arctan(tan(f*x+e))*a^2*b*d^3-3/f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*a*b
^2*c^3-3/f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*b^3*c^2*d+6/f*d/(c^2+d^2)^3*ln(
c+d*tan(f*x+e))*B*b^3*c^2-3/f*d/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*a^3*c^2-3/
f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*b^3*c*d^2-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x
+e)^2)*A*a^3*c^2*d+3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*a^2*b*c^3+3/f/d/(
c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*b^3*c^4+3/f*d^3/(c^2+d^2)^3*ln(c+d*tan(f*x+
e))*B*a^2*b+1/f/d^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*b^3*c^6-9/f/d^2/(c^2+d
^2)^3*ln(c+d*tan(f*x+e))*C*b^3*c^5-1/f*d^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A
*a^3-1/f/(c^2+d^2)^3*C*arctan(tan(f*x+e))*a^3*c^3-1/f/(c^2+d^2)^3*C*arctan(
tan(f*x+e))*b^3*d^3+1/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*a^3*c^2+1/f/(c^2+d^2
)^3*ln(c+d*tan(f*x+e))*A*b^3*c^3-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a^2
*b*d^3-9/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*a*b^2*c^2-9/f/(c^2+d^2)^2/(c+d*ta
n(f*x+e))*C*a^2*b*c^2-3/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*a^2*b*c^3-1/2/f/
d/(c^2+d^2)/(c+d*tan(f*x+e))^2*C*a^3*c^2-2/f*d/(c^2+d^2)^2/(c+d*tan(f*x+e))
*A*a^3*c-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a*b^2*c^3+3/f/(c^2+d^2)^3*C
*arctan(tan(f*x+e))*a^3*c*d^2+3/f/(c^2+d^2)^3*C*arctan(tan(f*x+e))*a^2*b*d^
3+3/f/(c^2+d^2)^3*C*arctan(tan(f*x+e))*a*b^2*c^3+2/f/d^3/(c^2+d^2)^2/(c+d*t
an(f*x+e))*B*b^3*c^5+4/f/d/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*b^3*c^3+2/f*d/(c^
2+d^2)^2/(c+d*tan(f*x+e))*C*a^3*c+3/f*d/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*a^
3*c^2-3/f*d^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*A*a^2*b-3/f*d^2/(c^2+d^2)^3*ln(c

```
+d*tan(f*x+e))*A*b^3*c-3/f/d^4/(c^2+d^2)^2/(c+d*tan(f*x+e))*C*b^3*c^6-5/f/d
^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*C*b^3*c^4-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)
^2)*B*a^3*c*d^2+1/f*d^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*a^3+1/f/(c^2+d^2)^
3*A*arctan(tan(f*x+e))*a^3*c^3+1/f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*b^3*d^3
-1/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*a^3*d^3-1/2/f/(c^2+d^2)^3*ln(1+tan(f*
x+e)^2)*A*b^3*c^3+1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a^3*c^3+1/f/(c^2+d
^2)^3*B*arctan(tan(f*x+e))*b^3*c^3+1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*b
^3*d^3-1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*a^3*C*d^3+1/2/f/(c^2+d^2)^3*ln(
1+tan(f*x+e)^2)*C*b^3*c^3-1/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*a^3*c^3-10/f
/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*b^3*c^3-3/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*
A*b^3*c^2+1/2/f/(c^2+d^2)/(c+d*tan(f*x+e))^2*B*a^3*c+1/2/f/(c^2+d^2)^3*ln(1
+tan(f*x+e)^2)*A*a^3*d^3-1/2/f*d/(c^2+d^2)/(c+d*tan(f*x+e))^2*A*a^3-1/f*d^2
/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*a^3-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*
b^3*c^2*d+3/f/(c^2+d^2)^3*C*arctan(tan(f*x+e))*b^3*c^2*d+3/f*d^3/(c^2+d^2)^
3*ln(c+d*tan(f*x+e))*A*a*b^2-3/f/d^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*b^3*c
^7
```

maxima [A] time = 0.60, size = 1110, normalized size = 1.38

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*C*b^3*tan(f*x + e)/d^3 + 2*(((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b
^2 + B*b^3)*c^3 + 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c^2
*d - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*c*d^2 - (B*a^3 +
 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*d^3)*(f*x + e)/(c^6 + 3*c^4*d^2
 + 3*c^2*d^4 + d^6) - 2*(3*C*b^3*c^7 + 9*C*b^3*c^5*d^2 - (3*C*a*b^2 + B*b^3
)*c^6*d - 3*(3*C*a*b^2 + B*b^3)*c^4*d^3 + (B*a^3 + 3*(A - C)*a^2*b - 3*B*a*
b^2 - (A - 10*C)*b^3)*c^3*d^4 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - 2*C)*a*
b^2 + 2*B*b^3)*c^2*d^5 - 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - A*b^3)*c*
d^6 + ((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*d^7)*log(d*tan(f*x + e) + c)/(c
^6*d^4 + 3*c^4*d^6 + 3*c^2*d^8 + d^10) + ((B*a^3 + 3*(A - C)*a^2*b - 3*B*a*
b^2 - (A - C)*b^3)*c^3 - 3*((A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b
^3)*c^2*d - 3*(B*a^3 + 3*(A - C)*a^2*b - 3*B*a*b^2 - (A - C)*b^3)*c*d^2 + (
(A - C)*a^3 - 3*B*a^2*b - 3*(A - C)*a*b^2 + B*b^3)*d^3)*log(tan(f*x + e)^2
+ 1)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) - (5*C*b^3*c^7 + A*a^3*d^7 - 3*(3*
C*a*b^2 + B*b^3)*c^6*d + (3*C*a^2*b + 3*B*a*b^2 + (A + 9*C)*b^3)*c^5*d^2 +
(C*a^3 + 3*B*a^2*b + 3*(A - 7*C)*a*b^2 - 7*B*b^3)*c^4*d^3 - (3*B*a^3 + 3*(3
*A - 5*C)*a^2*b - 15*B*a*b^2 - 5*A*b^3)*c^3*d^4 + ((5*A - 3*C)*a^3 - 9*B*a^
2*b - 9*A*a*b^2)*c^2*d^5 + (B*a^3 + 3*A*a^2*b)*c*d^6 + 2*(3*C*b^3*c^6*d - 2
*(3*C*a*b^2 + B*b^3)*c^5*d^2 + (3*C*a^2*b + 3*B*a*b^2 + (A + 5*C)*b^3)*c^4*
d^3 - 4*(3*C*a*b^2 + B*b^3)*c^3*d^4 - (B*a^3 + 3*(A - 3*C)*a^2*b - 9*B*a*b^
```

```
2 - 3*A*b^3)*c^2*d^5 + 2*((A - C)*a^3 - 3*B*a^2*b - 3*A*a*b^2)*c*d^6 + (B*a
^3 + 3*A*a^2*b)*d^7)*tan(f*x + e))/(c^6*d^4 + 2*c^4*d^6 + c^2*d^8 + (c^4*d^
6 + 2*c^2*d^8 + d^10)*tan(f*x + e)^2 + 2*(c^5*d^5 + 2*c^3*d^7 + c*d^9)*tan(
f*x + e)))/f
```

mupad [B] time = 20.60, size = 1172, normalized size = 1.46

$$\frac{\ln\left(c+d\tan\left(e+f\,x\right)\right)\,\left(d^3\,\left(3\,B\,b^3\,c^4+9\,C\,a\,b^2\,c^4\right)-d^6\,\left(3\,A\,b^3\,c-3\,B\,a^3\,c-9\,A\,a^2\,b\,c+9\,B\,a\,b^2\,c+9\,C\,a^2\,b\,c\right)}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^3,x)
```

```
[Out] (log(tan(e + f*x) + 1i)*(A*a^3 + A*b^3*1i - B*a^3*1i + B*b^3 - C*a^3 - C*b^
3*1i - 3*A*a*b^2 - A*a^2*b*3i + B*a*b^2*3i - 3*B*a^2*b + 3*C*a*b^2 + C*a^2*
b*3i))/(2*f*(c*d^2*3i - 3*c^2*d - c^3*1i + d^3)) - ((A*a^3*d^7 + 5*C*b^3*c^
7 + B*a^3*c*d^6 - 3*B*b^3*c^6*d + 5*A*a^3*c^2*d^5 + 5*A*b^3*c^3*d^4 + A*b^3
*c^5*d^2 - 3*B*a^3*c^3*d^4 - 7*B*b^3*c^4*d^3 - 3*C*a^3*c^2*d^5 + C*a^3*c^4*
d^3 + 9*C*b^3*c^5*d^2 - 9*A*a*b^2*c^2*d^5 + 3*A*a*b^2*c^4*d^3 - 9*A*a^2*b*c
^3*d^4 + 15*B*a*b^2*c^3*d^4 + 3*B*a*b^2*c^5*d^2 - 9*B*a^2*b*c^2*d^5 + 3*B*a
^2*b*c^4*d^3 - 21*C*a*b^2*c^4*d^3 + 15*C*a^2*b*c^3*d^4 + 3*C*a^2*b*c^5*d^2
+ 3*A*a^2*b*c*d^6 - 9*C*a*b^2*c^6*d)/(2*d*(c^4 + d^4 + 2*c^2*d^2)) + (tan(e
 + f*x)*(B*a^3*d^6 + 3*C*b^3*c^6 + 3*A*a^2*b*d^6 + 2*A*a^3*c*d^5 - 2*B*b^3*
c^5*d - 2*C*a^3*c*d^5 + 3*A*b^3*c^2*d^4 + A*b^3*c^4*d^2 - B*a^3*c^2*d^4 - 4
*B*b^3*c^3*d^3 + 5*C*b^3*c^4*d^2 - 3*A*a^2*b*c^2*d^4 + 9*B*a*b^2*c^2*d^4 +
3*B*a*b^2*c^4*d^2 - 12*C*a*b^2*c^3*d^3 + 9*C*a^2*b*c^2*d^4 + 3*C*a^2*b*c^4*
d^2 - 6*A*a*b^2*c*d^5 - 6*B*a^2*b*c*d^5 - 6*C*a*b^2*c^5*d))/(c^4 + d^4 + 2*
c^2*d^2))/(f*(c^2*d^3 + d^5*tan(e + f*x)^2 + 2*c*d^4*tan(e + f*x))) + (log(
c + d*tan(e + f*x))*(d^3*(3*B*b^3*c^4 + 9*C*a*b^2*c^4) - d^6*(3*A*b^3*c - 3
*B*a^3*c - 9*A*a^2*b*c + 9*B*a*b^2*c + 9*C*a^2*b*c) + d^5*(3*A*a^3*c^2 + 6*
B*b^3*c^2 - 3*C*a^3*c^2 - 9*A*a*b^2*c^2 - 9*B*a^2*b*c^2 + 18*C*a*b^2*c^2) +
 d^4*(A*b^3*c^3 - B*a^3*c^3 - 10*C*b^3*c^3 - 3*A*a^2*b*c^3 + 3*B*a*b^2*c^3
+ 3*C*a^2*b*c^3) + d^7*(C*a^3 - A*a^3 + 3*A*a*b^2 + 3*B*a^2*b) + d*(B*b^3*c
^6 + 3*C*a*b^2*c^6) - 3*C*b^3*c^7 - 9*C*b^3*c^5*d^2))/(f*(d^10 + 3*c^2*d^8
+ 3*c^4*d^6 + c^6*d^4)) + (log(tan(e + f*x) - 1i)*(A*a^3*1i + A*b^3 - B*a^3
 + B*b^3*1i - C*a^3*1i - C*b^3 - A*a*b^2*3i - 3*A*a^2*b + 3*B*a*b^2 - B*a^2
*b*3i + C*a*b^2*3i + 3*C*a^2*b))/(2*f*(3*c*d^2 - c^2*d*3i - c^3 + d^3*1i))
+ (C*b^3*tan(e + f*x))/(d^3*f)
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**3,x)
```

```
[Out] Exception raised: AttributeError
```

3.85 $\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$

Optimal. Leaf size=597

$$-\frac{\log(\cos(e+fx))\left(-\left(a^2\left(d(A-C)\left(3c^2-d^2\right)-B\left(c^3-3cd^2\right)\right)\right)+2ab\left(Ac^3-3Acd^2+3Bc^2d-Bd^3-c^3C+3c\right.}{f\left(c^2+d^2\right)^3}$$

```
[Out] -(b^2*(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)+a^2*(c^3*C-3*B*c^2*
d-3*c*C*d^2+B*d^3-A*(c^3-3*c*d^2))-2*a*b*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^
2)))*x/(c^2+d^2)^3-(2*a*b*(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)
-a^2*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2))+b^2*((A-C)*d*(3*c^2-d^2)-B*(c^3-
3*c*d^2)))*ln(cos(f*x+e))/(c^2+d^2)^3/f-(2*a*b*d^3*(A*c^3-3*A*c*d^2+3*B*c^2
*d-B*d^3-C*c^3+3*C*c*d^2)-b^2*(c^6*C+3*c^4*C*d^2+B*c^3*d^3-3*c^2*(A-2*C)*d^
4-3*B*c*d^5+A*d^6)-a^2*d^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2)))*ln(c+d*ta
n(f*x+e))/d^3/(c^2+d^2)^3/f-1/2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^2/d/(c
^2+d^2)/f/(c+d*tan(f*x+e))^2+(-a*d+b*c)*(b*(c^4*C-c^2*(A-3*C)*d^2-2*B*c*d^3
+A*d^4)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))/d^3/(c^2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] time = 1.39, antiderivative size = 597, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3645, 3635, 3626, 3617, 31, 3475}

$$-\frac{\left(-a^2d^3\left(d(A-C)\left(3c^2-d^2\right)-B\left(c^3-3cd^2\right)\right)+2abd^3\left(Ac^3-3Acd^2+3Bc^2d-Bd^3-c^3C+3cCd^2\right)-b^2\left(-3c\right.\right.}{d^3f\left(c^2+d^2\right)^3}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^3,x]
```

```
[Out] -(((b^2*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) + a^2*(
c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)) - 2*a*b*((A - C)
*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)))*x)/(c^2 + d^2)^3) - ((2*a*b*(A*c^3 -
 c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) - a^2*((A - C)*d*(3*c^2
 - d^2) - B*(c^3 - 3*c*d^2)) + b^2*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*
d^2)))*Log[Cos[e + f*x]])/((c^2 + d^2)^3*f) - ((2*a*b*d^3*(A*c^3 - c^3*C +
3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) - b^2*(c^6*C + 3*c^4*C*d^2 + B*c
^3*d^3 - 3*c^2*(A - 2*C)*d^4 - 3*B*c*d^5 + A*d^6) - a^2*d^3*((A - C)*d*(3*c
^2 - d^2) - B*(c^3 - 3*c*d^2)))*Log[c + d*Tan[e + f*x]])/(d^3*(c^2 + d^2)^3
*f) - ((c^2*C - B*c*d + A*d^2)*(a + b*Tan[e + f*x])^2)/(2*d*(c^2 + d^2)*f*(
c + d*Tan[e + f*x])^2) + ((b*c - a*d)*(b*(c^4*C - c^2*(A - 3*C)*d^2 - 2*B*c
*d^3 + A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2))))/(d^3*(c^2 + d^2)^2*
f*(c + d*Tan[e + f*x]))
```

Rule 31

```
Int[((a_) + (b_.)*(x_))^(-1), x_Symbol] :> Simp[Log[RemoveContent[a + b*x,
x]]/b, x] /; FreeQ[{a, b}, x]
```

Rule 3475

```
Int[tan[(c_.) + (d_.)*(x_)], x_Symbol] :> -Simp[Log[RemoveContent[Cos[c + d
*x], x]]/d, x] /; FreeQ[{c, d}, x]
```

Rule 3617

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_) + (C_.)*tan[(e_.) +
 (f_.)*(x_)]^2), x_Symbol] :> Dist[A/(b*f), Subst[Int[(a + x)^m, x], x, b*T
an[e + f*x]], x] /; FreeQ[{a, b, e, f, A, C, m}, x] && EqQ[A, C]
```

Rule 3626

```
Int[((A_) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2
)/((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)]), x_Symbol] :> Simp[((a*A + b*B -
a*C)*x)/(a^2 + b^2), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[(1
 + Tan[e + f*x]^2)/(a + b*Tan[e + f*x]), x], x] - Dist[(A*b - a*B - b*C)/(a
^2 + b^2), Int[Tan[e + f*x], x], x]) /; FreeQ[{a, b, e, f, A, B, C}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] && NeQ[a^2 + b^2, 0] && NeQ[A*b - a*B - b*C,
0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
```

```
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx = -\frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}+$$

$$= -\frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}+$$

$$= \frac{\left(a^2\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-\right.\right.}{}$$

$$= \frac{\left(a^2\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-\right.\right.}{}$$

$$= \frac{\left(a^2\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-\right.\right.}{}$$

Mathematica [C] time = 8.13, size = 2499, normalized size = 4.19

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^3,x]
```

```
[Out] ((-(b^2*c^4*C) + b^2*B*c^3*d + 2*a*b*c^3*C*d - A*b^2*c^2*d^2 - 2*a*b*B*c^2*
d^2 - a^2*c^2*C*d^2 + 2*a*A*b*c*d^3 + a^2*B*c*d^3 - a^2*A*d^4)*Sec[e + f*x]
*(c*Cos[e + f*x] + d*Sin[e + f*x])*(a + b*Tan[e + f*x])^2)/(2*(c - I*d)^2*(
c + I*d)^2*d*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^3)
+ ((a^2*A*c^3 - A*b^2*c^3 - 2*a*b*B*c^3 - a^2*c^3*C + b^2*c^3*C + 6*a*A*b*c
^2*d + 3*a^2*B*c^2*d - 3*b^2*B*c^2*d - 6*a*b*c^2*C*d - 3*a^2*A*c*d^2 + 3*A*
```

b^2*c*d^2 + 6*a*b*B*c*d^2 + 3*a^2*c*C*d^2 - 3*b^2*c*C*d^2 - 2*a*A*b*d^3 - a
^2*B*d^3 + b^2*B*d^3 + 2*a*b*C*d^3)*(e + f*x)*Sec[e + f*x]*(c*Cos[e + f*x]
+ d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^2)/((c - I*d)^3*(c + I*d)^3*f*(a*C
os[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^3) + ((I*b^2*c^13*C*d^
2 + b^2*c^12*C*d^3 + (5*I)*b^2*c^11*C*d^4 - (2*I)*a*A*b*c^10*d^5 - I*a^2*B*
c^10*d^5 + I*b^2*B*c^10*d^5 + (2*I)*a*b*c^10*C*d^5 + 5*b^2*c^10*C*d^5 + (3*
I)*a^2*A*c^9*d^6 - 2*a*A*b*c^9*d^6 - (3*I)*A*b^2*c^9*d^6 - a^2*B*c^9*d^6 -
(6*I)*a*b*B*c^9*d^6 + b^2*B*c^9*d^6 - (3*I)*a^2*c^9*C*d^6 + 2*a*b*c^9*C*d^6
 + (13*I)*b^2*c^9*C*d^6 + 3*a^2*A*c^8*d^7 + (2*I)*a*A*b*c^8*d^7 - 3*A*b^2*c
^8*d^7 + I*a^2*B*c^8*d^7 - 6*a*b*B*c^8*d^7 - I*b^2*B*c^8*d^7 - 3*a^2*c^8*C*
d^7 - (2*I)*a*b*c^8*C*d^7 + 13*b^2*c^8*C*d^7 + (5*I)*a^2*A*c^7*d^8 + 2*a*A*
b*c^7*d^8 - (5*I)*A*b^2*c^7*d^8 + a^2*B*c^7*d^8 - (10*I)*a*b*B*c^7*d^8 - b^
2*B*c^7*d^8 - (5*I)*a^2*c^7*C*d^8 - 2*a*b*c^7*C*d^8 + (15*I)*b^2*c^7*C*d^8
+ 5*a^2*A*c^6*d^9 + (10*I)*a*A*b*c^6*d^9 - 5*A*b^2*c^6*d^9 + (5*I)*a^2*B*c^
6*d^9 - 10*a*b*B*c^6*d^9 - (5*I)*b^2*B*c^6*d^9 - 5*a^2*c^6*C*d^9 - (10*I)*a
*b*c^6*C*d^9 + 15*b^2*c^6*C*d^9 + I*a^2*A*c^5*d^10 + 10*a*A*b*c^5*d^10 - I*
A*b^2*c^5*d^10 + 5*a^2*B*c^5*d^10 - (2*I)*a*b*B*c^5*d^10 - 5*b^2*B*c^5*d^10
 - I*a^2*c^5*C*d^10 - 10*a*b*c^5*C*d^10 + (6*I)*b^2*c^5*C*d^10 + a^2*A*c^4*
d^11 + (6*I)*a*A*b*c^4*d^11 - A*b^2*c^4*d^11 + (3*I)*a^2*B*c^4*d^11 - 2*a*b
*B*c^4*d^11 - (3*I)*b^2*B*c^4*d^11 - a^2*c^4*C*d^11 - (6*I)*a*b*c^4*C*d^11
+ 6*b^2*c^4*C*d^11 - I*a^2*A*c^3*d^12 + 6*a*A*b*c^3*d^12 + I*A*b^2*c^3*d^12
 + 3*a^2*B*c^3*d^12 + (2*I)*a*b*B*c^3*d^12 - 3*b^2*B*c^3*d^12 + I*a^2*c^3*C
*d^12 - 6*a*b*c^3*C*d^12 - a^2*A*c^2*d^13 + A*b^2*c^2*d^13 + 2*a*b*B*c^2*d^
13 + a^2*c^2*C*d^13)*(e + f*x)*Sec[e + f*x]*(c*Cos[e + f*x] + d*Sin[e + f*x
])^3*(a + b*Tan[e + f*x])^2)/(c^2*(c - I*d)^6*(c + I*d)^5*d^5*f*(a*Cos[e +
f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^3) - (I*(b^2*c^6*C + 3*b^2*c^
4*C*d^2 - 2*a*A*b*c^3*d^3 - a^2*B*c^3*d^3 + b^2*B*c^3*d^3 + 2*a*b*c^3*C*d^3
 + 3*a^2*A*c^2*d^4 - 3*A*b^2*c^2*d^4 - 6*a*b*B*c^2*d^4 - 3*a^2*c^2*C*d^4 +
6*b^2*c^2*C*d^4 + 6*a*A*b*c*d^5 + 3*a^2*B*c*d^5 - 3*b^2*B*c*d^5 - 6*a*b*c*C
*d^5 - a^2*A*d^6 + A*b^2*d^6 + 2*a*b*B*d^6 + a^2*C*d^6)*ArcTan[Tan[e + f*x]
]*Sec[e + f*x]*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^2)/
(d^3*(c^2 + d^2)^3*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x
])^3) - (b^2*C*Log[Cos[e + f*x]]*Sec[e + f*x]*(c*Cos[e + f*x] + d*Sin[e + f
x])^3(a + b*Tan[e + f*x])^2)/(d^3*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(
c + d*Tan[e + f*x])^3) + ((b^2*c^6*C + 3*b^2*c^4*C*d^2 - 2*a*A*b*c^3*d^3 -
a^2*B*c^3*d^3 + b^2*B*c^3*d^3 + 2*a*b*c^3*C*d^3 + 3*a^2*A*c^2*d^4 - 3*A*b^2
*c^2*d^4 - 6*a*b*B*c^2*d^4 - 3*a^2*c^2*C*d^4 + 6*b^2*c^2*C*d^4 + 6*a*A*b*c*
d^5 + 3*a^2*B*c*d^5 - 3*b^2*B*c*d^5 - 6*a*b*c*C*d^5 - a^2*A*d^6 + A*b^2*d^6
 + 2*a*b*B*d^6 + a^2*C*d^6)*Log[(c*Cos[e + f*x] + d*Sin[e + f*x])^2]*Sec[e
+ f*x]*(c*Cos[e + f*x] + d*Sin[e + f*x])^3*(a + b*Tan[e + f*x])^2)/(2*d^3*(
c^2 + d^2)^3*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e + f*x])^3)
+ (Sec[e + f*x]*(c*Cos[e + f*x] + d*Sin[e + f*x])^2*(-(b^2*c^5*C*Sin[e + f*
x]) + A*b^2*c^3*d^2*Sin[e + f*x] + 2*a*b*B*c^3*d^2*Sin[e + f*x] + a^2*c^3*C
*d^2*Sin[e + f*x] - 4*b^2*c^3*C*d^2*Sin[e + f*x] - 4*a*A*b*c^2*d^3*Sin[e +
f*x] - 2*a^2*B*c^2*d^3*Sin[e + f*x] + 3*b^2*B*c^2*d^3*Sin[e + f*x] + 6*a*b*

```
c^2*C*d^3*Sin[e + f*x] + 3*a^2*A*c*d^4*Sin[e + f*x] - 2*A*b^2*c*d^4*Sin[e +
 f*x] - 4*a*b*B*c*d^4*Sin[e + f*x] - 2*a^2*c*C*d^4*Sin[e + f*x] + 2*a*A*b*d
^5*Sin[e + f*x] + a^2*B*d^5*Sin[e + f*x])*(a + b*Tan[e + f*x])^2)/(c*(c - I
*d)^2*(c + I*d)^2*d^2*f*(a*Cos[e + f*x] + b*Sin[e + f*x])^2*(c + d*Tan[e +
f*x])^3)
```

fricas [B] time = 1.23, size = 1618, normalized size = 2.71

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] 1/2*(C*b^2*c^6*d^2 - A*a^2*d^8 + (2*C*a*b + B*b^2)*c^5*d^3 - (3*C*a^2 + 6*B
*a*b + (3*A - 7*C)*b^2)*c^4*d^4 + 5*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^3*d^5
 - ((7*A - 3*C)*a^2 - 6*B*a*b - 3*A*b^2)*c^2*d^6 - (B*a^2 + 2*A*a*b)*c*d^7
+ 2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^5*d^3 + 3*(B*a^2 + 2*(A - C)*a
*b - B*b^2)*c^4*d^4 - 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^3*d^5 - (B*
a^2 + 2*(A - C)*a*b - B*b^2)*c^2*d^6)*f*x - (3*C*b^2*c^6*d^2 + A*a^2*d^8 -
(2*C*a*b + B*b^2)*c^5*d^3 - (C*a^2 + 2*B*a*b + (A - 9*C)*b^2)*c^4*d^4 + (3*
B*a^2 + 2*(3*A - 7*C)*a*b - 7*B*b^2)*c^3*d^5 - 5*((A - C)*a^2 - 2*B*a*b - A
*b^2)*c^2*d^6 - 3*(B*a^2 + 2*A*a*b)*c*d^7 - 2*(((A - C)*a^2 - 2*B*a*b - (A
- C)*b^2)*c^3*d^5 + 3*(B*a^2 + 2*(A - C)*a*b - B*b^2)*c^2*d^6 - 3*((A - C)*
a^2 - 2*B*a*b - (A - C)*b^2)*c*d^7 - (B*a^2 + 2*(A - C)*a*b - B*b^2)*d^8)*f
*x)*tan(f*x + e)^2 + (C*b^2*c^8 + 3*C*b^2*c^6*d^2 - (B*a^2 + 2*(A - C)*a*b
- B*b^2)*c^5*d^3 + 3*((A - C)*a^2 - 2*B*a*b - (A - 2*C)*b^2)*c^4*d^4 + 3*(B
*a^2 + 2*(A - C)*a*b - B*b^2)*c^3*d^5 - ((A - C)*a^2 - 2*B*a*b - A*b^2)*c^2
*d^6 + (C*b^2*c^6*d^2 + 3*C*b^2*c^4*d^4 - (B*a^2 + 2*(A - C)*a*b - B*b^2)*c
^3*d^5 + 3*((A - C)*a^2 - 2*B*a*b - (A - 2*C)*b^2)*c^2*d^6 + 3*(B*a^2 + 2*(
A - C)*a*b - B*b^2)*c*d^7 - ((A - C)*a^2 - 2*B*a*b - A*b^2)*d^8)*tan(f*x +
e)^2 + 2*(C*b^2*c^7*d + 3*C*b^2*c^5*d^3 - (B*a^2 + 2*(A - C)*a*b - B*b^2)*c
^4*d^4 + 3*((A - C)*a^2 - 2*B*a*b - (A - 2*C)*b^2)*c^3*d^5 + 3*(B*a^2 + 2*(
A - C)*a*b - B*b^2)*c^2*d^6 - ((A - C)*a^2 - 2*B*a*b - A*b^2)*c*d^7)*tan(f*
x + e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2
 + 1)) - (C*b^2*c^8 + 3*C*b^2*c^6*d^2 + 3*C*b^2*c^4*d^4 + C*b^2*c^2*d^6 + (
C*b^2*c^6*d^2 + 3*C*b^2*c^4*d^4 + 3*C*b^2*c^2*d^6 + C*b^2*d^8)*tan(f*x + e)
^2 + 2*(C*b^2*c^7*d + 3*C*b^2*c^5*d^3 + 3*C*b^2*c^3*d^5 + C*b^2*c*d^7)*tan(
f*x + e))*log(1/(tan(f*x + e)^2 + 1)) - 2*(C*b^2*c^7*d - (C*a^2 + 2*B*a*b +
 (A - 3*C)*b^2)*c^5*d^3 + (2*B*a^2 + 2*(2*A - 3*C)*a*b - 3*B*b^2)*c^4*d^4 -
 (3*(A - C)*a^2 - 6*B*a*b - (3*A - 4*C)*b^2)*c^3*d^5 - 3*(B*a^2 + 2*(A - C)
*a*b - B*b^2)*c^2*d^6 + ((3*A - 2*C)*a^2 - 4*B*a*b - 2*A*b^2)*c*d^7 + (B*a^
2 + 2*A*a*b)*d^8 - 2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^4*d^4 + 3*(B*
a^2 + 2*(A - C)*a*b - B*b^2)*c^3*d^5 - 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b
^2)*c^2*d^6 - (B*a^2 + 2*(A - C)*a*b - B*b^2)*c*d^7)*f*x)*tan(f*x + e))/((c
```

```
^6*d^5 + 3*c^4*d^7 + 3*c^2*d^9 + d^11)*f*tan(f*x + e)^2 + 2*(c^7*d^4 + 3*c^
5*d^6 + 3*c^3*d^8 + c*d^10)*f*tan(f*x + e) + (c^8*d^3 + 3*c^6*d^5 + 3*c^4*d
^7 + c^2*d^9)*f)
```

giac [B] time = 4.65, size = 1709, normalized size = 2.86

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c^3 - C*a^2*c^3 - 2*B*a*b*c^3 - A*b^2*c^3 + C*b^2*c^3 + 3*B*a
^2*c^2*d + 6*A*a*b*c^2*d - 6*C*a*b*c^2*d - 3*B*b^2*c^2*d - 3*A*a^2*c*d^2 +
3*C*a^2*c*d^2 + 6*B*a*b*c*d^2 + 3*A*b^2*c*d^2 - 3*C*b^2*c*d^2 - B*a^2*d^3 -
 2*A*a*b*d^3 + 2*C*a*b*d^3 + B*b^2*d^3)*(f*x + e)/(c^6 + 3*c^4*d^2 + 3*c^2*
d^4 + d^6) + (B*a^2*c^3 + 2*A*a*b*c^3 - 2*C*a*b*c^3 - B*b^2*c^3 - 3*A*a^2*c
^2*d + 3*C*a^2*c^2*d + 6*B*a*b*c^2*d + 3*A*b^2*c^2*d - 3*C*b^2*c^2*d - 3*B*
a^2*c*d^2 - 6*A*a*b*c*d^2 + 6*C*a*b*c*d^2 + 3*B*b^2*c*d^2 + A*a^2*d^3 - C*a
^2*d^3 - 2*B*a*b*d^3 - A*b^2*d^3 + C*b^2*d^3)*log(tan(f*x + e)^2 + 1)/(c^6
+ 3*c^4*d^2 + 3*c^2*d^4 + d^6) + 2*(C*b^2*c^6 + 3*C*b^2*c^4*d^2 - B*a^2*c^3
*d^3 - 2*A*a*b*c^3*d^3 + 2*C*a*b*c^3*d^3 + B*b^2*c^3*d^3 + 3*A*a^2*c^2*d^4
- 3*C*a^2*c^2*d^4 - 6*B*a*b*c^2*d^4 - 3*A*b^2*c^2*d^4 + 6*C*b^2*c^2*d^4 + 3
*B*a^2*c*d^5 + 6*A*a*b*c*d^5 - 6*C*a*b*c*d^5 - 3*B*b^2*c*d^5 - A*a^2*d^6 +
C*a^2*d^6 + 2*B*a*b*d^6 + A*b^2*d^6)*log(abs(d*tan(f*x + e) + c))/(c^6*d^3
+ 3*c^4*d^5 + 3*c^2*d^7 + d^9) - (3*C*b^2*c^6*d*tan(f*x + e)^2 + 9*C*b^2*c^
4*d^3*tan(f*x + e)^2 - 3*B*a^2*c^3*d^4*tan(f*x + e)^2 - 6*A*a*b*c^3*d^4*tan
(f*x + e)^2 + 6*C*a*b*c^3*d^4*tan(f*x + e)^2 + 3*B*b^2*c^3*d^4*tan(f*x + e)
^2 + 9*A*a^2*c^2*d^5*tan(f*x + e)^2 - 9*C*a^2*c^2*d^5*tan(f*x + e)^2 - 18*B
*a*b*c^2*d^5*tan(f*x + e)^2 - 9*A*b^2*c^2*d^5*tan(f*x + e)^2 + 18*C*b^2*c^2
*d^5*tan(f*x + e)^2 + 9*B*a^2*c*d^6*tan(f*x + e)^2 + 18*A*a*b*c*d^6*tan(f*x
 + e)^2 - 18*C*a*b*c*d^6*tan(f*x + e)^2 - 9*B*b^2*c*d^6*tan(f*x + e)^2 - 3*
A*a^2*d^7*tan(f*x + e)^2 + 3*C*a^2*d^7*tan(f*x + e)^2 + 6*B*a*b*d^7*tan(f*x
 + e)^2 + 3*A*b^2*d^7*tan(f*x + e)^2 + 2*C*b^2*c^7*tan(f*x + e) + 4*C*a*b*c
^6*d*tan(f*x + e) + 2*B*b^2*c^6*d*tan(f*x + e) + 6*C*b^2*c^5*d^2*tan(f*x +
e) - 8*B*a^2*c^4*d^3*tan(f*x + e) - 16*A*a*b*c^4*d^3*tan(f*x + e) + 28*C*a*
b*c^4*d^3*tan(f*x + e) + 14*B*b^2*c^4*d^3*tan(f*x + e) + 22*A*a^2*c^3*d^4*t
an(f*x + e) - 22*C*a^2*c^3*d^4*tan(f*x + e) - 44*B*a*b*c^3*d^4*tan(f*x + e)
 - 22*A*b^2*c^3*d^4*tan(f*x + e) + 28*C*b^2*c^3*d^4*tan(f*x + e) + 18*B*a^2
*c^2*d^5*tan(f*x + e) + 36*A*a*b*c^2*d^5*tan(f*x + e) - 24*C*a*b*c^2*d^5*ta
n(f*x + e) - 12*B*b^2*c^2*d^5*tan(f*x + e) - 2*A*a^2*c*d^6*tan(f*x + e) + 2
*C*a^2*c*d^6*tan(f*x + e) + 4*B*a*b*c*d^6*tan(f*x + e) + 2*A*b^2*c*d^6*tan(
f*x + e) + 2*B*a^2*d^7*tan(f*x + e) + 4*A*a*b*d^7*tan(f*x + e) + 2*C*a*b*c^
7 + B*b^2*c^7 + C*a^2*c^6*d + 2*B*a*b*c^6*d + A*b^2*c^6*d - C*b^2*c^6*d - 6
*B*a^2*c^5*d^2 - 12*A*a*b*c^5*d^2 + 18*C*a*b*c^5*d^2 + 9*B*b^2*c^5*d^2 + 14
```

```
*A*a^2*c^4*d^3 - 11*C*a^2*c^4*d^3 - 22*B*a*b*c^4*d^3 - 11*A*b^2*c^4*d^3 + 1
1*C*b^2*c^4*d^3 + 7*B*a^2*c^3*d^4 + 14*A*a*b*c^3*d^4 - 8*C*a*b*c^3*d^4 - 4*
B*b^2*c^3*d^4 + 3*A*a^2*c^2*d^5 + B*a^2*c*d^6 + 2*A*a*b*c*d^6 + A*a^2*d^7)/
((c^6*d^2 + 3*c^4*d^4 + 3*c^2*d^6 + d^8)*(d*tan(f*x + e) + c)^2))/f
```

maple [B] time = 0.32, size = 2465, normalized size = 4.13

Expression too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x
)
```

```
[Out] 1/f/d^2/(c^2+d^2)/(c+d*tan(f*x+e))^2*C*a*b*c^3-1/f/d/(c^2+d^2)/(c+d*tan(f*x
+e))^2*B*c^2*a*b+4/f*d/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*a*b*c-2/f/d^2/(c^2+d^
2)^2/(c+d*tan(f*x+e))*C*a*b*c^4+6/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*a*b*c*
d^2-6/f/(c^2+d^2)^3*C*arctan(tan(f*x+e))*a*b*c^2*d-3/f/(c^2+d^2)^3*ln(1+tan
(f*x+e)^2)*A*a*b*c*d^2+3/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a*b*c^2*d+3/f/(
c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a*b*c*d^2+6/f/(c^2+d^2)^3*A*arctan(tan(f*x+
e))*a*b*c^2*d+6/f/(c^2+d^2)^3*d^2*ln(c+d*tan(f*x+e))*A*a*b*c-6/f/(c^2+d^2)^
3*d*ln(c+d*tan(f*x+e))*B*a*b*c^2-6/f/(c^2+d^2)^3*d^2*ln(c+d*tan(f*x+e))*C*a
*b*c-2/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*a*b*c^3+2/f/(c^2+d^2)^2/(c+d*tan(
f*x+e))*A*a*b*c^2-6/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*C*a*b*c^2+1/f/(c^2+d^2)/
(c+d*tan(f*x+e))^2*A*a*c*b-1/f/d^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*b^2*c^4+2
/f*d/(c^2+d^2)^2/(c+d*tan(f*x+e))*C*a^2*c-1/f/(c^2+d^2)^3*B*arctan(tan(f*x+
e))*a^2*d^3+1/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*b^2*d^3-1/f/(c^2+d^2)^3*C*
arctan(tan(f*x+e))*a^2*c^3+1/f/(c^2+d^2)^3*C*arctan(tan(f*x+e))*b^2*c^3+1/2
/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*a^2*d^3-1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+
e)^2)*A*b^2*d^3+1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a^2*c^3-1/2/f/(c^2+d
^2)^3*ln(1+tan(f*x+e)^2)*B*b^2*c^3-1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a
^2*d^3+1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*b^2*d^3-1/2/f*d/(c^2+d^2)/(c+
d*tan(f*x+e))^2*a^2*A-1/f*d^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*a^2-1/f/(c^2+d
^2)^3*d^3*ln(c+d*tan(f*x+e))*A*a^2+1/f/(c^2+d^2)^3*d^3*ln(c+d*tan(f*x+e))*A
*b^2+1/f/(c^2+d^2)^3*d^3*ln(c+d*tan(f*x+e))*C*a^2-1/f/(c^2+d^2)^3*ln(c+d*ta
n(f*x+e))*B*a^2*c^3+1/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*b^2*c^3+1/f/(c^2+d
^2)^2/(c+d*tan(f*x+e))*B*a^2*c^2-3/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*b^2*c^2
+1/2/f/(c^2+d^2)/(c+d*tan(f*x+e))^2*B*a^2*c+1/f/(c^2+d^2)^3*A*arctan(tan(f*
x+e))*a^2*c^3-1/f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*b^2*c^3-3/2/f/(c^2+d^2)^
3*ln(1+tan(f*x+e)^2)*A*a^2*c^2*d+1/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*a*b*c
^3+3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*b^2*c^2*d-1/2/f/d/(c^2+d^2)/(c+d*
tan(f*x+e))^2*A*c^2*b^2+2/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*a*b*c^3+2/f/d^
3/(c^2+d^2)^2/(c+d*tan(f*x+e))*C*b^2*c^5+4/f/d/(c^2+d^2)^2/(c+d*tan(f*x+e))
*C*b^2*c^3+1/2/f/d^2/(c^2+d^2)/(c+d*tan(f*x+e))^2*B*b^2*c^3+3/f/(c^2+d^2)^3
*d*ln(c+d*tan(f*x+e))*A*a^2*c^2-3/f/(c^2+d^2)^3*d*ln(c+d*tan(f*x+e))*A*b^2*
c^2+3/f/(c^2+d^2)^3*d^2*ln(c+d*tan(f*x+e))*B*a^2*c+2/f/(c^2+d^2)^3*d^3*ln(c
```

```
+d*tan(f*x+e))*B*a*b-3/f/(c^2+d^2)^3*d^2*ln(c+d*tan(f*x+e))*B*b^2*c-3/f/(c^
2+d^2)^3*d*ln(c+d*tan(f*x+e))*C*a^2*c^2+1/f/(c^2+d^2)^3/d^3*ln(c+d*tan(f*x+
e))*C*b^2*c^6+3/f/(c^2+d^2)^3/d*ln(c+d*tan(f*x+e))*C*b^2*c^4-3/2/f/(c^2+d^2
)^3*ln(1+tan(f*x+e)^2)*B*a^2*c*d^2-1/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a*b
*d^3-1/2/f/d/(c^2+d^2)/(c+d*tan(f*x+e))^2*C*c^2*a^2-1/2/f/d^3/(c^2+d^2)/(c+
d*tan(f*x+e))^2*b^2*C*c^4-3/f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*a^2*c*d^2-2/
f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*a*b*d^3+3/f/(c^2+d^2)^3*A*arctan(tan(f*x
+e))*b^2*c*d^2+3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a^2*c^2*d-1/f/(c^2+d^
2)^3*ln(1+tan(f*x+e)^2)*C*a*b*c^3-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*b^
2*c^2*d+3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*b^2*c*d^2-2/f*d^2/(c^2+d^2)^
2/(c+d*tan(f*x+e))*A*a*b+2/f*d/(c^2+d^2)^2/(c+d*tan(f*x+e))*A*b^2*c+3/f/(c^
2+d^2)^3*C*arctan(tan(f*x+e))*a^2*c*d^2+2/f/(c^2+d^2)^3*C*arctan(tan(f*x+e)
)*a*b*d^3-3/f/(c^2+d^2)^3*C*arctan(tan(f*x+e))*b^2*c*d^2+3/f/(c^2+d^2)^3*B*
arctan(tan(f*x+e))*a^2*c^2*d-2/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*a*b*c^3-3
/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*b^2*c^2*d+6/f/(c^2+d^2)^3*d*ln(c+d*tan(
f*x+e))*C*b^2*c^2-2/f*d/(c^2+d^2)^2/(c+d*tan(f*x+e))*A*a^2*c
```

maxima [A] time = 0.58, size = 827, normalized size = 1.39

$$\frac{\frac{2\left(\left((A-C)a^2-2\,Bab-(A-C)b^2\right)c^3+3\left(Ba^2+2\,(A-C)ab-Bb^2\right)c^2d-3\left((A-C)a^2-2\,Bab-(A-C)b^2\right)cd^2-\left(Ba^2+2\,(A-C)ab-Bb^2\right)d^3\right)(fx+e)}{c^6+3\,c^4d^2+3\,c^2d^4+d^6}+\frac{2\left(Cb^2c^6+3\,Cb\right.}{}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^3 + 3*(B*a^2 + 2*(A - C)*a*
b - B*b^2)*c^2*d - 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c*d^2 - (B*a^2 +
 2*(A - C)*a*b - B*b^2)*d^3)*(f*x + e)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6)
+ 2*(C*b^2*c^6 + 3*C*b^2*c^4*d^2 - (B*a^2 + 2*(A - C)*a*b - B*b^2)*c^3*d^3
+ 3*((A - C)*a^2 - 2*B*a*b - (A - 2*C)*b^2)*c^2*d^4 + 3*(B*a^2 + 2*(A - C)*
a*b - B*b^2)*c*d^5 - ((A - C)*a^2 - 2*B*a*b - A*b^2)*d^6)*log(d*tan(f*x + e
) + c)/(c^6*d^3 + 3*c^4*d^5 + 3*c^2*d^7 + d^9) + ((B*a^2 + 2*(A - C)*a*b -
B*b^2)*c^3 - 3*((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*c^2*d - 3*(B*a^2 + 2*(
A - C)*a*b - B*b^2)*c*d^2 + ((A - C)*a^2 - 2*B*a*b - (A - C)*b^2)*d^3)*log(
tan(f*x + e)^2 + 1)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) + (3*C*b^2*c^6 - A*
a^2*d^6 - (2*C*a*b + B*b^2)*c^5*d - (C*a^2 + 2*B*a*b + (A - 7*C)*b^2)*c^4*d
^2 + (3*B*a^2 + 2*(3*A - 5*C)*a*b - 5*B*b^2)*c^3*d^3 - ((5*A - 3*C)*a^2 - 6
*B*a*b - 3*A*b^2)*c^2*d^4 - (B*a^2 + 2*A*a*b)*c*d^5 + 2*(2*C*b^2*c^5*d + 4*
C*b^2*c^3*d^3 - (2*C*a*b + B*b^2)*c^4*d^2 + (B*a^2 + 2*(A - 3*C)*a*b - 3*B*
b^2)*c^2*d^4 - 2*((A - C)*a^2 - 2*B*a*b - A*b^2)*c*d^5 - (B*a^2 + 2*A*a*b)*
d^6)*tan(f*x + e))/(c^6*d^3 + 2*c^4*d^5 + c^2*d^7 + (c^4*d^5 + 2*c^2*d^7 +
d^9)*tan(f*x + e)^2 + 2*(c^5*d^4 + 2*c^3*d^6 + c*d^8)*tan(f*x + e)))/f
```

mupad [B] time = 30.69, size = 807, normalized size = 1.35

$$-\frac{\frac{A a^2 d^6-3 C b^2 c^6+B a^2 c d^5+B b^2 c^5 d+5 A a^2 c^2 d^4-3 A b^2 c^2 d^4+A b^2 c^4 d^2-3 B a^2 c^3 d^3+5 B b^2 c^3 d^3-3 C a^2 c^2 d^4+C a^2 c^4 d^2-7 C b^2 c^4 d^2+2 A a b c d}{2 d^3 \left(c^4+2 c^2 d^2+d^4\right)}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^3,x)
```

```
[Out] - ((A*a^2*d^6 - 3*C*b^2*c^6 + B*a^2*c*d^5 + B*b^2*c^5*d + 5*A*a^2*c^2*d^4 -
 3*A*b^2*c^2*d^4 + A*b^2*c^4*d^2 - 3*B*a^2*c^3*d^3 + 5*B*b^2*c^3*d^3 - 3*C*
a^2*c^2*d^4 + C*a^2*c^4*d^2 - 7*C*b^2*c^4*d^2 + 2*A*a*b*c*d^5 + 2*C*a*b*c^5
*d - 6*A*a*b*c^3*d^3 - 6*B*a*b*c^2*d^4 + 2*B*a*b*c^4*d^2 + 10*C*a*b*c^3*d^3
)/(2*d^3*(c^4 + d^4 + 2*c^2*d^2)) + (tan(e + f*x)*(B*a^2*d^5 - 2*C*b^2*c^5
+ 2*A*a*b*d^5 + 2*A*a^2*c*d^4 - 2*A*b^2*c*d^4 + B*b^2*c^4*d - 2*C*a^2*c*d^4
 - B*a^2*c^2*d^3 + 3*B*b^2*c^2*d^3 - 4*C*b^2*c^3*d^2 - 4*B*a*b*c*d^4 + 2*C*
a*b*c^4*d - 2*A*a*b*c^2*d^3 + 6*C*a*b*c^2*d^3))/(d^2*(c^4 + d^4 + 2*c^2*d^2
)))/(f*(c^2 + d^2*tan(e + f*x)^2 + 2*c*d*tan(e + f*x))) - (log(c + d*tan(e
+ f*x))*((c^2*(d^4*(3*A*b^2 - 3*A*a^2 + 3*C*a^2 - 6*C*b^2 + 6*B*a*b) + 3*C*
b^2*d^4) - d^6*(A*b^2 - A*a^2 + C*a^2 + 2*B*a*b) + C*b^2*d^6 - c*d^5*(3*B*a
^2 - 3*B*b^2 + 6*A*a*b - 6*C*a*b) + c^3*d^3*(B*a^2 - B*b^2 + 2*A*a*b - 2*C*
a*b))/(d^9 + 3*c^2*d^7 + 3*c^4*d^5 + c^6*d^3) - (C*b^2)/d^3))/f - (log(tan(
e + f*x) - 1i)*(A*b^2*1i - A*a^2*1i + B*a^2 - B*b^2 + C*a^2*1i - C*b^2*1i +
 2*A*a*b + B*a*b*2i - 2*C*a*b))/(2*f*(3*c*d^2 - c^2*d*3i - c^3 + d^3*1i)) -
 (log(tan(e + f*x) + 1i)*(A*b^2 - A*a^2 + B*a^2*1i - B*b^2*1i + C*a^2 - C*b
^2 + A*a*b*2i + 2*B*a*b - C*a*b*2i))/(2*f*(c*d^2*3i - 3*c^2*d - c^3*1i + d^
3))
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**3,x)
```

```
[Out] Exception raised: AttributeError
```

3.86 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$

Optimal. Leaf size=352

$$\frac{(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{2d^2f\left(c^2+d^2\right)(c+d\tan(e+fx))^2}-\frac{ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c^4C\right)}{d^2f\left(c^2+d^2\right)^2(c+d\tan(e+fx))}+$$

```
[Out] -(a*(c^3*C-3*B*c^2*d-3*c*C*d^2+B*d^3-A*(c^3-3*c*d^2))-b*((A-C)*d*(3*c^2-d^2
)-B*(c^3-3*c*d^2)))*x/(c^2+d^2)^3+(b*(-3*B*c^2*d+B*d^3+C*c^3-3*C*c*d^2)-a*(
B*c^3-3*B*c*d^2+3*C*c^2*d-C*d^3)+A*(a*d*(3*c^2-d^2)-b*(c^3-3*c*d^2)))*ln(c*
cos(f*x+e)+d*sin(f*x+e))/(c^2+d^2)^3/f+1/2*(-a*d+b*c)*(A*d^2-B*c*d+C*c^2)/d
^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^2+(-b*(c^4*C-c^2*(A-3*C)*d^2-2*B*c*d^3+A*d^
4)-a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))/d^2/(c^2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] time = 0.71, antiderivative size = 349, normalized size of antiderivative = 0.99, number of steps used = 4, number of rules used = 4, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.093, Rules used = {3635, 3628, 3531, 3530}

$$\frac{(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{2d^2f\left(c^2+d^2\right)(c+d\tan(e+fx))^2}-\frac{ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c^4C\right)}{d^2f\left(c^2+d^2\right)^2(c+d\tan(e+fx))}+$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d*T
an[e + f*x])^3,x]
```

```
[Out] ((b*(A - C)*d*(3*c^2 - d^2) - b*B*(c^3 - 3*c*d^2) - a*(c^3*C - 3*B*c^2*d -
3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)))*x)/(c^2 + d^2)^3 + ((a*A*d*(3*c^2 -
 d^2) - A*b*(c^3 - 3*c*d^2) + b*(c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3) - a
*(B*c^3 + 3*c^2*C*d - 3*B*c*d^2 - C*d^3))*Log[c*Cos[e + f*x] + d*Sin[e + f*
x]])/((c^2 + d^2)^3*f) + ((b*c - a*d)*(c^2*C - B*c*d + A*d^2))/(2*d^2*(c^2
+ d^2)*f*(c + d*Tan[e + f*x])^2) - (b*(c^4*C - c^2*(A - 3*C)*d^2 - 2*B*c*d^
3 + A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))/(d^2*(c^2 + d^2)^2*f*(c
 + d*Tan[e + f*x]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx = \frac{(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{2d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} + \int \frac{ad(Ac-\cdots}{\cdots}$$

$$= \frac{(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{2d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} - \frac{b\left(c^4C-\cdots\right.}{\cdots}$$

$$= \frac{\left(b(A-C)d\left(3c^2-d^2\right)-bB\left(c^3-3cd^2\right)-a\left(c^3\cdots\right.\right.}{\left(c^2+\cdots\right.}$$

$$= \frac{\left(b(A-C)d\left(3c^2-d^2\right)-bB\left(c^3-3cd^2\right)-a\left(c^3\cdots\right.\right.}{\left(c^2+\cdots\right.}$$

Mathematica [C] time = 6.34, size = 378, normalized size = 1.07

$$-\frac{C(a+b\tan(e+fx))}{df(c+d\tan(e+fx))^2} - \frac{\frac{-aCd+bBd+bcC}{2df(c+d\tan(e+fx))^2} + \frac{\frac{\left(2cd^2(aB+Ab-bC)+2d^3(bB-a(A-C))\right)\left(-\frac{2cd}{\left(c^2+d^2\right)^2(c+d\tan(e+fx))} - \frac{d}{2\left(c^2+d^2\right)(c+d\tan(e+fx))^2} + \frac{d\left(3c^2-d^2\right)\cdots}{\cdots}\right.}{d}}{\cdots}}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^3,x]
```

```
[Out] -((C*(a + b*Tan[e + f*x]))/(d*f*(c + d*Tan[e + f*x])^2)) - ((b*c*C + b*B*d
- a*C*d)/(2*d*f*(c + d*Tan[e + f*x])^2) + (((2*c*(A*b + a*B - b*C)*d^2 + 2*
(b*B - a*(A - C))*d^3)*(-1/2*Log[I - Tan[e + f*x]]/(I*c - d)^3 + Log[I + Ta
n[e + f*x]]/(2*(I*c + d)^3) + (d*(3*c^2 - d^2)*Log[c + d*Tan[e + f*x]])/(c^
2 + d^2)^3 - d/(2*(c^2 + d^2)*(c + d*Tan[e + f*x])^2) - (2*c*d)/((c^2 + d^2
)^2*(c + d*Tan[e + f*x]))))/d - 2*(A*b + a*B - b*C)*d*(((-1/2*I)*Log[I - Ta
n[e + f*x]])/(c + I*d)^2 + ((I/2)*Log[I + Tan[e + f*x]])/(c - I*d)^2 + (2*c
*d*Log[c + d*Tan[e + f*x]])/(c^2 + d^2)^2 - d/((c^2 + d^2)*(c + d*Tan[e + f
*x]))))/(2*d*f))/d
```

fricas [B] time = 0.54, size = 897, normalized size = 2.55

$$\frac{Cbc^5 - Aad^5 - 3\,(Ca+Bb)c^4d + 5\,(Ba+(A-C)b)c^3d^2 - ((7\,A-3\,C)a-3\,Bb)c^2d^3 - (Ba+Ab)cd^4 + 2\left(((A-C\cdots\right.}{\cdots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^3,x, algorithm="fricas")
```

```
[Out] 1/2*(C*b*c^5 - A*a*d^5 - 3*(C*a + B*b)*c^4*d + 5*(B*a + (A - C)*b)*c^3*d^2
- ((7*A - 3*C)*a - 3*B*b)*c^2*d^3 - (B*a + A*b)*c*d^4 + 2*(((A - C)*a - B*b
)*c^5 + 3*(B*a + (A - C)*b)*c^4*d - 3*((A - C)*a - B*b)*c^3*d^2 - (B*a + (A
 - C)*b)*c^2*d^3)*f*x + (C*b*c^5 - A*a*d^5 + (C*a + B*b)*c^4*d - (3*B*a + (
3*A - 7*C)*b)*c^3*d^2 + 5*((A - C)*a - B*b)*c^2*d^3 + 3*(B*a + A*b)*c*d^4 +
 2*(((A - C)*a - B*b)*c^3*d^2 + 3*(B*a + (A - C)*b)*c^2*d^3 - 3*((A - C)*a
- B*b)*c*d^4 - (B*a + (A - C)*b)*d^5)*f*x)*tan(f*x + e)^2 - ((B*a + (A - C)
*b)*c^5 - 3*((A - C)*a - B*b)*c^4*d - 3*(B*a + (A - C)*b)*c^3*d^2 + ((A - C
)*a - B*b)*c^2*d^3 + ((B*a + (A - C)*b)*c^3*d^2 - 3*((A - C)*a - B*b)*c^2*d
^3 - 3*(B*a + (A - C)*b)*c*d^4 + ((A - C)*a - B*b)*d^5)*tan(f*x + e)^2 + 2*
((B*a + (A - C)*b)*c^4*d - 3*((A - C)*a - B*b)*c^3*d^2 - 3*(B*a + (A - C)*b
)*c^2*d^3 + ((A - C)*a - B*b)*c*d^4)*tan(f*x + e))*log((d^2*tan(f*x + e)^2
+ 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) + 2*((C*a + B*b)*c^5 - (2
*B*a + (2*A - 3*C)*b)*c^4*d + 3*((A - C)*a - B*b)*c^3*d^2 + 3*(B*a + (A - C
)*b)*c^2*d^3 - ((3*A - 2*C)*a - 2*B*b)*c*d^4 - (B*a + A*b)*d^5 + 2*(((A - C
)*a - B*b)*c^4*d + 3*(B*a + (A - C)*b)*c^3*d^2 - 3*((A - C)*a - B*b)*c^2*d^
3 - (B*a + (A - C)*b)*c*d^4)*f*x)*tan(f*x + e))/((c^6*d^2 + 3*c^4*d^4 + 3*c
^2*d^6 + d^8)*f*tan(f*x + e)^2 + 2*(c^7*d + 3*c^5*d^3 + 3*c^3*d^5 + c*d^7)*
f*tan(f*x + e) + (c^8 + 3*c^6*d^2 + 3*c^4*d^4 + c^2*d^6)*f)
```

giac [B] time = 2.70, size = 1037, normalized size = 2.95

$$\frac{\frac{2\left(Aac^3-Cac^3-Bbc^3+3\,Bac^2d+3\,Abc^2d-3\,Cbc^2d-3\,Aacd^2+3\,Cacd^2+3\,Bbcd^2-Bad^3-Abd^3+Cbd^3\right)\left(fx+e\right)}{c^6+3\,c^4d^2+3\,c^2d^4+d^6}+\frac{\left(Bac^3+Abc^3-Cbc^3-3\,Aac^2d+3\,Cac^2d+3\cdots\right.}{\cdots}}{\cdots}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a*c^3 - C*a*c^3 - B*b*c^3 + 3*B*a*c^2*d + 3*A*b*c^2*d - 3*C*b*c^2
*d - 3*A*a*c*d^2 + 3*C*a*c*d^2 + 3*B*b*c*d^2 - B*a*d^3 - A*b*d^3 + C*b*d^3)
*(f*x + e)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) + (B*a*c^3 + A*b*c^3 - C*b*c
^3 - 3*A*a*c^2*d + 3*C*a*c^2*d + 3*B*b*c^2*d - 3*B*a*c*d^2 - 3*A*b*c*d^2 +
3*C*b*c*d^2 + A*a*d^3 - C*a*d^3 - B*b*d^3)*log(tan(f*x + e)^2 + 1)/(c^6 + 3
*c^4*d^2 + 3*c^2*d^4 + d^6) - 2*(B*a*c^3*d + A*b*c^3*d - C*b*c^3*d - 3*A*a*
c^2*d^2 + 3*C*a*c^2*d^2 + 3*B*b*c^2*d^2 - 3*B*a*c*d^3 - 3*A*b*c*d^3 + 3*C*b
*c*d^3 + A*a*d^4 - C*a*d^4 - B*b*d^4)*log(abs(d*tan(f*x + e) + c))/(c^6*d +
 3*c^4*d^3 + 3*c^2*d^5 + d^7) + (3*B*a*c^3*d^4*tan(f*x + e)^2 + 3*A*b*c^3*d
```

```
^4*tan(f*x + e)^2 - 3*C*b*c^3*d^4*tan(f*x + e)^2 - 9*A*a*c^2*d^5*tan(f*x +
e)^2 + 9*C*a*c^2*d^5*tan(f*x + e)^2 + 9*B*b*c^2*d^5*tan(f*x + e)^2 - 9*B*a*
c*d^6*tan(f*x + e)^2 - 9*A*b*c*d^6*tan(f*x + e)^2 + 9*C*b*c*d^6*tan(f*x + e
)^2 + 3*A*a*d^7*tan(f*x + e)^2 - 3*C*a*d^7*tan(f*x + e)^2 - 3*B*b*d^7*tan(f
*x + e)^2 - 2*C*b*c^6*d*tan(f*x + e) + 8*B*a*c^4*d^3*tan(f*x + e) + 8*A*b*c
^4*d^3*tan(f*x + e) - 14*C*b*c^4*d^3*tan(f*x + e) - 22*A*a*c^3*d^4*tan(f*x
+ e) + 22*C*a*c^3*d^4*tan(f*x + e) + 22*B*b*c^3*d^4*tan(f*x + e) - 18*B*a*c
^2*d^5*tan(f*x + e) - 18*A*b*c^2*d^5*tan(f*x + e) + 12*C*b*c^2*d^5*tan(f*x
+ e) + 2*A*a*c*d^6*tan(f*x + e) - 2*C*a*c*d^6*tan(f*x + e) - 2*B*b*c*d^6*ta
n(f*x + e) - 2*B*a*d^7*tan(f*x + e) - 2*A*b*d^7*tan(f*x + e) - C*b*c^7 - C*
a*c^6*d - B*b*c^6*d + 6*B*a*c^5*d^2 + 6*A*b*c^5*d^2 - 9*C*b*c^5*d^2 - 14*A*
a*c^4*d^3 + 11*C*a*c^4*d^3 + 11*B*b*c^4*d^3 - 7*B*a*c^3*d^4 - 7*A*b*c^3*d^4
 + 4*C*b*c^3*d^4 - 3*A*a*c^2*d^5 - B*a*c*d^6 - A*b*c*d^6 - A*a*d^7)/((c^6*d
^2 + 3*c^4*d^4 + 3*c^2*d^6 + d^8)*(d*tan(f*x + e) + c)^2))/f
```

maple [B] time = 0.33, size = 1513, normalized size = 4.30

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x)
```

```
[Out] 1/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*A*b*c^2-1/f/(c^2+d^2)^2*d^2/(c+d*tan(f*x+e
))*B*a-1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*b*c^3+1/2/f/(c^2+d^2)^3*ln(1+
tan(f*x+e)^2)*A*b*c^3+1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a*c^3-1/2/f/(c
^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*b*d^3-1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C
*a*d^3-3/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*C*b*c^2-1/f/(c^2+d^2)^3*ln(c+d*tan(
f*x+e))*B*a*c^3+1/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*b*d^3+1/f/(c^2+d^2)^3*
ln(c+d*tan(f*x+e))*C*a*d^3+1/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*b*c^3+2/f/(
c^2+d^2)^2*d/(c+d*tan(f*x+e))*C*a*c+2/f/(c^2+d^2)^2*d/(c+d*tan(f*x+e))*B*b*
c+3/f/(c^2+d^2)^3*C*arctan(tan(f*x+e))*a*c*d^2-3/f/(c^2+d^2)^3*C*arctan(tan
(f*x+e))*b*c^2*d+1/2/f/d^2/(c^2+d^2)/(c+d*tan(f*x+e))^2*C*b*c^3-3/f/(c^2+d^
2)^3*ln(c+d*tan(f*x+e))*C*a*c^2*d-3/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*b*c*
d^2-3/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*b*c^2*d-1/2/f/d/(c^2+d^2)/(c+d*tan
(f*x+e))^2*B*b*c^2-1/2/f/d/(c^2+d^2)/(c+d*tan(f*x+e))^2*C*a*c^2+3/2/f/(c^2+
d^2)^3*ln(1+tan(f*x+e)^2)*B*b*c^2*d-2/f/(c^2+d^2)^2*d/(c+d*tan(f*x+e))*A*a*
c-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*b*c*d^2-3/2/f/(c^2+d^2)^3*ln(1+tan
(f*x+e)^2)*B*a*c*d^2+3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a*c^2*d+3/2/f/(
c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*b*c*d^2-3/f/(c^2+d^2)^3*A*arctan(tan(f*x+e)
)*a*c*d^2+3/f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*b*c^2*d+3/f/(c^2+d^2)^3*B*ar
ctan(tan(f*x+e))*a*c^2*d+3/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*b*c*d^2-1/f/(
c^2+d^2)^2/d^2/(c+d*tan(f*x+e))*C*b*c^4-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2
)*A*a*c^2*d+3/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*a*c^2*d+3/f/(c^2+d^2)^3*ln
(c+d*tan(f*x+e))*A*b*c*d^2+3/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*a*c*d^2+1/2
/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*a*d^3+1/f/(c^2+d^2)^3*A*arctan(tan(f*x+
```

```
e))*a*c^3-1/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*a*d^3-1/f/(c^2+d^2)^3*ln(c+d
*tan(f*x+e))*A*b*c^3+1/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*a*c^2-1/f/(c^2+d^2)
^2*d^2/(c+d*tan(f*x+e))*A*b+1/2/f/(c^2+d^2)/(c+d*tan(f*x+e))^2*A*b*c+1/2/f/
(c^2+d^2)/(c+d*tan(f*x+e))^2*B*a*c-1/2/f*d/(c^2+d^2)/(c+d*tan(f*x+e))^2*A*a
-1/f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*b*d^3-1/f/(c^2+d^2)^3*B*arctan(tan(f*
x+e))*a*d^3-1/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*b*c^3-1/f/(c^2+d^2)^3*C*ar
ctan(tan(f*x+e))*a*c^3+1/f/(c^2+d^2)^3*C*arctan(tan(f*x+e))*b*d^3
```

maxima [A] time = 0.60, size = 543, normalized size = 1.54

$$\frac{2\left(\left((A-C)a-Bb\right)c^3+3\left(Ba+(A-C)b\right)c^2d-3\left((A-C)a-Bb\right)cd^2-\left(Ba+(A-C)b\right)d^3\right)\left(fx+e\right)}{c^6+3c^4d^2+3c^2d^4+d^6}-\frac{2\left(\left(Ba+(A-C)b\right)c^3-3\left((A-C)a-Bb\right)c^2d-3\left(Ba+(A-C)b\right)cd^2+\cdots\right.}{c^6+3c^4d^2+3c^2d^4+d^6}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a - B*b)*c^3 + 3*(B*a + (A - C)*b)*c^2*d - 3*((A - C)*a -
B*b)*c*d^2 - (B*a + (A - C)*b)*d^3)*(f*x + e)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4
+ d^6) - 2*((B*a + (A - C)*b)*c^3 - 3*((A - C)*a - B*b)*c^2*d - 3*(B*a + (A
 - C)*b)*c*d^2 + ((A - C)*a - B*b)*d^3)*log(d*tan(f*x + e) + c)/(c^6 + 3*c^
4*d^2 + 3*c^2*d^4 + d^6) + ((B*a + (A - C)*b)*c^3 - 3*((A - C)*a - B*b)*c^2
*d - 3*(B*a + (A - C)*b)*c*d^2 + ((A - C)*a - B*b)*d^3)*log(tan(f*x + e)^2
+ 1)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) - (C*b*c^5 + A*a*d^5 + (C*a + B*b)
*c^4*d - (3*B*a + (3*A - 5*C)*b)*c^3*d^2 + ((5*A - 3*C)*a - 3*B*b)*c^2*d^3
+ (B*a + A*b)*c*d^4 + 2*(C*b*c^4*d - (B*a + (A - 3*C)*b)*c^2*d^3 + 2*((A -
C)*a - B*b)*c*d^4 + (B*a + A*b)*d^5)*tan(f*x + e))/(c^6*d^2 + 2*c^4*d^4 + c
^2*d^6 + (c^4*d^4 + 2*c^2*d^6 + d^8)*tan(f*x + e)^2 + 2*(c^5*d^3 + 2*c^3*d^
5 + c*d^7)*tan(f*x + e)))/f
```

mupad [B] time = 16.53, size = 502, normalized size = 1.43

$$-\frac{\frac{A\,a\,d^5+C\,b\,c^5+A\,b\,c\,d^4+B\,a\,c\,d^4+B\,b\,c^4\,d+C\,a\,c^4\,d+5\,A\,a\,c^2\,d^3-3\,A\,b\,c^3\,d^2-3\,B\,a\,c^3\,d^2-3\,B\,b\,c^2\,d^3-3\,C\,a\,c^2\,d^3+5\,C\,b\,c^3\,d^2}{2\,d^2\,\left(c^4+2\,c^2\,d^2+d^4\right)}+\frac{\mathrm{tan}\left(e+f\,x\right)\,\left(A\,b\,d^4+B\,a\cdots\right.}{\cdots}}{f\,\left(c^2+2\,c\,d\,\mathrm{tan}\left(e+f\,x\right)+d^2\,{\mathrm{tan}\left(e+f\,x\right)}^2\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x))^3,x)
```

```
[Out] - ((A*a*d^5 + C*b*c^5 + A*b*c*d^4 + B*a*c*d^4 + B*b*c^4*d + C*a*c^4*d + 5*A
*a*c^2*d^3 - 3*A*b*c^3*d^2 - 3*B*a*c^3*d^2 - 3*B*b*c^2*d^3 - 3*C*a*c^2*d^3
```

```
+ 5*C*b*c^3*d^2)/(2*d^2*(c^4 + d^4 + 2*c^2*d^2)) + (tan(e + f*x)*(A*b*d^4 +
 B*a*d^4 + C*b*c^4 + 2*A*a*c*d^3 - 2*B*b*c*d^3 - 2*C*a*c*d^3 - A*b*c^2*d^2
- B*a*c^2*d^2 + 3*C*b*c^2*d^2))/(d*(c^4 + d^4 + 2*c^2*d^2)))/(f*(c^2 + d^2*
tan(e + f*x)^2 + 2*c*d*tan(e + f*x))) - (log(tan(e + f*x) + 1i)*(A*b*1i - A
*a + B*a*1i + B*b + C*a - C*b*1i))/(2*f*(c*d^2*3i - 3*c^2*d - c^3*1i + d^3)
) - (log(tan(e + f*x) - 1i)*(A*b - A*a*1i + B*a + B*b*1i + C*a*1i - C*b))/(
2*f*(3*c*d^2 - c^2*d*3i - c^3 + d^3*1i)) - (log(c + d*tan(e + f*x))*(c^3*(A
*b + B*a - C*b) - d^3*(B*b - A*a + C*a) + c^2*d*(3*B*b - 3*A*a + 3*C*a) - c
*d^2*(3*A*b + 3*B*a - 3*C*b)))/(f*(c^6 + d^6 + 3*c^2*d^4 + 3*c^4*d^2))
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
)**3,x)
```

```
[Out] Exception raised: AttributeError
```

3.87 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^3}\,dx$

Optimal. Leaf size=209

$$-\frac{Ad^2-Bcd+c^2C}{2df\left(c^2+d^2\right)(c+d\tan(e+fx))^2}-\frac{2cd(A-C)-B\left(c^2-d^2\right)}{f\left(c^2+d^2\right)^2(c+d\tan(e+fx))}+\frac{\left(d(A-C)\left(3c^2-d^2\right)-B\left(c^3-3cd^2\right)\right)\log}{f\left(c^2+d^2\right)}$$

```
[Out] -(c^3*C-3*B*c^2*d-3*c*C*d^2+B*d^3-A*(c^3-3*c*d^2))*x/(c^2+d^2)^3+((A-C)*d*(
3*c^2-d^2)-B*(c^3-3*c*d^2))*ln(c*cos(f*x+e)+d*sin(f*x+e))/(c^2+d^2)^3/f+1/2
*(-A*d^2+B*c*d-C*c^2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^2+(-2*c*(A-C)*d+B*(c^2
-d^2))/(c^2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] time = 0.38, antiderivative size = 209, normalized size of antiderivative = 1.00, number of steps used = 4, number of rules used = 4, integrand size = 33, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.121, Rules used = {3628, 3529, 3531, 3530}

$$-\frac{Ad^2-Bcd+c^2C}{2df\left(c^2+d^2\right)(c+d\tan(e+fx))^2}-\frac{2cd(A-C)-B\left(c^2-d^2\right)}{f\left(c^2+d^2\right)^2(c+d\tan(e+fx))}+\frac{\left(d(A-C)\left(3c^2-d^2\right)-B\left(c^3-3cd^2\right)\right)\log}{f\left(c^2+d^2\right)}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^3,x]
```

```
[Out] -(((c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2))*x)/(c^2 + d^
2)^3) + (((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2))*Log[c*Cos[e + f*x] +
 d*Sin[e + f*x]])/((c^2 + d^2)^3*f) - (c^2*C - B*c*d + A*d^2)/(2*d*(c^2 + d
^2)*f*(c + d*Tan[e + f*x])^2) - (2*c*(A - C)*d - B*(c^2 - d^2))/((c^2 + d^2
)^2*f*(c + d*Tan[e + f*x]))
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3531

```
Int[((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_.) + (b_.)*tan[(e_.) + (f_.
)*(x_)]), x_Symbol] :> Simp[((a*c + b*d)*x)/(a^2 + b^2), x] + Dist[(b*c - a
*d)/(a^2 + b^2), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] /; F
reeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && Ne
Q[a*c + b*d, 0]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^3}\,dx = -\frac{c^2C-Bcd+Ad^2}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}+\frac{\int \frac{Ac-cC+Bd+(Bc-(A-C)d)\tan(e+fx}{(c+d\tan(e+fx))^2}}{c^2+d^2}$$

$$= -\frac{c^2C-Bcd+Ad^2}{2d\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}-\frac{2c(A-C)d-B\left(c^2-d^2\right)}{\left(c^2+d^2\right)^2 f(c+d\tan(e+fx)}$$

$$= \frac{\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3\right)x}{\left(c^2+d^2\right)^3}-\frac{c^2C-Bc}{2d\left(c^2+d^2\right)f(c+}$$

$$= \frac{\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3\right)x}{\left(c^2+d^2\right)^3}+\frac{\left((A-C)d\left(3c^2-\right.\right.}{}$$

Mathematica [C] time = 5.28, size = 261, normalized size = 1.25

$$-\frac{-(d(C-A)+Bc)\left(\frac{d\left(\frac{\left(c^2+d^2\right)\left(5c^2+4cd\tan(e+fx)+d^2\right)}{(c+d\tan(e+fx))^2}+\left(2d^2-6c^2\right)\log(c+d\tan(e+fx))\right)}{\left(c^2+d^2\right)^3}+\frac{i\log(-\tan(e+fx)+i)}{(c+id)^3}-\frac{\log(\tan(e+fx)+i)}{(d+ic)^3}\right)+B\left(\frac{2d}{}\right.}{2df}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^3,x]
```

```
[Out] -1/2*(C/(c + d*Tan[e + f*x])^2 + B*((I*Log[I - Tan[e + f*x]])/(c + I*d)^2 -
 (I*Log[I + Tan[e + f*x]])/(c - I*d)^2 + (2*d*(-2*c*Log[c + d*Tan[e + f*x]]
 + (c^2 + d^2)/(c + d*Tan[e + f*x])))/(c^2 + d^2)^2) - (B*c + (-A + C)*d)*(
(I*Log[I - Tan[e + f*x]])/(c + I*d)^3 - Log[I + Tan[e + f*x]]/(I*c + d)^3 +
 (d*((-6*c^2 + 2*d^2)*Log[c + d*Tan[e + f*x]] + ((c^2 + d^2)*(5*c^2 + d^2 +
 4*c*d*Tan[e + f*x]))/(c + d*Tan[e + f*x])^2))/(c^2 + d^2)^3))/(d*f)
```

fricas [B] time = 0.72, size = 566, normalized size = 2.71

$$-\frac{3\,Cc^4d-5\,Bc^3d^2+(7\,A-3\,C)c^2d^3+Bcd^4+Ad^5-2\left((A-C)c^5+3\,Bc^4d-3\,(A-C)c^3d^2-Bc^2d^3\right)fx-\left(Cc^4\right.}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x, algorithm="
fricas")
```

```
[Out] -1/2*(3*C*c^4*d - 5*B*c^3*d^2 + (7*A - 3*C)*c^2*d^3 + B*c*d^4 + A*d^5 - 2*(
(A - C)*c^5 + 3*B*c^4*d - 3*(A - C)*c^3*d^2 - B*c^2*d^3)*f*x - (C*c^4*d - 3
*B*c^3*d^2 + 5*(A - C)*c^2*d^3 + 3*B*c*d^4 - A*d^5 + 2*((A - C)*c^3*d^2 + 3
*B*c^2*d^3 - 3*(A - C)*c*d^4 - B*d^5)*f*x)*tan(f*x + e)^2 + (B*c^5 - 3*(A -
 C)*c^4*d - 3*B*c^3*d^2 + (A - C)*c^2*d^3 + (B*c^3*d^2 - 3*(A - C)*c^2*d^3
- 3*B*c*d^4 + (A - C)*d^5)*tan(f*x + e)^2 + 2*(B*c^4*d - 3*(A - C)*c^3*d^2
- 3*B*c^2*d^3 + (A - C)*c*d^4)*tan(f*x + e))*log((d^2*tan(f*x + e)^2 + 2*c*
d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) - 2*(C*c^5 - 2*B*c^4*d + 3*(A -
 C)*c^3*d^2 + 3*B*c^2*d^3 - (3*A - 2*C)*c*d^4 - B*d^5 + 2*((A - C)*c^4*d +
3*B*c^3*d^2 - 3*(A - C)*c^2*d^3 - B*c*d^4)*f*x)*tan(f*x + e))/((c^6*d^2 + 3
*c^4*d^4 + 3*c^2*d^6 + d^8)*f*tan(f*x + e)^2 + 2*(c^7*d + 3*c^5*d^3 + 3*c^3
*d^5 + c*d^7)*f*tan(f*x + e) + (c^8 + 3*c^6*d^2 + 3*c^4*d^4 + c^2*d^6)*f)
```

giac [B] time = 1.39, size = 548, normalized size = 2.62

$$\frac{\frac{2\left(Ac^3-Cc^3+3\,Bc^2d-3\,Acd^2+3\,Ccd^2-Bd^3\right)\left(fx+e\right)}{c^6+3\,c^4d^2+3\,c^2d^4+d^6}+\frac{\left(Bc^3-3\,Ac^2d+3\,Cc^2d-3\,Bcd^2+Ad^3-Cd^3\right)\log\left(\tan\left(fx+e\right)^2+1\right)}{c^6+3\,c^4d^2+3\,c^2d^4+d^6}-\frac{2\left(Bc^3d-3\,Ac^2d^2+3\,Cc^2d^2-3\right.}{c^6d+3}}{}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x, algorithm="
giac")
```

```
[Out] 1/2*(2*(A*c^3 - C*c^3 + 3*B*c^2*d - 3*A*c*d^2 + 3*C*c*d^2 - B*d^3)*(f*x + e
)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) + (B*c^3 - 3*A*c^2*d + 3*C*c^2*d - 3*
```

```
B*c*d^2 + A*d^3 - C*d^3)*log(tan(f*x + e)^2 + 1)/(c^6 + 3*c^4*d^2 + 3*c^2*d
^4 + d^6) - 2*(B*c^3*d - 3*A*c^2*d^2 + 3*C*c^2*d^2 - 3*B*c*d^3 + A*d^4 - C*
d^4)*log(abs(d*tan(f*x + e) + c))/(c^6*d + 3*c^4*d^3 + 3*c^2*d^5 + d^7) + (
3*B*c^3*d^3*tan(f*x + e)^2 - 9*A*c^2*d^4*tan(f*x + e)^2 + 9*C*c^2*d^4*tan(f
*x + e)^2 - 9*B*c*d^5*tan(f*x + e)^2 + 3*A*d^6*tan(f*x + e)^2 - 3*C*d^6*tan
(f*x + e)^2 + 8*B*c^4*d^2*tan(f*x + e) - 22*A*c^3*d^3*tan(f*x + e) + 22*C*c
^3*d^3*tan(f*x + e) - 18*B*c^2*d^4*tan(f*x + e) + 2*A*c*d^5*tan(f*x + e) -
2*C*c*d^5*tan(f*x + e) - 2*B*d^6*tan(f*x + e) - C*c^6 + 6*B*c^5*d - 14*A*c^
4*d^2 + 11*C*c^4*d^2 - 7*B*c^3*d^3 - 3*A*c^2*d^4 - B*c*d^5 - A*d^6)/((c^6*d
 + 3*c^4*d^3 + 3*c^2*d^5 + d^7)*(d*tan(f*x + e) + c)^2))/f
```

maple [B] time = 0.29, size = 713, normalized size = 3.41

$$-\frac{dA}{2f\left(c^2+d^2\right)\left(c+d\tan\left(fx+e\right)\right)^2}+\frac{Bc}{2f\left(c^2+d^2\right)\left(c+d\tan\left(fx+e\right)\right)^2}-\frac{B\arctan\left(\tan\left(fx+e\right)\right)d^3}{f\left(c^2+d^2\right)^3}-\frac{C\arctan\left(\mathrm{ta}\right.}{f\left(c^2\right.}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x)
```

```
[Out] -1/2/f/(c^2+d^2)*d/(c+d*tan(f*x+e))^2*A+1/2/f/(c^2+d^2)/(c+d*tan(f*x+e))^2*
B*c-1/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*d^3-1/f/(c^2+d^2)^3*C*arctan(tan(f
*x+e))*c^3+1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*d^3+1/2/f/(c^2+d^2)^3*ln(
1+tan(f*x+e)^2)*B*c^3-1/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*d^3-1/f/(c^2+d
^2)^3*ln(c+d*tan(f*x+e))*A*d^3-1/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*c^3+1/f
/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*d^3+1/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*c^
2+3/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*c^2*d+3/f/(c^2+d^2)^3*ln(c+d*tan(f*x
+e))*B*c*d^2-3/f/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*c^2*d-3/2/f/(c^2+d^2)^3*l
n(1+tan(f*x+e)^2)*A*c^2*d-3/2/f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*c*d^2+3/2/
f/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*c^2*d-3/f/(c^2+d^2)^3*A*arctan(tan(f*x+e
))*c*d^2+1/f/(c^2+d^2)^3*A*arctan(tan(f*x+e))*c^3-1/f/(c^2+d^2)^2/(c+d*tan(
f*x+e))*d^2*B+3/f/(c^2+d^2)^3*B*arctan(tan(f*x+e))*c^2*d+3/f/(c^2+d^2)^3*C*
arctan(tan(f*x+e))*c*d^2-1/2/f/(c^2+d^2)/d/(c+d*tan(f*x+e))^2*c^2*C-2/f/(c^
2+d^2)^2/(c+d*tan(f*x+e))*A*c*d+2/f/(c^2+d^2)^2/(c+d*tan(f*x+e))*c*C*d
```

maxima [A] time = 0.64, size = 367, normalized size = 1.76

$$\frac{\frac{2\left((A-C)c^3+3\,Bc^2d-3\,(A-C)cd^2-Bd^3\right)\left(fx+e\right)}{c^6+3\,c^4d^2+3\,c^2d^4+d^6}-\frac{2\left(Bc^3-3\,(A-C)c^2d-3\,Bcd^2+(A-C)d^3\right)\log\left(d\tan\left(fx+e\right)+c\right)}{c^6+3\,c^4d^2+3\,c^2d^4+d^6}+\frac{\left(Bc^3-3\,(A-C)c^2d-3\,Bcd^2+(A-C)d^3\right)}{c^6+3\,c^4d^2+3\,c^2d^4+}}{2\,f}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x, algorithm="
maxima")
```

```
[Out] 1/2*(2*((A - C)*c^3 + 3*B*c^2*d - 3*(A - C)*c*d^2 - B*d^3)*(f*x + e)/(c^6 +
 3*c^4*d^2 + 3*c^2*d^4 + d^6) - 2*(B*c^3 - 3*(A - C)*c^2*d - 3*B*c*d^2 + (A
 - C)*d^3)*log(d*tan(f*x + e) + c)/(c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) + (B
*c^3 - 3*(A - C)*c^2*d - 3*B*c*d^2 + (A - C)*d^3)*log(tan(f*x + e)^2 + 1)/(
c^6 + 3*c^4*d^2 + 3*c^2*d^4 + d^6) - (C*c^4 - 3*B*c^3*d + (5*A - 3*C)*c^2*d
^2 + B*c*d^3 + A*d^4 - 2*(B*c^2*d^2 - 2*(A - C)*c*d^3 - B*d^4)*tan(f*x + e)
)/(c^6*d + 2*c^4*d^3 + c^2*d^5 + (c^4*d^3 + 2*c^2*d^5 + d^7)*tan(f*x + e)^2
 + 2*(c^5*d^2 + 2*c^3*d^4 + c*d^6)*tan(f*x + e)))/f
```

mupad [B] time = 11.88, size = 327, normalized size = 1.56

$$-\frac{\frac{\tan\left(e+f\,x\right)\left(B\,d^3+2\,A\,c\,d^2-B\,c^2\,d-2\,C\,c\,d^2\right)}{c^4+2\,c^2\,d^2+d^4}+\frac{A\,d^4+C\,c^4+5\,A\,c^2\,d^2-3\,C\,c^2\,d^2+B\,c\,d^3-3\,B\,c^3\,d}{2\,d\left(c^4+2\,c^2\,d^2+d^4\right)}}{f\,\left(c^2+2\,c\,d\tan\left(e+f\,x\right)+d^2\tan\left(e+f\,x\right)^2\right)}-\frac{\ln\left(\tan\left(e+f\,x\right)-\mathrm{i}\right)\,\left(B-A\,1\mathrm{i}+C\,1\mathrm{i}\right.}{2\,f\,\left(-c^3-c^2\,d\,3\mathrm{i}+3\,c\,d^2+d^3\,1\mathrm{i}\right)}$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x))^3,x)
```

```
[Out] - ((tan(e + f*x)*(B*d^3 + 2*A*c*d^2 - B*c^2*d - 2*C*c*d^2))/(c^4 + d^4 + 2*
c^2*d^2) + (A*d^4 + C*c^4 + 5*A*c^2*d^2 - 3*C*c^2*d^2 + B*c*d^3 - 3*B*c^3*d
)/(2*d*(c^4 + d^4 + 2*c^2*d^2)))/(f*(c^2 + d^2*tan(e + f*x)^2 + 2*c*d*tan(e
 + f*x))) - (log(tan(e + f*x) - 1i)*(B - A*1i + C*1i))/(2*f*(3*c*d^2 - c^2*
d*3i - c^3 + d^3*1i)) - (log(c + d*tan(e + f*x))*(B*c^3 + d^3*(A - C) - c^2
*d*(3*A - 3*C) - 3*B*c*d^2))/(f*(c^6 + d^6 + 3*c^2*d^4 + 3*c^4*d^2)) - (log
(tan(e + f*x) + 1i)*(B*1i - A + C))/(2*f*(c*d^2*3i - 3*c^2*d - c^3*1i + d^3
))
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: AttributeError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**3,x)
```

```
[Out] Exception raised: AttributeError
```

3.88 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx$

Optimal. Leaf size=487

$$-\frac{x\left(a\left(-A\left(c^3-3cd^2\right)-3Bc^2d+Bd^3+c^3C-3cCd^2\right)+b\left(d(A-C)\left(3c^2-d^2\right)-B\left(c^3-3cd^2\right)\right)\right)}{\left(a^2+b^2\right)\left(c^2+d^2\right)^3}-\frac{\left(a^2d^3\left(d(A-C\right.\right.}{}$$

```
[Out] -(a*(c^3*C-3*B*c^2*d-3*c*C*d^2+B*d^3-A*(c^3-3*c*d^2))+b*((A-C)*d*(3*c^2-d^2
)-B*(c^3-3*c*d^2)))*x/(a^2+b^2)/(c^2+d^2)^3+b^2*(A*b^2-a*(B*b-C*a))*ln(a*co
s(f*x+e)+b*sin(f*x+e))/(a^2+b^2)/(-a*d+b*c)^3/f-(b^2*(c^6*C-3*B*c^5*d+3*c^4
*(2*A-C)*d^2+B*c^3*d^3+3*A*c^2*d^4+A*d^6)+a^2*d^3*((A-C)*d*(3*c^2-d^2)-B*(c
^3-3*c*d^2))-a*b*d^2*(8*c^3*(A-C)*d-B*(3*c^4-6*c^2*d^2-d^4)))*ln(c*cos(f*x+
e)+d*sin(f*x+e))/(-a*d+b*c)^3/(c^2+d^2)^3/f+1/2*(A*d^2-B*c*d+C*c^2)/(-a*d+b
*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))^2+(b*(c^4*C-2*B*c^3*d+c^2*(3*A-C)*d^2+A*d^
4)-a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))/(-a*d+b*c)^2/(c^2+d^2)^2/f/(c+d*tan(f*x
+e))
```

Rubi [A] time = 1.83, antiderivative size = 487, normalized size of antiderivative = 1.00, number of steps used = 5, number of rules used = 3, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3649, 3651, 3530}

$$-\frac{\left(a^2d^3\left(d(A-C)\left(3c^2-d^2\right)-B\left(c^3-3cd^2\right)\right)-abd^2\left(8c^3d(A-C)-B\left(-6c^2d^2+3c^4-d^4\right)\right)+b^2\left(3c^4d^2(2A-C)\right.\right.}{f\left(c^2+d^2\right)^3(bc-ad)^3}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c + d*Ta
n[e + f*x])^3),x]
```

```
[Out] -(((a*(c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)) + b*((A -
 C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)))*x)/((a^2 + b^2)*(c^2 + d^2)^3)) +
 (b^2*(A*b^2 - a*(b*B - a*C))*Log[a*Cos[e + f*x] + b*Sin[e + f*x]])/((a^2 +
 b^2)*(b*c - a*d)^3*f) - ((b^2*(c^6*C - 3*B*c^5*d + 3*c^4*(2*A - C)*d^2 + B
*c^3*d^3 + 3*A*c^2*d^4 + A*d^6) + a^2*d^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3
 - 3*c*d^2)) - a*b*d^2*(8*c^3*(A - C)*d - B*(3*c^4 - 6*c^2*d^2 - d^4)))*Log
[c*Cos[e + f*x] + d*Sin[e + f*x]])/((b*c - a*d)^3*(c^2 + d^2)^3*f) + (c^2*C
 - B*c*d + A*d^2)/(2*(b*c - a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^2) + (b
*(c^4*C - 2*B*c^3*d + c^2*(3*A - C)*d^2 + A*d^4) - a*d^2*(2*c*(A - C)*d - B
*(c^2 - d^2)))/((b*c - a*d)^2*(c^2 + d^2)^2*f*(c + d*Tan[e + f*x]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
```

```
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^3}\,dx = \frac{c^2C-Bcd+Ad^2}{2(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} + \int \frac{-2\big(aAcd-ad(cC-Bd)-A}{}$$

$$= \frac{c^2C-Bcd+Ad^2}{2(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} + \frac{b\left(c^4C-2Bc^3d+c^2\right.}{(bc-}$$

$$= -\frac{\left(b(A-C)d\left(3c^2-d^2\right)-bB\left(c^3-3cd^2\right)-a\left(Ac^3-c^3C+3Bc^2d\right.\right.}{\left(a^2+b^2\right)\left(c^2+d^2\right)^3}$$

$$= -\frac{\left(b(A-C)d\left(3c^2-d^2\right)-bB\left(c^3-3cd^2\right)-a\left(Ac^3-c^3C+3Bc^2d\right.\right.}{\left(a^2+b^2\right)\left(c^2+d^2\right)^3}$$

Mathematica [A] time = 9.24, size = 912, normalized size = 1.87

$$-\frac{Ad^2-c(Bd-cC)}{2(ad-bc)\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} - \frac{-\frac{-2\left(aAcd-a(cC-Bd)d-Ab\left(c^2+d^2\right)\right)d^2-c\left(2d(bc-ad)(Bc-(A-C)d)-2bc\left(Cc^2-Bdc+Ad^2\right)\right)}{(ad-bc)\left(c^2+d^2\right)f(c+d\tan(e+fx))}-}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c
+ d*Tan[e + f*x])^3),x]
```

```
[Out] -1/2*(A*d^2 - c*(-(c*C) + B*d))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e
+ f*x])^2) - (-((-((b*(b*c - a*d)^2*(A*b*c^3 - a*B*c^3 - b*c^3*C + 3*a*A*c^
2*d + 3*b*B*c^2*d - 3*a*c^2*C*d - 3*A*b*c*d^2 + 3*a*B*c*d^2 + 3*b*c*C*d^2 -
 a*A*d^3 - b*B*d^3 + a*C*d^3 - (Sqrt[-b^2]*(a*(c^3*C - 3*B*c^2*d - 3*c*C*d^
2 + B*d^3 - A*(c^3 - 3*c*d^2)) + b*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*
d^2))))/b)*Log[Sqrt[-b^2] - b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2))) + (
2*b^3*(A*b^2 - a*(b*B - a*C))*(c^2 + d^2)^2*Log[a + b*Tan[e + f*x]])/((a^2
+ b^2)*(b*c - a*d)) - (b*(b*c - a*d)^2*(A*b*c^3 - a*B*c^3 - b*c^3*C + 3*a*A
*c^2*d + 3*b*B*c^2*d - 3*a*c^2*C*d - 3*A*b*c*d^2 + 3*a*B*c*d^2 + 3*b*c*C*d^
2 - a*A*d^3 - b*B*d^3 + a*C*d^3 + (Sqrt[-b^2]*(b*(A - C)*d*(3*c^2 - d^2) -
b*B*(c^3 - 3*c*d^2) - a*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2
- B*d^3)))/b)*Log[Sqrt[-b^2] + b*Tan[e + f*x]])/((a^2 + b^2)*(c^2 + d^2)) -
 (2*b*(b^2*(c^6*C - 3*B*c^5*d + 3*c^4*(2*A - C)*d^2 + B*c^3*d^3 + 3*A*c^2*d
^4 + A*d^6) + a^2*d^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) - a*b*d
```

```
^2*(8*c^3*(A - C)*d - B*(3*c^4 - 6*c^2*d^2 - d^4)))*Log[c + d*Tan[e + f*x]]
)/((b*c - a*d)*(c^2 + d^2)))/(b*(-(b*c) + a*d)*(c^2 + d^2)*f)) - (-2*d^2*(a
*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)) - c*(2*d*(b*c - a*d)*(B*c - (A
- C)*d) - 2*b*c*(c^2*C - B*c*d + A*d^2)))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c
+ d*Tan[e + f*x])))/(2*(-(b*c) + a*d)*(c^2 + d^2))
```

fricas [B] time = 6.77, size = 3496, normalized size = 7.18

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^3,x, algorithm="fricas")
```

```
[Out] 1/2*(5*(C*a^2*b^2 + C*b^4)*c^6*d^2 - (8*C*a^3*b + 7*B*a^2*b^2 + 8*C*a*b^3 +
 7*B*b^4)*c^5*d^3 + (3*C*a^4 + 12*B*a^3*b + (9*A + 2*C)*a^2*b^2 + 12*B*a*b^
3 + (9*A - C)*b^4)*c^4*d^4 - (5*B*a^4 + 4*(4*A - C)*a^3*b + 6*B*a^2*b^2 + 4
*(4*A - C)*a*b^3 + B*b^4)*c^3*d^5 + ((7*A - 3*C)*a^4 + (10*A - 3*C)*a^2*b^2
 + 3*A*b^4)*c^2*d^6 + (B*a^4 - 4*A*a^3*b + B*a^2*b^2 - 4*A*a*b^3)*c*d^7 + (
A*a^4 + A*a^2*b^2)*d^8 + 2*(((A - C)*a*b^3 + B*b^4)*c^8 - 3*((A - C)*a^2*b^
2 + (A - C)*b^4)*c^7*d + 3*((A - C)*a^3*b - 2*B*a^2*b^2 + 2*(A - C)*a*b^3 -
 B*b^4)*c^6*d^2 - ((A - C)*a^4 - 8*B*a^3*b - 8*B*a*b^3 - (A - C)*b^4)*c^5*d
^3 - 3*(B*a^4 + 2*(A - C)*a^3*b + 2*B*a^2*b^2 + (A - C)*a*b^3)*c^4*d^4 + 3*
((A - C)*a^4 + (A - C)*a^2*b^2)*c^3*d^5 + (B*a^4 - (A - C)*a^3*b)*c^2*d^6)*
f*x - (3*(C*a^2*b^2 + C*b^4)*c^6*d^2 - (4*C*a^3*b + 5*B*a^2*b^2 + 4*C*a*b^3
 + 5*B*b^4)*c^5*d^3 + (C*a^4 + 8*B*a^3*b + (7*A - 2*C)*a^2*b^2 + 8*B*a*b^3
+ (7*A - 3*C)*b^4)*c^4*d^4 - (3*B*a^4 + 4*(3*A - 2*C)*a^3*b + 2*B*a^2*b^2 +
 4*(3*A - 2*C)*a*b^3 - B*b^4)*c^3*d^5 + (5*(A - C)*a^4 - 4*B*a^3*b + (6*A -
 5*C)*a^2*b^2 - 4*B*a*b^3 + A*b^4)*c^2*d^6 + 3*(B*a^4 + B*a^2*b^2)*c*d^7 -
(A*a^4 + A*a^2*b^2)*d^8 - 2*(((A - C)*a*b^3 + B*b^4)*c^6*d^2 - 3*((A - C)*a
^2*b^2 + (A - C)*b^4)*c^5*d^3 + 3*((A - C)*a^3*b - 2*B*a^2*b^2 + 2*(A - C)*
a*b^3 - B*b^4)*c^4*d^4 - ((A - C)*a^4 - 8*B*a^3*b - 8*B*a*b^3 - (A - C)*b^4
)*c^3*d^5 - 3*(B*a^4 + 2*(A - C)*a^3*b + 2*B*a^2*b^2 + (A - C)*a*b^3)*c^2*d
^6 + 3*((A - C)*a^4 + (A - C)*a^2*b^2)*c*d^7 + (B*a^4 - (A - C)*a^3*b)*d^8)
*f*x)*tan(f*x + e)^2 + ((C*a^2*b^2 - B*a*b^3 + A*b^4)*c^8 + 3*(C*a^2*b^2 -
B*a*b^3 + A*b^4)*c^6*d^2 + 3*(C*a^2*b^2 - B*a*b^3 + A*b^4)*c^4*d^4 + (C*a^2
*b^2 - B*a*b^3 + A*b^4)*c^2*d^6 + ((C*a^2*b^2 - B*a*b^3 + A*b^4)*c^6*d^2 +
3*(C*a^2*b^2 - B*a*b^3 + A*b^4)*c^4*d^4 + 3*(C*a^2*b^2 - B*a*b^3 + A*b^4)*c
^2*d^6 + (C*a^2*b^2 - B*a*b^3 + A*b^4)*d^8)*tan(f*x + e)^2 + 2*((C*a^2*b^2
- B*a*b^3 + A*b^4)*c^7*d + 3*(C*a^2*b^2 - B*a*b^3 + A*b^4)*c^5*d^3 + 3*(C*a
^2*b^2 - B*a*b^3 + A*b^4)*c^3*d^5 + (C*a^2*b^2 - B*a*b^3 + A*b^4)*c*d^7)*ta
n(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x +
e)^2 + 1)) - ((C*a^2*b^2 + C*b^4)*c^8 - 3*(B*a^2*b^2 + B*b^4)*c^7*d + 3*(B*
a^3*b + (2*A - C)*a^2*b^2 + B*a*b^3 + (2*A - C)*b^4)*c^6*d^2 - (B*a^4 + 8*(
A - C)*a^3*b + 8*(A - C)*a*b^3 - B*b^4)*c^5*d^3 + 3*((A - C)*a^4 - 2*B*a^3*
```

```
b + (2*A - C)*a^2*b^2 - 2*B*a*b^3 + A*b^4)*c^4*d^4 + 3*(B*a^4 + B*a^2*b^2)*
c^3*d^5 - ((A - C)*a^4 + B*a^3*b - C*a^2*b^2 + B*a*b^3 - A*b^4)*c^2*d^6 + (
(C*a^2*b^2 + C*b^4)*c^6*d^2 - 3*(B*a^2*b^2 + B*b^4)*c^5*d^3 + 3*(B*a^3*b +
(2*A - C)*a^2*b^2 + B*a*b^3 + (2*A - C)*b^4)*c^4*d^4 - (B*a^4 + 8*(A - C)*a
^3*b + 8*(A - C)*a*b^3 - B*b^4)*c^3*d^5 + 3*((A - C)*a^4 - 2*B*a^3*b + (2*A
 - C)*a^2*b^2 - 2*B*a*b^3 + A*b^4)*c^2*d^6 + 3*(B*a^4 + B*a^2*b^2)*c*d^7 -
((A - C)*a^4 + B*a^3*b - C*a^2*b^2 + B*a*b^3 - A*b^4)*d^8)*tan(f*x + e)^2 +
 2*((C*a^2*b^2 + C*b^4)*c^7*d - 3*(B*a^2*b^2 + B*b^4)*c^6*d^2 + 3*(B*a^3*b
+ (2*A - C)*a^2*b^2 + B*a*b^3 + (2*A - C)*b^4)*c^5*d^3 - (B*a^4 + 8*(A - C)
*a^3*b + 8*(A - C)*a*b^3 - B*b^4)*c^4*d^4 + 3*((A - C)*a^4 - 2*B*a^3*b + (2
*A - C)*a^2*b^2 - 2*B*a*b^3 + A*b^4)*c^3*d^5 + 3*(B*a^4 + B*a^2*b^2)*c^2*d^
6 - ((A - C)*a^4 + B*a^3*b - C*a^2*b^2 + B*a*b^3 - A*b^4)*c*d^7)*tan(f*x +
e))*log((d^2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1
)) - 2*(2*(C*a^2*b^2 + C*b^4)*c^7*d - 3*(C*a^3*b + B*a^2*b^2 + C*a*b^3 + B*
b^4)*c^6*d^2 + (C*a^4 + 5*B*a^3*b + 2*(2*A - C)*a^2*b^2 + 5*B*a*b^3 + (4*A
- 3*C)*b^4)*c^5*d^3 - (2*B*a^4 + (7*A - 6*C)*a^3*b - B*a^2*b^2 + (7*A - 6*C
)*a*b^3 - 3*B*b^4)*c^4*d^4 + (3*(A - C)*a^4 - 6*B*a^3*b - 2*C*a^2*b^2 - 6*B
*a*b^3 - (3*A - C)*b^4)*c^3*d^5 + 3*(B*a^4 + (2*A - C)*a^3*b + B*a^2*b^2 +
(2*A - C)*a*b^3)*c^2*d^6 - ((3*A - 2*C)*a^4 - B*a^3*b + 2*(2*A - C)*a^2*b^2
 - B*a*b^3 + A*b^4)*c*d^7 - (B*a^4 - A*a^3*b + B*a^2*b^2 - A*a*b^3)*d^8 - 2
*(((A - C)*a*b^3 + B*b^4)*c^7*d - 3*((A - C)*a^2*b^2 + (A - C)*b^4)*c^6*d^2
 + 3*((A - C)*a^3*b - 2*B*a^2*b^2 + 2*(A - C)*a*b^3 - B*b^4)*c^5*d^3 - ((A
- C)*a^4 - 8*B*a^3*b - 8*B*a*b^3 - (A - C)*b^4)*c^4*d^4 - 3*(B*a^4 + 2*(A -
 C)*a^3*b + 2*B*a^2*b^2 + (A - C)*a*b^3)*c^3*d^5 + 3*((A - C)*a^4 + (A - C)
*a^2*b^2)*c^2*d^6 + (B*a^4 - (A - C)*a^3*b)*c*d^7)*f*x)*tan(f*x + e))/(((a^
2*b^3 + b^5)*c^9*d^2 - 3*(a^3*b^2 + a*b^4)*c^8*d^3 + 3*(a^4*b + 2*a^2*b^3 +
 b^5)*c^7*d^4 - (a^5 + 10*a^3*b^2 + 9*a*b^4)*c^6*d^5 + 3*(3*a^4*b + 4*a^2*b
^3 + b^5)*c^5*d^6 - 3*(a^5 + 4*a^3*b^2 + 3*a*b^4)*c^4*d^7 + (9*a^4*b + 10*a
^2*b^3 + b^5)*c^3*d^8 - 3*(a^5 + 2*a^3*b^2 + a*b^4)*c^2*d^9 + 3*(a^4*b + a^
2*b^3)*c*d^10 - (a^5 + a^3*b^2)*d^11)*f*tan(f*x + e)^2 + 2*((a^2*b^3 + b^5)
*c^10*d - 3*(a^3*b^2 + a*b^4)*c^9*d^2 + 3*(a^4*b + 2*a^2*b^3 + b^5)*c^8*d^3
 - (a^5 + 10*a^3*b^2 + 9*a*b^4)*c^7*d^4 + 3*(3*a^4*b + 4*a^2*b^3 + b^5)*c^6
*d^5 - 3*(a^5 + 4*a^3*b^2 + 3*a*b^4)*c^5*d^6 + (9*a^4*b + 10*a^2*b^3 + b^5)
*c^4*d^7 - 3*(a^5 + 2*a^3*b^2 + a*b^4)*c^3*d^8 + 3*(a^4*b + a^2*b^3)*c^2*d^
9 - (a^5 + a^3*b^2)*c*d^10)*f*tan(f*x + e) + ((a^2*b^3 + b^5)*c^11 - 3*(a^3
*b^2 + a*b^4)*c^10*d + 3*(a^4*b + 2*a^2*b^3 + b^5)*c^9*d^2 - (a^5 + 10*a^3*
b^2 + 9*a*b^4)*c^8*d^3 + 3*(3*a^4*b + 4*a^2*b^3 + b^5)*c^7*d^4 - 3*(a^5 + 4
*a^3*b^2 + 3*a*b^4)*c^6*d^5 + (9*a^4*b + 10*a^2*b^3 + b^5)*c^5*d^6 - 3*(a^5
 + 2*a^3*b^2 + a*b^4)*c^4*d^7 + 3*(a^4*b + a^2*b^3)*c^3*d^8 - (a^5 + a^3*b^
2)*c^2*d^9)*f)
```

giac [B] time = 17.82, size = 2125, normalized size = 4.36

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^3,x, algorithm="giac")

[Out] 1/2*(2*(A*a*c^3 - C*a*c^3 + B*b*c^3 + 3*B*a*c^2*d - 3*A*b*c^2*d + 3*C*b*c^2
*d - 3*A*a*c*d^2 + 3*C*a*c*d^2 - 3*B*b*c*d^2 - B*a*d^3 + A*b*d^3 - C*b*d^3)
*(f*x + e)/(a^2*c^6 + b^2*c^6 + 3*a^2*c^4*d^2 + 3*b^2*c^4*d^2 + 3*a^2*c^2*d
^4 + 3*b^2*c^2*d^4 + a^2*d^6 + b^2*d^6) + (B*a*c^3 - A*b*c^3 + C*b*c^3 - 3*
A*a*c^2*d + 3*C*a*c^2*d - 3*B*b*c^2*d - 3*B*a*c*d^2 + 3*A*b*c*d^2 - 3*C*b*c
*d^2 + A*a*d^3 - C*a*d^3 + B*b*d^3)*log(tan(f*x + e)^2 + 1)/(a^2*c^6 + b^2*
c^6 + 3*a^2*c^4*d^2 + 3*b^2*c^4*d^2 + 3*a^2*c^2*d^4 + 3*b^2*c^2*d^4 + a^2*d
^6 + b^2*d^6) + 2*(C*a^2*b^3 - B*a*b^4 + A*b^5)*log(abs(b*tan(f*x + e) + a)
)/(a^2*b^4*c^3 + b^6*c^3 - 3*a^3*b^3*c^2*d - 3*a*b^5*c^2*d + 3*a^4*b^2*c*d^
2 + 3*a^2*b^4*c*d^2 - a^5*b*d^3 - a^3*b^3*d^3) - 2*(C*b^2*c^6*d - 3*B*b^2*c
^5*d^2 + 3*B*a*b*c^4*d^3 + 6*A*b^2*c^4*d^3 - 3*C*b^2*c^4*d^3 - B*a^2*c^3*d^
4 - 8*A*a*b*c^3*d^4 + 8*C*a*b*c^3*d^4 + B*b^2*c^3*d^4 + 3*A*a^2*c^2*d^5 - 3
*C*a^2*c^2*d^5 - 6*B*a*b*c^2*d^5 + 3*A*b^2*c^2*d^5 + 3*B*a^2*c*d^6 - A*a^2*
d^7 + C*a^2*d^7 - B*a*b*d^7 + A*b^2*d^7)*log(abs(d*tan(f*x + e) + c))/(b^3*
c^9*d - 3*a*b^2*c^8*d^2 + 3*a^2*b*c^7*d^3 + 3*b^3*c^7*d^3 - a^3*c^6*d^4 - 9
*a*b^2*c^6*d^4 + 9*a^2*b*c^5*d^5 + 3*b^3*c^5*d^5 - 3*a^3*c^4*d^6 - 9*a*b^2*
c^4*d^6 + 9*a^2*b*c^3*d^7 + b^3*c^3*d^7 - 3*a^3*c^2*d^8 - 3*a*b^2*c^2*d^8 +
 3*a^2*b*c*d^9 - a^3*d^10) + (3*C*b^2*c^6*d^2*tan(f*x + e)^2 - 9*B*b^2*c^5*
d^3*tan(f*x + e)^2 + 9*B*a*b*c^4*d^4*tan(f*x + e)^2 + 18*A*b^2*c^4*d^4*tan(
f*x + e)^2 - 9*C*b^2*c^4*d^4*tan(f*x + e)^2 - 3*B*a^2*c^3*d^5*tan(f*x + e)^
2 - 24*A*a*b*c^3*d^5*tan(f*x + e)^2 + 24*C*a*b*c^3*d^5*tan(f*x + e)^2 + 3*B
*b^2*c^3*d^5*tan(f*x + e)^2 + 9*A*a^2*c^2*d^6*tan(f*x + e)^2 - 9*C*a^2*c^2*
d^6*tan(f*x + e)^2 - 18*B*a*b*c^2*d^6*tan(f*x + e)^2 + 9*A*b^2*c^2*d^6*tan(
f*x + e)^2 + 9*B*a^2*c*d^7*tan(f*x + e)^2 - 3*A*a^2*d^8*tan(f*x + e)^2 + 3*
C*a^2*d^8*tan(f*x + e)^2 - 3*B*a*b*d^8*tan(f*x + e)^2 + 3*A*b^2*d^8*tan(f*x
 + e)^2 + 8*C*b^2*c^7*d*tan(f*x + e) - 2*C*a*b*c^6*d^2*tan(f*x + e) - 22*B*
b^2*c^6*d^2*tan(f*x + e) + 24*B*a*b*c^5*d^3*tan(f*x + e) + 42*A*b^2*c^5*d^3
*tan(f*x + e) - 18*C*b^2*c^5*d^3*tan(f*x + e) - 8*B*a^2*c^4*d^4*tan(f*x + e
) - 58*A*a*b*c^4*d^4*tan(f*x + e) + 52*C*a*b*c^4*d^4*tan(f*x + e) + 2*B*b^2
*c^4*d^4*tan(f*x + e) + 22*A*a^2*c^3*d^5*tan(f*x + e) - 22*C*a^2*c^3*d^5*ta
n(f*x + e) - 32*B*a*b*c^3*d^5*tan(f*x + e) + 26*A*b^2*c^3*d^5*tan(f*x + e)
- 2*C*b^2*c^3*d^5*tan(f*x + e) + 18*B*a^2*c^2*d^6*tan(f*x + e) - 12*A*a*b*c
^2*d^6*tan(f*x + e) + 6*C*a*b*c^2*d^6*tan(f*x + e) - 2*A*a^2*c*d^7*tan(f*x
+ e) + 2*C*a^2*c*d^7*tan(f*x + e) - 8*B*a*b*c*d^7*tan(f*x + e) + 8*A*b^2*c*
d^7*tan(f*x + e) + 2*B*a^2*d^8*tan(f*x + e) - 2*A*a*b*d^8*tan(f*x + e) + 6*
C*b^2*c^8 - 4*C*a*b*c^7*d - 14*B*b^2*c^7*d + C*a^2*c^6*d^2 + 17*B*a*b*c^6*d
^2 + 25*A*b^2*c^6*d^2 - 7*C*b^2*c^6*d^2 - 6*B*a^2*c^5*d^3 - 36*A*a*b*c^5*d^
3 + 24*C*a*b*c^5*d^3 - 3*B*b^2*c^5*d^3 + 14*A*a^2*c^4*d^4 - 11*C*a^2*c^4*d^
4 - 10*B*a*b*c^4*d^4 + 19*A*b^2*c^4*d^4 - C*b^2*c^4*d^4 + 7*B*a^2*c^3*d^5 -
 16*A*a*b*c^3*d^5 + 4*C*a*b*c^3*d^5 - B*b^2*c^3*d^5 + 3*A*a^2*c^2*d^6 - 3*B
*a*b*c^2*d^6 + 6*A*b^2*c^2*d^6 + B*a^2*c*d^7 - 4*A*a*b*c*d^7 + A*a^2*d^8)/(
(b^3*c^9 - 3*a*b^2*c^8*d + 3*a^2*b*c^7*d^2 + 3*b^3*c^7*d^2 - a^3*c^6*d^3 -
```

```
9*a*b^2*c^6*d^3 + 9*a^2*b*c^5*d^4 + 3*b^3*c^5*d^4 - 3*a^3*c^4*d^5 - 9*a*b^2
*c^4*d^5 + 9*a^2*b*c^3*d^6 + b^3*c^3*d^6 - 3*a^3*c^2*d^7 - 3*a*b^2*c^2*d^7
+ 3*a^2*b*c*d^8 - a^3*d^9)*(d*tan(f*x + e) + c)^2))/f
```

maple [B] time = 0.53, size = 2298, normalized size = 4.72

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))^3,x)
```

```
[Out] -8/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*a*b*c^3*d^3+1/f*b^3/(a*d-
b*c)^3/(a^2+b^2)*ln(a+b*tan(f*x+e))*B*a-1/f*b^2/(a*d-b*c)^3/(a^2+b^2)*ln(a+
b*tan(f*x+e))*a^2*C+1/f/(a*d-b*c)^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*A*b*d^4-1/
f/(a*d-b*c)^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*a*d^4+1/f/(a*d-b*c)^2/(c^2+d^2
)^2/(c+d*tan(f*x+e))*C*b*c^4-1/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))
*A*a^2*d^6-1/f/(a^2+b^2)/(c^2+d^2)^3*C*arctan(tan(f*x+e))*b*d^3-3/2/f/(a^2+
b^2)/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*b*c^2*d-3/f/(a*d-b*c)^3/(c^2+d^2)^3*l
n(c+d*tan(f*x+e))*B*b^2*c^5*d+1/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e)
)*B*b^2*c^3*d^3-3/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*a^2*c^2*d^
4-3/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*b^2*c^4*d^2-3/2/f/(a^2+b
^2)/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*a*c^2*d+3/2/f/(a^2+b^2)/(c^2+d^2)^3*ln
(1+tan(f*x+e)^2)*A*b*c*d^2-3/2/f/(a^2+b^2)/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B
*a*c*d^2-1/f/(a*d-b*c)^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*C*b*c^2*d^2-2/f/(a*d-
b*c)^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*A*a*c*d^3+3/f/(a*d-b*c)^2/(c^2+d^2)^2/(
c+d*tan(f*x+e))*A*b*c^2*d^2+1/f/(a*d-b*c)^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*
a*c^2*d^2-2/f/(a*d-b*c)^2/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*b*c^3*d+3/f/(a^2+b
^2)/(c^2+d^2)^3*C*arctan(tan(f*x+e))*a*c*d^2+3/f/(a^2+b^2)/(c^2+d^2)^3*C*ar
ctan(tan(f*x+e))*b*c^2*d-3/f/(a^2+b^2)/(c^2+d^2)^3*A*arctan(tan(f*x+e))*b*c
^2*d+3/f/(a^2+b^2)/(c^2+d^2)^3*B*arctan(tan(f*x+e))*a*c^2*d+3/2/f/(a^2+b^2)
/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a*c^2*d+2/f/(a*d-b*c)^2/(c^2+d^2)^2/(c+d*
tan(f*x+e))*C*a*c*d^3-3/2/f/(a^2+b^2)/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*b*c*
d^2-3/f/(a^2+b^2)/(c^2+d^2)^3*A*arctan(tan(f*x+e))*a*c*d^2+3/f/(a*d-b*c)^3/
(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*a^2*c^2*d^4+6/f/(a*d-b*c)^3/(c^2+d^2)^3*ln
(c+d*tan(f*x+e))*A*b^2*c^4*d^2+3/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e
))*A*b^2*c^2*d^4-1/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*a^2*c^3*d
^3+3/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*a^2*c*d^5-1/f/(a*d-b*c)
^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*a*b*d^6-3/f/(a^2+b^2)/(c^2+d^2)^3*B*arc
tan(tan(f*x+e))*b*c*d^2+1/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*b^
2*d^6+1/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*a^2*d^6+1/f/(a*d-b*c
)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*b^2*c^6+1/2/f/(a*d-b*c)/(c^2+d^2)/(c+d
*tan(f*x+e))^2*B*c*d+1/2/f/(a^2+b^2)/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*a*d^3
-1/2/f/(a^2+b^2)/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*b*c^3+1/2/f/(a^2+b^2)/(c^
2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a*c^3+1/2/f/(a^2+b^2)/(c^2+d^2)^3*ln(1+tan(f*
x+e)^2)*B*b*d^3-1/2/f/(a^2+b^2)/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a*d^3+1/2/
```

```
f/(a^2+b^2)/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*b*c^3+1/f/(a^2+b^2)/(c^2+d^2)^
3*A*arctan(tan(f*x+e))*a*c^3+1/f/(a^2+b^2)/(c^2+d^2)^3*A*arctan(tan(f*x+e))
*b*d^3-1/f/(a^2+b^2)/(c^2+d^2)^3*B*arctan(tan(f*x+e))*a*d^3+1/f/(a^2+b^2)/(
c^2+d^2)^3*B*arctan(tan(f*x+e))*b*c^3-1/f/(a^2+b^2)/(c^2+d^2)^3*C*arctan(ta
n(f*x+e))*a*c^3-1/2/f/(a*d-b*c)/(c^2+d^2)/(c+d*tan(f*x+e))^2*A*d^2-1/2/f/(a
*d-b*c)/(c^2+d^2)/(c+d*tan(f*x+e))^2*c^2*C-1/f*b^4/(a*d-b*c)^3/(a^2+b^2)*ln
(a+b*tan(f*x+e))*A+3/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*a*b*c^4
*d^2-6/f/(a*d-b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*a*b*c^2*d^4+8/f/(a*d-
b*c)^3/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*a*b*c^3*d^3
```

maxima [B] time = 0.56, size = 1078, normalized size = 2.21

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a + B*b)*c^3 + 3*(B*a - (A - C)*b)*c^2*d - 3*((A - C)*a +
B*b)*c*d^2 - (B*a - (A - C)*b)*d^3)*(f*x + e)/((a^2 + b^2)*c^6 + 3*(a^2 + b
^2)*c^4*d^2 + 3*(a^2 + b^2)*c^2*d^4 + (a^2 + b^2)*d^6) + 2*(C*a^2*b^2 - B*a
*b^3 + A*b^4)*log(b*tan(f*x + e) + a)/((a^2*b^3 + b^5)*c^3 - 3*(a^3*b^2 + a
*b^4)*c^2*d + 3*(a^4*b + a^2*b^3)*c*d^2 - (a^5 + a^3*b^2)*d^3) - 2*(C*b^2*c
^6 - 3*B*b^2*c^5*d + 3*B*a^2*c*d^5 + 3*(B*a*b + (2*A - C)*b^2)*c^4*d^2 - (B
*a^2 + 8*(A - C)*a*b - B*b^2)*c^3*d^3 + 3*((A - C)*a^2 - 2*B*a*b + A*b^2)*c
^2*d^4 - ((A - C)*a^2 + B*a*b - A*b^2)*d^6)*log(d*tan(f*x + e) + c)/(b^3*c^
9 - 3*a*b^2*c^8*d + 3*a^2*b*c*d^8 - a^3*d^9 + 3*(a^2*b + b^3)*c^7*d^2 - (a^
3 + 9*a*b^2)*c^6*d^3 + 3*(3*a^2*b + b^3)*c^5*d^4 - 3*(a^3 + 3*a*b^2)*c^4*d^
5 + (9*a^2*b + b^3)*c^3*d^6 - 3*(a^3 + a*b^2)*c^2*d^7) + ((B*a - (A - C)*b)
*c^3 - 3*((A - C)*a + B*b)*c^2*d - 3*(B*a - (A - C)*b)*c*d^2 + ((A - C)*a +
 B*b)*d^3)*log(tan(f*x + e)^2 + 1)/((a^2 + b^2)*c^6 + 3*(a^2 + b^2)*c^4*d^2
 + 3*(a^2 + b^2)*c^2*d^4 + (a^2 + b^2)*d^6) + (3*C*b*c^5 - A*a*d^5 - (C*a +
 5*B*b)*c^4*d + (3*B*a + (7*A - C)*b)*c^3*d^2 - ((5*A - 3*C)*a + B*b)*c^2*d
^3 - (B*a - 3*A*b)*c*d^4 + 2*(C*b*c^4*d - 2*B*b*c^3*d^2 - 2*(A - C)*a*c*d^4
 + (B*a + (3*A - C)*b)*c^2*d^3 - (B*a - A*b)*d^5)*tan(f*x + e))/(b^2*c^8 -
2*a*b*c^7*d - 4*a*b*c^5*d^3 - 2*a*b*c^3*d^5 + a^2*c^2*d^6 + (a^2 + 2*b^2)*c
^6*d^2 + (2*a^2 + b^2)*c^4*d^4 + (b^2*c^6*d^2 - 2*a*b*c^5*d^3 - 4*a*b*c^3*d
^5 - 2*a*b*c*d^7 + a^2*d^8 + (a^2 + 2*b^2)*c^4*d^4 + (2*a^2 + b^2)*c^2*d^6)
*tan(f*x + e)^2 + 2*(b^2*c^7*d - 2*a*b*c^6*d^2 - 4*a*b*c^4*d^4 - 2*a*b*c^2*
d^6 + a^2*c*d^7 + (a^2 + 2*b^2)*c^5*d^3 + (2*a^2 + b^2)*c^3*d^5)*tan(f*x +
e)))/f
```

mupad [B] time = 24.61, size = 65817, normalized size = 135.15

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))^3),x)
```

```
[Out] (symsum(log(- root(480*a^9*b*c^7*d^11*f^4 + 480*a*b^9*c^11*d^7*f^4 + 360*a^
9*b*c^9*d^9*f^4 + 360*a^9*b*c^5*d^13*f^4 + 360*a*b^9*c^13*d^5*f^4 + 360*a*b
^9*c^9*d^9*f^4 + 144*a^9*b*c^11*d^7*f^4 + 144*a^9*b*c^3*d^15*f^4 + 144*a*b^
9*c^15*d^3*f^4 + 144*a*b^9*c^7*d^11*f^4 + 48*a^7*b^3*c*d^17*f^4 + 48*a^3*b^
7*c^17*d*f^4 + 24*a^9*b*c^13*d^5*f^4 + 24*a^5*b^5*c^17*d*f^4 + 24*a^5*b^5*c
*d^17*f^4 + 24*a*b^9*c^5*d^13*f^4 + 24*a^9*b*c*d^17*f^4 + 24*a*b^9*c^17*d*f
^4 + 3920*a^5*b^5*c^9*d^9*f^4 - 3360*a^6*b^4*c^8*d^10*f^4 - 3360*a^4*b^6*c^
10*d^8*f^4 - 3024*a^6*b^4*c^10*d^8*f^4 + 3024*a^5*b^5*c^11*d^7*f^4 + 3024*a
^5*b^5*c^7*d^11*f^4 - 3024*a^4*b^6*c^8*d^10*f^4 + 2320*a^7*b^3*c^9*d^9*f^4
+ 2320*a^3*b^7*c^9*d^9*f^4 - 2240*a^6*b^4*c^6*d^12*f^4 - 2240*a^4*b^6*c^12*
d^6*f^4 + 2160*a^7*b^3*c^7*d^11*f^4 + 2160*a^3*b^7*c^11*d^7*f^4 - 1624*a^6*
b^4*c^12*d^6*f^4 - 1624*a^4*b^6*c^6*d^12*f^4 + 1488*a^7*b^3*c^11*d^7*f^4 +
1488*a^3*b^7*c^7*d^11*f^4 + 1344*a^5*b^5*c^13*d^5*f^4 + 1344*a^5*b^5*c^5*d^
13*f^4 - 1320*a^8*b^2*c^8*d^10*f^4 - 1320*a^2*b^8*c^10*d^8*f^4 + 1200*a^7*b
^3*c^5*d^13*f^4 + 1200*a^3*b^7*c^13*d^5*f^4 - 1060*a^8*b^2*c^6*d^12*f^4 - 1
060*a^2*b^8*c^12*d^6*f^4 - 948*a^8*b^2*c^10*d^8*f^4 - 948*a^2*b^8*c^8*d^10*
f^4 - 840*a^6*b^4*c^4*d^14*f^4 - 840*a^4*b^6*c^14*d^4*f^4 + 528*a^7*b^3*c^1
3*d^5*f^4 + 528*a^3*b^7*c^5*d^13*f^4 - 480*a^8*b^2*c^4*d^14*f^4 - 480*a^6*b
^4*c^14*d^4*f^4 - 480*a^4*b^6*c^4*d^14*f^4 - 480*a^2*b^8*c^14*d^4*f^4 - 368
*a^8*b^2*c^12*d^6*f^4 + 368*a^7*b^3*c^3*d^15*f^4 + 368*a^3*b^7*c^15*d^3*f^4
 - 368*a^2*b^8*c^6*d^12*f^4 + 304*a^5*b^5*c^15*d^3*f^4 + 304*a^5*b^5*c^3*d^
15*f^4 - 144*a^6*b^4*c^2*d^16*f^4 - 144*a^4*b^6*c^16*d^2*f^4 - 108*a^8*b^2*
c^2*d^16*f^4 - 108*a^2*b^8*c^16*d^2*f^4 + 80*a^7*b^3*c^15*d^3*f^4 + 80*a^3*
b^7*c^3*d^15*f^4 - 60*a^8*b^2*c^14*d^4*f^4 - 60*a^6*b^4*c^16*d^2*f^4 - 60*a
^4*b^6*c^2*d^16*f^4 - 60*a^2*b^8*c^4*d^14*f^4 - 80*b^10*c^12*d^6*f^4 - 60*b
^10*c^14*d^4*f^4 - 60*b^10*c^10*d^8*f^4 - 24*b^10*c^16*d^2*f^4 - 24*b^10*c^
8*d^10*f^4 - 4*b^10*c^6*d^12*f^4 - 80*a^10*c^6*d^12*f^4 - 60*a^10*c^8*d^10*
f^4 - 60*a^10*c^4*d^14*f^4 - 24*a^10*c^10*d^8*f^4 - 24*a^10*c^2*d^16*f^4 -
4*a^10*c^12*d^6*f^4 - 8*a^8*b^2*d^18*f^4 - 4*a^6*b^4*d^18*f^4 - 8*a^2*b^8*c
^18*f^4 - 4*a^4*b^6*c^18*f^4 - 4*b^10*c^18*f^4 - 4*a^10*d^18*f^4 - 12*A*C*a
^7*b*c*d^11*f^2 - 12*A*C*a*b^7*c^11*d*f^2 - 912*B*C*a^4*b^4*c^5*d^7*f^2 + 7
92*B*C*a^5*b^3*c^4*d^8*f^2 - 792*B*C*a^3*b^5*c^8*d^4*f^2 + 720*B*C*a^4*b^4*
c^7*d^5*f^2 - 480*B*C*a^6*b^2*c^5*d^7*f^2 - 408*B*C*a^2*b^6*c^5*d^7*f^2 + 3
84*B*C*a^2*b^6*c^7*d^5*f^2 - 336*B*C*a^5*b^3*c^8*d^4*f^2 + 324*B*C*a^3*b^5*
c^4*d^8*f^2 + 312*B*C*a^6*b^2*c^7*d^5*f^2 - 248*B*C*a^6*b^2*c^3*d^9*f^2 + 2
16*B*C*a^2*b^6*c^9*d^3*f^2 - 196*B*C*a^4*b^4*c^3*d^9*f^2 + 132*B*C*a^4*b^4*
c^9*d^3*f^2 + 80*B*C*a^3*b^5*c^6*d^6*f^2 - 64*B*C*a^5*b^3*c^6*d^6*f^2 - 36*
B*C*a^3*b^5*c^2*d^10*f^2 - 28*B*C*a^2*b^6*c^3*d^9*f^2 + 12*B*C*a^5*b^3*c^10
*d^2*f^2 - 12*B*C*a^5*b^3*c^2*d^10*f^2 - 12*B*C*a^3*b^5*c^10*d^2*f^2 - 4*B*
C*a^6*b^2*c^9*d^3*f^2 - 1468*A*C*a^4*b^4*c^6*d^6*f^2 + 996*A*C*a^3*b^5*c^7*
```

d^5*f^2 + 900*A*C*a^5*b^3*c^5*d^7*f^2 - 676*A*C*a^6*b^2*c^6*d^6*f^2 - 660*A
*C*a^2*b^6*c^6*d^6*f^2 + 636*A*C*a^3*b^5*c^5*d^7*f^2 + 540*A*C*a^5*b^3*c^7*
d^5*f^2 - 236*A*C*a^5*b^3*c^3*d^9*f^2 - 204*A*C*a^3*b^5*c^9*d^3*f^2 + 156*A
*C*a^2*b^6*c^10*d^2*f^2 + 132*A*C*a^6*b^2*c^2*d^10*f^2 - 72*A*C*a^6*b^2*c^4
*d^8*f^2 - 72*A*C*a^5*b^3*c^9*d^3*f^2 + 66*A*C*a^2*b^6*c^4*d^8*f^2 + 54*A*C
*a^4*b^4*c^10*d^2*f^2 + 54*A*C*a^4*b^4*c^2*d^10*f^2 - 48*A*C*a^4*b^4*c^4*d^
8*f^2 - 48*A*C*a^2*b^6*c^8*d^4*f^2 + 42*A*C*a^6*b^2*c^8*d^4*f^2 - 40*A*C*a^
3*b^5*c^3*d^9*f^2 - 36*A*C*a^4*b^4*c^8*d^4*f^2 + 24*A*C*a^2*b^6*c^2*d^10*f^
2 + 960*A*B*a^4*b^4*c^5*d^7*f^2 - 864*A*B*a^5*b^3*c^4*d^8*f^2 + 756*A*B*a^3
*b^5*c^8*d^4*f^2 - 744*A*B*a^4*b^4*c^7*d^5*f^2 - 528*A*B*a^3*b^5*c^4*d^8*f^
2 + 504*A*B*a^6*b^2*c^5*d^7*f^2 - 432*A*B*a^2*b^6*c^7*d^5*f^2 + 432*A*B*a^2
*b^6*c^5*d^7*f^2 + 348*A*B*a^5*b^3*c^8*d^4*f^2 - 312*A*B*a^6*b^2*c^7*d^5*f^
2 - 284*A*B*a^2*b^6*c^9*d^3*f^2 + 280*A*B*a^6*b^2*c^3*d^9*f^2 + 264*A*B*a^4
*b^4*c^3*d^9*f^2 - 240*A*B*a^3*b^5*c^6*d^6*f^2 - 172*A*B*a^4*b^4*c^9*d^3*f^
2 + 68*A*B*a^2*b^6*c^3*d^9*f^2 - 60*A*B*a^3*b^5*c^2*d^10*f^2 + 24*A*B*a^5*b
^3*c^6*d^6*f^2 - 24*A*B*a^5*b^3*c^2*d^10*f^2 + 12*A*B*a^3*b^5*c^10*d^2*f^2
+ 360*B*C*a^7*b*c^4*d^8*f^2 - 336*B*C*a*b^7*c^8*d^4*f^2 + 168*B*C*a*b^7*c^6
*d^6*f^2 - 136*B*C*a^7*b*c^6*d^6*f^2 + 36*B*C*a^6*b^2*c*d^11*f^2 - 36*B*C*a
^2*b^6*c^11*d*f^2 - 24*B*C*a^7*b*c^2*d^10*f^2 + 24*B*C*a*b^7*c^10*d^2*f^2 -
12*B*C*a^4*b^4*c^11*d*f^2 + 12*B*C*a^4*b^4*c*d^11*f^2 + 12*B*C*a*b^7*c^4*d
^8*f^2 + 444*A*C*a*b^7*c^7*d^5*f^2 + 348*A*C*a^7*b*c^5*d^7*f^2 - 164*A*C*a^
7*b*c^3*d^9*f^2 - 132*A*C*a*b^7*c^9*d^3*f^2 + 84*A*C*a*b^7*c^5*d^7*f^2 + 32
*A*C*a*b^7*c^3*d^9*f^2 - 12*A*C*a^7*b*c^7*d^5*f^2 - 12*A*C*a^5*b^3*c*d^11*f
^2 - 12*A*C*a^3*b^5*c^11*d*f^2 - 360*A*B*a^7*b*c^4*d^8*f^2 + 288*A*B*a*b^7*
c^8*d^4*f^2 - 288*A*B*a*b^7*c^6*d^6*f^2 - 144*A*B*a*b^7*c^4*d^8*f^2 + 136*A
*B*a^7*b*c^6*d^6*f^2 - 60*A*B*a*b^7*c^2*d^10*f^2 - 36*A*B*a*b^7*c^10*d^2*f^
2 + 24*A*B*a^7*b*c^2*d^10*f^2 - 24*A*B*a^6*b^2*c*d^11*f^2 + 12*A*B*a^4*b^4*
c*d^11*f^2 + 12*A*B*a^2*b^6*c^11*d*f^2 + 12*A*B*a^2*b^6*c*d^11*f^2 + 80*B*C
*b^8*c^9*d^3*f^2 - 24*B*C*b^8*c^7*d^5*f^2 - 90*A*C*b^8*c^8*d^4*f^2 - 80*B*C
*a^8*c^3*d^9*f^2 + 54*A*C*b^8*c^10*d^2*f^2 - 30*A*C*b^8*c^6*d^6*f^2 + 24*B*
C*a^8*c^5*d^7*f^2 - 12*A*C*b^8*c^4*d^8*f^2 - 112*A*B*b^8*c^9*d^3*f^2 - 66*A
*C*a^8*c^4*d^8*f^2 + 54*A*C*a^8*c^2*d^10*f^2 - 8*B*C*a^5*b^3*d^12*f^2 - 8*B
*C*a^3*b^5*d^12*f^2 + 4*A*B*b^8*c^3*d^9*f^2 + 2*A*C*a^8*c^6*d^6*f^2 + 80*A*
B*a^8*c^3*d^9*f^2 - 24*A*B*a^8*c^5*d^7*f^2 + 8*A*C*a^2*b^6*d^12*f^2 - 4*B*C
*a^3*b^5*c^12*f^2 + 4*A*C*a^4*b^4*d^12*f^2 - 2*A*C*a^6*b^2*d^12*f^2 + 6*A*C
*a^2*b^6*c^12*f^2 + 4*A*B*a^5*b^3*d^12*f^2 - 4*A*B*a^3*b^5*d^12*f^2 + 726*C
^2*a^4*b^4*c^6*d^6*f^2 - 402*C^2*a^5*b^3*c^5*d^7*f^2 - 402*C^2*a^3*b^5*c^7*
d^5*f^2 + 322*C^2*a^6*b^2*c^6*d^6*f^2 + 322*C^2*a^2*b^6*c^6*d^6*f^2 - 222*C
^2*a^5*b^3*c^7*d^5*f^2 - 222*C^2*a^3*b^5*c^5*d^7*f^2 + 134*C^2*a^5*b^3*c^3*
d^9*f^2 + 134*C^2*a^3*b^5*c^9*d^3*f^2 - 66*C^2*a^6*b^2*c^2*d^10*f^2 - 66*C^
2*a^2*b^6*c^10*d^2*f^2 + 52*C^2*a^5*b^3*c^9*d^3*f^2 + 52*C^2*a^3*b^5*c^3*d^
9*f^2 - 27*C^2*a^6*b^2*c^8*d^4*f^2 - 27*C^2*a^2*b^6*c^4*d^8*f^2 + 24*C^2*a^
6*b^2*c^4*d^8*f^2 + 24*C^2*a^4*b^4*c^8*d^4*f^2 + 24*C^2*a^4*b^4*c^4*d^8*f^2
+ 24*C^2*a^2*b^6*c^8*d^4*f^2 - 15*C^2*a^4*b^4*c^10*d^2*f^2 - 15*C^2*a^4*b^
4*c^2*d^10*f^2 - 570*B^2*a^4*b^4*c^6*d^6*f^2 + 366*B^2*a^3*b^5*c^7*d^5*f^2

+ 318*B^2*a^5*b^3*c^5*d^7*f^2 - 262*B^2*a^6*b^2*c^6*d^6*f^2 - 222*B^2*a^2*b
^6*c^6*d^6*f^2 - 210*B^2*a^5*b^3*c^3*d^9*f^2 + 186*B^2*a^5*b^3*c^7*d^5*f^2
+ 162*B^2*a^3*b^5*c^5*d^7*f^2 - 142*B^2*a^3*b^5*c^9*d^3*f^2 + 132*B^2*a^4*b
^4*c^4*d^8*f^2 + 117*B^2*a^2*b^6*c^4*d^8*f^2 + 102*B^2*a^6*b^2*c^2*d^10*f^2
 - 96*B^2*a^3*b^5*c^3*d^9*f^2 + 90*B^2*a^2*b^6*c^10*d^2*f^2 + 81*B^2*a^4*b^
4*c^2*d^10*f^2 - 56*B^2*a^5*b^3*c^9*d^3*f^2 + 48*B^2*a^6*b^2*c^4*d^8*f^2 +
48*B^2*a^4*b^4*c^8*d^4*f^2 + 45*B^2*a^6*b^2*c^8*d^4*f^2 + 36*B^2*a^2*b^6*c^
8*d^4*f^2 + 36*B^2*a^2*b^6*c^2*d^10*f^2 + 33*B^2*a^4*b^4*c^10*d^2*f^2 + 822
*A^2*a^4*b^4*c^6*d^6*f^2 - 594*A^2*a^3*b^5*c^7*d^5*f^2 - 498*A^2*a^5*b^3*c^
5*d^7*f^2 + 498*A^2*a^2*b^6*c^6*d^6*f^2 - 414*A^2*a^3*b^5*c^5*d^7*f^2 + 354
*A^2*a^6*b^2*c^6*d^6*f^2 - 318*A^2*a^5*b^3*c^7*d^5*f^2 + 144*A^2*a^2*b^6*c^
8*d^4*f^2 + 102*A^2*a^5*b^3*c^3*d^9*f^2 + 84*A^2*a^4*b^4*c^4*d^8*f^2 + 81*A
^2*a^2*b^6*c^4*d^8*f^2 + 72*A^2*a^4*b^4*c^8*d^4*f^2 + 70*A^2*a^3*b^5*c^9*d^
3*f^2 - 66*A^2*a^6*b^2*c^2*d^10*f^2 + 48*A^2*a^6*b^2*c^4*d^8*f^2 - 42*A^2*a
^2*b^6*c^10*d^2*f^2 + 24*A^2*a^2*b^6*c^2*d^10*f^2 + 20*A^2*a^5*b^3*c^9*d^3*
f^2 - 15*A^2*a^6*b^2*c^8*d^4*f^2 - 15*A^2*a^4*b^4*c^10*d^2*f^2 - 15*A^2*a^4
*b^4*c^2*d^10*f^2 - 12*A^2*a^3*b^5*c^3*d^9*f^2 - 24*B*C*b^8*c^11*d*f^2 + 24
*B*C*a^8*c*d^11*f^2 + 12*A*B*b^8*c^11*d*f^2 - 8*B*C*a^7*b*d^12*f^2 - 24*A*B
*a^8*c*d^11*f^2 + 4*B*C*a*b^7*c^12*f^2 + 8*A*B*a^7*b*d^12*f^2 - 8*A*B*a*b^7
*d^12*f^2 - 8*A*B*a*b^7*c^12*f^2 - 174*C^2*a^7*b*c^5*d^7*f^2 - 174*C^2*a*b^
7*c^7*d^5*f^2 + 82*C^2*a^7*b*c^3*d^9*f^2 + 82*C^2*a*b^7*c^9*d^3*f^2 + 6*C^2
*a^7*b*c^7*d^5*f^2 + 6*C^2*a^5*b^3*c*d^11*f^2 + 6*C^2*a^3*b^5*c^11*d*f^2 +
6*C^2*a*b^7*c^5*d^7*f^2 + 162*B^2*a*b^7*c^7*d^5*f^2 + 138*B^2*a^7*b*c^5*d^7
*f^2 - 118*B^2*a^7*b*c^3*d^9*f^2 - 86*B^2*a*b^7*c^9*d^3*f^2 - 30*B^2*a^5*b^
3*c*d^11*f^2 - 18*B^2*a^7*b*c^7*d^5*f^2 - 18*B^2*a*b^7*c^5*d^7*f^2 - 12*B^2
*a^3*b^5*c*d^11*f^2 - 6*B^2*a^3*b^5*c^11*d*f^2 - 4*B^2*a*b^7*c^3*d^9*f^2 -
270*A^2*a*b^7*c^7*d^5*f^2 - 174*A^2*a^7*b*c^5*d^7*f^2 - 90*A^2*a*b^7*c^5*d^
7*f^2 + 82*A^2*a^7*b*c^3*d^9*f^2 + 50*A^2*a*b^7*c^9*d^3*f^2 - 32*A^2*a*b^7*
c^3*d^9*f^2 + 6*A^2*a^7*b*c^7*d^5*f^2 + 6*A^2*a^5*b^3*c*d^11*f^2 + 6*A^2*a^
3*b^5*c^11*d*f^2 + 6*C^2*a^7*b*c*d^11*f^2 + 6*C^2*a*b^7*c^11*d*f^2 - 18*B^2
*a^7*b*c*d^11*f^2 - 6*B^2*a*b^7*c^11*d*f^2 + 6*A^2*a^7*b*c*d^11*f^2 + 6*A^2
*a*b^7*c^11*d*f^2 - 6*A*C*a^8*d^12*f^2 - 2*A*C*b^8*c^12*f^2 + 33*C^2*b^8*c^
8*d^4*f^2 - 27*C^2*b^8*c^10*d^2*f^2 - C^2*b^8*c^6*d^6*f^2 + 33*C^2*a^8*c^4*
d^8*f^2 + 33*B^2*b^8*c^10*d^2*f^2 - 27*C^2*a^8*c^2*d^10*f^2 - 27*B^2*b^8*c^
8*d^4*f^2 + 3*B^2*b^8*c^6*d^6*f^2 - C^2*a^8*c^6*d^6*f^2 + 117*A^2*b^8*c^8*d
^4*f^2 + 111*A^2*b^8*c^6*d^6*f^2 + 72*A^2*b^8*c^4*d^8*f^2 + 33*B^2*a^8*c^2*
d^10*f^2 - 27*B^2*a^8*c^4*d^8*f^2 + 24*A^2*b^8*c^2*d^10*f^2 + 4*C^2*a^4*b^4
*d^12*f^2 + 3*C^2*a^6*b^2*d^12*f^2 + 3*B^2*a^8*c^6*d^6*f^2 - 3*A^2*b^8*c^10
*d^2*f^2 + 33*A^2*a^8*c^4*d^8*f^2 - 27*A^2*a^8*c^2*d^10*f^2 + 4*C^2*a^4*b^4
*c^12*f^2 + 4*B^2*a^4*b^4*d^12*f^2 + 4*B^2*a^2*b^6*d^12*f^2 + 3*C^2*a^2*b^6
*c^12*f^2 + 3*B^2*a^6*b^2*d^12*f^2 - A^2*a^8*c^6*d^6*f^2 - 4*A^2*a^4*b^4*d^
12*f^2 + 3*B^2*a^2*b^6*c^12*f^2 - A^2*a^6*b^2*d^12*f^2 - A^2*a^2*b^6*c^12*f
^2 + 3*C^2*b^8*c^12*f^2 + 3*C^2*a^8*d^12*f^2 + 4*A^2*b^8*d^12*f^2 - B^2*b^8
*c^12*f^2 - B^2*a^8*d^12*f^2 + 3*A^2*b^8*c^12*f^2 + 3*A^2*a^8*d^12*f^2 - 24
*A*B*C*a*b^6*c*d^8*f + 342*A*B*C*a^2*b^5*c^4*d^5*f - 186*A*B*C*a^3*b^4*c^5*

d^4*f - 66*A*B*C*a^4*b^3*c^2*d^7*f + 48*A*B*C*a^2*b^5*c^2*d^7*f + 42*A*B*C*
a^2*b^5*c^6*d^3*f + 26*A*B*C*a^5*b^2*c^3*d^6*f + 24*A*B*C*a^4*b^3*c^6*d^3*f
 - 18*A*B*C*a^4*b^3*c^4*d^5*f - 18*A*B*C*a^3*b^4*c^7*d^2*f - 8*A*B*C*a^3*b^
4*c^3*d^6*f + 6*A*B*C*a^5*b^2*c^5*d^4*f - 128*A*B*C*a*b^6*c^3*d^6*f + 126*A
*B*C*a*b^6*c^7*d^2*f + 72*A*B*C*a^3*b^4*c*d^8*f - 36*A*B*C*a^5*b^2*c*d^8*f
 - 36*A*B*C*a^2*b^5*c^8*d*f + 30*A*B*C*a^6*b*c^2*d^7*f - 12*A*B*C*a^6*b*c^4*
d^5*f - 12*A*B*C*a*b^6*c^5*d^4*f - 21*B^2*C*a*b^6*c^8*d*f - 3*B^2*C*a^6*b*c
*d^8*f + 21*A^2*C*a*b^6*c^8*d*f - 21*A*C^2*a*b^6*c^8*d*f - 9*A^2*C*a^6*b*c*
d^8*f + 9*A*C^2*a^6*b*c*d^8*f + 36*A^2*B*a*b^6*c*d^8*f + 21*A*B^2*a*b^6*c^8
*d*f + 3*A*B^2*a^6*b*c*d^8*f - 78*A*B*C*b^7*c^6*d^3*f + 24*A*B*C*b^7*c^4*d^
5*f + 2*A*B*C*a^7*c^3*d^6*f + 16*A*B*C*a^4*b^3*d^9*f - 16*A*B*C*a^2*b^5*d^9
*f - 237*B^2*C*a^3*b^4*c^4*d^5*f + 165*B*C^2*a^3*b^4*c^5*d^4*f + 92*B^2*C*a
^2*b^5*c^3*d^6*f - 81*B^2*C*a^2*b^5*c^7*d^2*f + 77*B^2*C*a^4*b^3*c^3*d^6*f
 - 75*B*C^2*a^2*b^5*c^4*d^5*f + 69*B^2*C*a^4*b^3*c^5*d^4*f + 69*B*C^2*a^4*b^
3*c^4*d^5*f - 68*B*C^2*a^3*b^4*c^3*d^6*f - 63*B^2*C*a^5*b^2*c^4*d^5*f - 61*
B*C^2*a^2*b^5*c^6*d^3*f + 57*B*C^2*a^4*b^3*c^2*d^7*f - 53*B*C^2*a^5*b^2*c^3
*d^6*f - 44*B*C^2*a^4*b^3*c^6*d^3*f - 36*B^2*C*a^3*b^4*c^2*d^7*f + 35*B^2*C
*a^3*b^4*c^6*d^3*f + 33*B^2*C*a^5*b^2*c^2*d^7*f - 33*B^2*C*a^2*b^5*c^5*d^4*
f + 33*B*C^2*a^3*b^4*c^7*d^2*f - 12*B^2*C*a^4*b^3*c^7*d^2*f + 9*B*C^2*a^5*b
^2*c^5*d^4*f + 4*B^2*C*a^5*b^2*c^6*d^3*f + 225*A^2*C*a^2*b^5*c^5*d^4*f - 10
5*A*C^2*a^2*b^5*c^5*d^4*f - 99*A^2*C*a^3*b^4*c^4*d^5*f - 81*A^2*C*a^5*b^2*c
^4*d^5*f + 67*A^2*C*a^4*b^3*c^3*d^6*f - 59*A*C^2*a^4*b^3*c^3*d^6*f + 57*A*C
^2*a^5*b^2*c^2*d^7*f - 57*A*C^2*a^2*b^5*c^7*d^2*f + 51*A^2*C*a^4*b^3*c^5*d^
4*f + 48*A^2*C*a^3*b^4*c^2*d^7*f + 45*A*C^2*a^5*b^2*c^4*d^5*f - 35*A^2*C*a^
3*b^4*c^6*d^3*f - 33*A^2*C*a^5*b^2*c^2*d^7*f + 33*A^2*C*a^2*b^5*c^7*d^2*f +
 33*A*C^2*a^4*b^3*c^5*d^4*f + 27*A*C^2*a^3*b^4*c^6*d^3*f - 24*A*C^2*a^3*b^4
*c^2*d^7*f + 24*A*C^2*a^2*b^5*c^3*d^6*f - 21*A*C^2*a^3*b^4*c^4*d^5*f - 16*A
^2*C*a^2*b^5*c^3*d^6*f - 243*A^2*B*a^2*b^5*c^4*d^5*f - 156*A*B^2*a^2*b^5*c^
3*d^6*f + 141*A*B^2*a^3*b^4*c^4*d^5*f + 108*A^2*B*a^3*b^4*c^3*d^6*f - 105*A
*B^2*a^4*b^3*c^3*d^6*f + 84*A*B^2*a^3*b^4*c^2*d^7*f + 81*A*B^2*a^2*b^5*c^5*
d^4*f - 51*A^2*B*a^4*b^3*c^4*d^5*f + 51*A^2*B*a^2*b^5*c^6*d^3*f - 48*A^2*B*
a^2*b^5*c^2*d^7*f + 45*A^2*B*a^3*b^4*c^5*d^4*f + 39*A*B^2*a^5*b^2*c^4*d^5*f
 - 35*A*B^2*a^3*b^4*c^6*d^3*f + 33*A*B^2*a^2*b^5*c^7*d^2*f + 27*A^2*B*a^5*b
^2*c^3*d^6*f - 21*A*B^2*a^4*b^3*c^5*d^4*f + 20*A^2*B*a^4*b^3*c^6*d^3*f - 15
*A^2*B*a^5*b^2*c^5*d^4*f - 15*A^2*B*a^3*b^4*c^7*d^2*f + 9*A^2*B*a^4*b^3*c^2
*d^7*f + 3*A*B^2*a^5*b^2*c^2*d^7*f + 18*A*B*C*b^7*c^8*d*f - 6*A*B*C*a^7*c*d
^8*f + 2*A*B*C*a^6*b*d^9*f - 6*A*B*C*a*b^6*c^9*f + 63*B^2*C*a*b^6*c^6*d^3*f
 - 48*B^2*C*a^4*b^3*c*d^8*f + 42*B*C^2*a^2*b^5*c^8*d*f + 42*B*C^2*a*b^6*c^5
*d^4*f - 39*B*C^2*a*b^6*c^7*d^2*f + 30*B*C^2*a^5*b^2*c*d^8*f - 24*B^2*C*a*b
^6*c^4*d^5*f - 24*B*C^2*a^3*b^4*c*d^8*f + 17*B^2*C*a^6*b*c^3*d^6*f - 15*B*C
^2*a^6*b*c^2*d^7*f + 12*B^2*C*a^3*b^4*c^8*d*f + 12*B^2*C*a^2*b^5*c*d^8*f +
6*B*C^2*a^6*b*c^4*d^5*f - 192*A^2*C*a*b^6*c^4*d^5*f - 99*A^2*C*a*b^6*c^6*d^
3*f + 84*A*C^2*a*b^6*c^4*d^5*f + 59*A*C^2*a*b^6*c^6*d^3*f + 51*A^2*C*a^6*b*
c^3*d^6*f - 51*A*C^2*a^6*b*c^3*d^6*f - 36*A^2*C*a^2*b^5*c*d^8*f - 24*A*C^2*
a^4*b^3*c*d^8*f + 24*A*C^2*a^2*b^5*c*d^8*f + 12*A^2*C*a^4*b^3*c*d^8*f + 12*

A*C^2*a^3*b^4*c^8*d*f + 160*A^2*B*a*b^6*c^3*d^6*f - 99*A*B^2*a*b^6*c^6*d^3*
f - 87*A^2*B*a*b^6*c^7*d^2*f - 72*A*B^2*a*b^6*c^4*d^5*f - 48*A*B^2*a^2*b^5*
c*d^8*f - 36*A^2*B*a^3*b^4*c*d^8*f + 24*A*B^2*a^4*b^3*c*d^8*f - 17*A*B^2*a^
6*b*c^3*d^6*f - 15*A^2*B*a^6*b*c^2*d^7*f + 12*A*B^2*a*b^6*c^2*d^7*f + 6*A^2
*B*a^6*b*c^4*d^5*f + 6*A^2*B*a^5*b^2*c*d^8*f + 6*A^2*B*a^2*b^5*c^8*d*f - 6*
A^2*B*a*b^6*c^5*d^4*f + 3*B^2*C*b^7*c^7*d^2*f - B*C^2*b^7*c^6*d^3*f + 96*A^
2*C*b^7*c^5*d^4*f - 39*A^2*C*b^7*c^7*d^2*f - 36*A*C^2*b^7*c^5*d^4*f + 32*A^
2*C*b^7*c^3*d^6*f + 15*A*C^2*b^7*c^7*d^2*f - 3*B^2*C*a^7*c^2*d^7*f - B*C^2*
a^7*c^3*d^6*f + 111*A^2*B*b^7*c^6*d^3*f - 39*A*B^2*b^7*c^7*d^2*f + 24*A*B^2
*b^7*c^5*d^4*f + 12*B^2*C*a^3*b^4*d^9*f - 12*B*C^2*a^4*b^3*d^9*f - 9*A^2*C*
a^7*c^2*d^7*f + 9*A*C^2*a^7*c^2*d^7*f - 4*A*B^2*b^7*c^3*d^6*f - 12*A^2*C*a^
3*b^4*d^9*f - 8*A*C^2*a^5*b^2*d^9*f + 8*A*C^2*a^3*b^4*d^9*f + 4*B^2*C*a^2*b
^5*c^9*f + 4*A^2*C*a^5*b^2*d^9*f - 4*B*C^2*a^3*b^4*c^9*f + 3*A*B^2*a^7*c^2*
d^7*f - A^2*B*a^7*c^3*d^6*f + 12*A^2*B*a^2*b^5*d^9*f - 8*A*B^2*a^3*b^4*d^9*
f - 4*A^2*B*a^4*b^3*d^9*f + 4*A*C^2*a^2*b^5*c^9*f - 3*C^3*a^6*b*c*d^8*f + 3
*C^3*a*b^6*c^8*d*f + 3*A^3*a^6*b*c*d^8*f - 3*A^3*a*b^6*c^8*d*f + 3*B*C^2*b^
7*c^8*d*f + 12*A^2*C*b^7*c*d^8*f + 3*B*C^2*a^7*c*d^8*f - 9*A^2*B*b^7*c^8*d*
f - B*C^2*a^6*b*d^9*f + 4*A^2*C*a*b^6*d^9*f + 3*A^2*B*a^7*c*d^8*f + 3*B*C^2
*a*b^6*c^9*f + 8*A*B^2*a*b^6*d^9*f - A^2*B*a^6*b*d^9*f - A^2*B*a*b^6*c^9*f
- 39*C^3*a^4*b^3*c^5*d^4*f + 39*C^3*a^3*b^4*c^4*d^5*f - 27*C^3*a^5*b^2*c^2*
d^7*f + 27*C^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^4*b^3*c^3*d^6*f - 17*C^3*a^3*b^
4*c^6*d^3*f - 3*C^3*a^5*b^2*c^4*d^5*f + 3*C^3*a^2*b^5*c^5*d^4*f - 63*B^3*a^
3*b^4*c^5*d^4*f + 57*B^3*a^2*b^5*c^4*d^5*f - 51*B^3*a^4*b^3*c^2*d^7*f + 48*
B^3*a^3*b^4*c^3*d^6*f + 31*B^3*a^2*b^5*c^6*d^3*f + 27*B^3*a^5*b^2*c^3*d^6*f
+ 16*B^3*a^4*b^3*c^6*d^3*f - 15*B^3*a^5*b^2*c^5*d^4*f - 12*B^3*a^2*b^5*c^2
*d^7*f + 9*B^3*a^4*b^3*c^4*d^5*f - 3*B^3*a^3*b^4*c^7*d^2*f - 123*A^3*a^2*b^
5*c^5*d^4*f + 81*A^3*a^3*b^4*c^4*d^5*f - 45*A^3*a^4*b^3*c^5*d^4*f + 39*A^3*
a^5*b^2*c^4*d^5*f - 25*A^3*a^4*b^3*c^3*d^6*f + 25*A^3*a^3*b^4*c^6*d^3*f - 2
4*A^3*a^3*b^4*c^2*d^7*f - 8*A^3*a^2*b^5*c^3*d^6*f + 3*A^3*a^5*b^2*c^2*d^7*f
- 3*A^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^6*b*c^3*d^6*f - 17*C^3*a*b^6*c^6*d^3*
f + 12*C^3*a^4*b^3*c*d^8*f - 12*C^3*a^3*b^4*c^8*d*f + 24*B^3*a^3*b^4*c*d^8*
f + 21*B^3*a*b^6*c^7*d^2*f - 18*B^3*a*b^6*c^5*d^4*f - 15*B^3*a^6*b*c^2*d^7*
f + 6*B^3*a^6*b*c^4*d^5*f + 6*B^3*a^5*b^2*c*d^8*f - 6*B^3*a^2*b^5*c^8*d*f +
4*B^3*a*b^6*c^3*d^6*f + 108*A^3*a*b^6*c^4*d^5*f + 57*A^3*a*b^6*c^6*d^3*f -
17*A^3*a^6*b*c^3*d^6*f + 12*A^3*a^2*b^5*c*d^8*f + 3*C^3*b^7*c^7*d^2*f - 3*
C^3*a^7*c^2*d^7*f - B^3*b^7*c^6*d^3*f - 60*A^3*b^7*c^5*d^4*f - 32*A^3*b^7*c
^3*d^6*f + 21*A^3*b^7*c^7*d^2*f + 4*C^3*a^5*b^2*d^9*f - B^3*a^7*c^3*d^6*f -
4*C^3*a^2*b^5*c^9*f - 4*B^3*a^2*b^5*d^9*f + 3*A^3*a^7*c^2*d^7*f + 4*A^3*a^
3*b^4*d^9*f + 3*B^3*b^7*c^8*d*f - 12*A^3*b^7*c*d^8*f + 3*B^3*a^7*c*d^8*f -
B^3*a^6*b*d^9*f - 4*A^3*a*b^6*d^9*f - B^3*a*b^6*c^9*f - B^2*C*b^7*c^9*f - 4
*A^2*B*b^7*d^9*f + 3*A^2*C*a^7*d^9*f - 3*A*C^2*a^7*d^9*f - A*C^2*b^7*c^9*f
- A*B^2*a^7*d^9*f - C^3*b^7*c^9*f - A^3*a^7*d^9*f + B^2*C*a^7*d^9*f + A^2*C
*b^7*c^9*f + A*B^2*b^7*c^9*f + C^3*a^7*d^9*f + A^3*b^7*c^9*f - 6*A*B^2*C*a*
b^5*c^5*d - 21*A^2*B*C*a^2*b^4*c^3*d^3 + 21*A*B*C^2*a^2*b^4*c^3*d^3 + 12*A*
B^2*C*a^2*b^4*c^4*d^2 - 12*A*B^2*C*a^2*b^4*c^2*d^4 - 10*A*B^2*C*a^3*b^3*c^3

*d^3 - 6*A*B*C^2*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^
3*b^3*c^2*d^4 + 3*A*B^2*C*a^4*b^2*c^2*d^4 + 3*A*B*C^2*a^3*b^3*c^2*d^4 + 2*A
*B*C^2*a^4*b^2*c^3*d^3 - A^2*B*C*a^4*b^2*c^3*d^3 + 18*A^2*B*C*a*b^5*c^2*d^4
 + 10*A*B^2*C*a*b^5*c^3*d^3 + 9*A^2*B*C*a*b^5*c^4*d^2 - 9*A*B*C^2*a*b^5*c^4
*d^2 - 9*A*B*C^2*a*b^5*c^2*d^4 - 6*A^2*B*C*a^2*b^4*c*d^5 + 6*A*B^2*C*a^3*b^
3*c*d^5 - 6*A*B*C^2*a^4*b^2*c*d^5 + 6*A*B*C^2*a^2*b^4*c^5*d + 3*A^2*B*C*a^4
*b^2*c*d^5 - 3*A^2*B*C*a^2*b^4*c^5*d + 3*A*B*C^2*a^2*b^4*c*d^5 + 3*B^3*C*a^
4*b^2*c*d^5 - 3*B^3*C*a^2*b^4*c^5*d + 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a*b
^5*c^5*d + 3*B*C^3*a^4*b^2*c*d^5 - 3*B*C^3*a^2*b^4*c^5*d + 3*B*C^3*a*b^5*c^
4*d^2 + 24*A^3*C*a*b^5*c^3*d^3 + 8*A*C^3*a*b^5*c^3*d^3 - 9*A^3*B*a*b^5*c^2*
d^4 - 9*A*B^3*a*b^5*c^2*d^4 + 3*A^3*B*a^2*b^4*c*d^5 - 3*A^3*B*a*b^5*c^4*d^2
 + 3*A^2*B^2*a*b^5*c^5*d + 3*A*B^3*a^2*b^4*c*d^5 - 3*A*B^3*a*b^5*c^4*d^2 -
3*A*B^2*C*b^6*c^4*d^2 - 2*A^2*B*C*b^6*c^3*d^3 + 5*A*B*C^2*a^3*b^3*d^6 - 4*A
^2*B*C*a^3*b^3*d^6 - A*B^2*C*a^4*b^2*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^
2*C^2*a^2*b^4*c^4*d^2 + 6*B^2*C^2*a^2*b^4*c^2*d^4 - 3*B^2*C^2*a^4*b^2*c^2*d
^4 + 24*A^2*C^2*a^3*b^3*c^3*d^3 - 15*A^2*C^2*a^2*b^4*c^4*d^2 - 9*A^2*C^2*a^
4*b^2*c^2*d^4 + 3*A^2*C^2*a^2*b^4*c^2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A
^2*B^2*a^2*b^4*c^4*d^2 + 6*A^2*B*C*b^6*c^5*d - 3*A*B*C^2*b^6*c^5*d + 4*A^2*
B*C*a*b^5*d^6 - 2*A*B*C^2*a*b^5*d^6 + 2*A*B*C^2*a*b^5*c^6 - A^2*B*C*a*b^5*c
^6 - 7*B^3*C*a^2*b^4*c^3*d^3 - 7*B*C^3*a^2*b^4*c^3*d^3 + 3*B^3*C*a^3*b^3*c^
4*d^2 - 3*B^3*C*a^3*b^3*c^2*d^4 - 3*B^2*C^2*a^3*b^3*c*d^5 + 3*B*C^3*a^3*b^3
*c^4*d^2 - 3*B*C^3*a^3*b^3*c^2*d^4 - B^3*C*a^4*b^2*c^3*d^3 - B^2*C^2*a*b^5*
c^3*d^3 - B*C^3*a^4*b^2*c^3*d^3 - 24*A^2*C^2*a*b^5*c^3*d^3 - 24*A*C^3*a^3*b
^3*c^3*d^3 + 12*A*C^3*a^2*b^4*c^4*d^2 + 9*A*C^3*a^4*b^2*c^2*d^4 - 8*A^3*C*a
^3*b^3*c^3*d^3 + 6*A^3*C*a^2*b^4*c^4*d^2 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*
C*a^4*b^2*c^2*d^4 - 9*A^2*B^2*a*b^5*c^3*d^3 + 7*A^3*B*a^2*b^4*c^3*d^3 + 7*A
*B^3*a^2*b^4*c^3*d^3 - 3*A^3*B*a^3*b^3*c^2*d^4 - 3*A^2*B^2*a^3*b^3*c*d^5 -
3*A*B^3*a^3*b^3*c^2*d^4 + 12*A^2*C^2*b^6*c^4*d^2 + 3*A^2*C^2*b^6*c^2*d^4 +
6*A^2*B^2*b^6*c^4*d^2 + 3*A^2*B^2*b^6*c^2*d^4 - 5*A^2*C^2*a^2*b^4*d^6 + 3*A
^2*C^2*a^4*b^2*d^6 + A*B*C^2*b^6*c^3*d^3 - 3*B^4*a^3*b^3*c*d^5 - B^4*a*b^5*
c^3*d^3 + A^2*B^2*a^3*b^3*c^3*d^3 - 8*A^4*a*b^5*c^3*d^3 - 15*A^3*C*b^6*c^4*
d^2 - 6*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 - 2*B^3*C*a^3*b^3*d^6 - 2*B
*C^3*a^3*b^3*d^6 + 4*A^3*C*a^2*b^4*d^6 - 3*A*C^3*a^4*b^2*d^6 + 2*A*C^3*a^2*
b^4*d^6 - A^3*C*a^4*b^2*d^6 - 2*A*C^3*a^2*b^4*c^6 + 3*B^4*a*b^5*c^5*d - 3*A
^3*B*b^6*c^5*d - 3*A*B^3*b^6*c^5*d - B^3*C*a*b^5*c^6 - B*C^3*a*b^5*c^6 - 2*
A^3*B*a*b^5*d^6 - 2*A*B^3*a*b^5*d^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2
*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2 + 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^2*b^4*c^
4*d^2 + 3*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*a^4*b^2*d^6 + B^2*C^2*a^2*b^4*d^6 +
 B^2*C^2*a^2*b^4*c^6 + A^2*C^2*a^2*b^4*c^6 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^
3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 + 6*A^4*b
^6*c^4*d^2 + 3*A^4*b^6*c^2*d^4 - A^4*a^2*b^4*d^6 - 2*A^2*C^2*b^6*c^6 + A*B^
2*C*b^6*c^6 + B^4*a^3*b^3*c^3*d^3 + A^3*C*b^6*c^6 + A*C^3*b^6*c^6 + C^4*a^4
*b^2*d^6 + C^4*a^2*b^4*c^6 + B^4*a^2*b^4*d^6 + A^2*C^2*b^6*d^6 + A^2*B^2*b^
6*d^6 + A^4*b^6*d^6, f, k)*(root(480*a^9*b*c^7*d^11*f^4 + 480*a*b^9*c^11*d^
7*f^4 + 360*a^9*b*c^9*d^9*f^4 + 360*a^9*b*c^5*d^13*f^4 + 360*a*b^9*c^13*d^5

*f^4 + 360*a*b^9*c^9*d^9*f^4 + 144*a^9*b*c^11*d^7*f^4 + 144*a^9*b*c^3*d^15*
f^4 + 144*a*b^9*c^15*d^3*f^4 + 144*a*b^9*c^7*d^11*f^4 + 48*a^7*b^3*c*d^17*f
^4 + 48*a^3*b^7*c^17*d*f^4 + 24*a^9*b*c^13*d^5*f^4 + 24*a^5*b^5*c^17*d*f^4
+ 24*a^5*b^5*c*d^17*f^4 + 24*a*b^9*c^5*d^13*f^4 + 24*a^9*b*c*d^17*f^4 + 24*
a*b^9*c^17*d*f^4 + 3920*a^5*b^5*c^9*d^9*f^4 - 3360*a^6*b^4*c^8*d^10*f^4 - 3
360*a^4*b^6*c^10*d^8*f^4 - 3024*a^6*b^4*c^10*d^8*f^4 + 3024*a^5*b^5*c^11*d^
7*f^4 + 3024*a^5*b^5*c^7*d^11*f^4 - 3024*a^4*b^6*c^8*d^10*f^4 + 2320*a^7*b^
3*c^9*d^9*f^4 + 2320*a^3*b^7*c^9*d^9*f^4 - 2240*a^6*b^4*c^6*d^12*f^4 - 2240
*a^4*b^6*c^12*d^6*f^4 + 2160*a^7*b^3*c^7*d^11*f^4 + 2160*a^3*b^7*c^11*d^7*f
^4 - 1624*a^6*b^4*c^12*d^6*f^4 - 1624*a^4*b^6*c^6*d^12*f^4 + 1488*a^7*b^3*c
^11*d^7*f^4 + 1488*a^3*b^7*c^7*d^11*f^4 + 1344*a^5*b^5*c^13*d^5*f^4 + 1344*
a^5*b^5*c^5*d^13*f^4 - 1320*a^8*b^2*c^8*d^10*f^4 - 1320*a^2*b^8*c^10*d^8*f^
4 + 1200*a^7*b^3*c^5*d^13*f^4 + 1200*a^3*b^7*c^13*d^5*f^4 - 1060*a^8*b^2*c^
6*d^12*f^4 - 1060*a^2*b^8*c^12*d^6*f^4 - 948*a^8*b^2*c^10*d^8*f^4 - 948*a^2
*b^8*c^8*d^10*f^4 - 840*a^6*b^4*c^4*d^14*f^4 - 840*a^4*b^6*c^14*d^4*f^4 + 5
28*a^7*b^3*c^13*d^5*f^4 + 528*a^3*b^7*c^5*d^13*f^4 - 480*a^8*b^2*c^4*d^14*f
^4 - 480*a^6*b^4*c^14*d^4*f^4 - 480*a^4*b^6*c^4*d^14*f^4 - 480*a^2*b^8*c^14
*d^4*f^4 - 368*a^8*b^2*c^12*d^6*f^4 + 368*a^7*b^3*c^3*d^15*f^4 + 368*a^3*b^
7*c^15*d^3*f^4 - 368*a^2*b^8*c^6*d^12*f^4 + 304*a^5*b^5*c^15*d^3*f^4 + 304*
a^5*b^5*c^3*d^15*f^4 - 144*a^6*b^4*c^2*d^16*f^4 - 144*a^4*b^6*c^16*d^2*f^4
- 108*a^8*b^2*c^2*d^16*f^4 - 108*a^2*b^8*c^16*d^2*f^4 + 80*a^7*b^3*c^15*d^3
*f^4 + 80*a^3*b^7*c^3*d^15*f^4 - 60*a^8*b^2*c^14*d^4*f^4 - 60*a^6*b^4*c^16*
d^2*f^4 - 60*a^4*b^6*c^2*d^16*f^4 - 60*a^2*b^8*c^4*d^14*f^4 - 80*b^10*c^12*
d^6*f^4 - 60*b^10*c^14*d^4*f^4 - 60*b^10*c^10*d^8*f^4 - 24*b^10*c^16*d^2*f^
4 - 24*b^10*c^8*d^10*f^4 - 4*b^10*c^6*d^12*f^4 - 80*a^10*c^6*d^12*f^4 - 60*
a^10*c^8*d^10*f^4 - 60*a^10*c^4*d^14*f^4 - 24*a^10*c^10*d^8*f^4 - 24*a^10*c
^2*d^16*f^4 - 4*a^10*c^12*d^6*f^4 - 8*a^8*b^2*d^18*f^4 - 4*a^6*b^4*d^18*f^4
- 8*a^2*b^8*c^18*f^4 - 4*a^4*b^6*c^18*f^4 - 4*b^10*c^18*f^4 - 4*a^10*d^18*
f^4 - 12*A*C*a^7*b*c*d^11*f^2 - 12*A*C*a*b^7*c^11*d*f^2 - 912*B*C*a^4*b^4*c
^5*d^7*f^2 + 792*B*C*a^5*b^3*c^4*d^8*f^2 - 792*B*C*a^3*b^5*c^8*d^4*f^2 + 72
0*B*C*a^4*b^4*c^7*d^5*f^2 - 480*B*C*a^6*b^2*c^5*d^7*f^2 - 408*B*C*a^2*b^6*c
^5*d^7*f^2 + 384*B*C*a^2*b^6*c^7*d^5*f^2 - 336*B*C*a^5*b^3*c^8*d^4*f^2 + 32
4*B*C*a^3*b^5*c^4*d^8*f^2 + 312*B*C*a^6*b^2*c^7*d^5*f^2 - 248*B*C*a^6*b^2*c
^3*d^9*f^2 + 216*B*C*a^2*b^6*c^9*d^3*f^2 - 196*B*C*a^4*b^4*c^3*d^9*f^2 + 13
2*B*C*a^4*b^4*c^9*d^3*f^2 + 80*B*C*a^3*b^5*c^6*d^6*f^2 - 64*B*C*a^5*b^3*c^6
*d^6*f^2 - 36*B*C*a^3*b^5*c^2*d^10*f^2 - 28*B*C*a^2*b^6*c^3*d^9*f^2 + 12*B*
C*a^5*b^3*c^10*d^2*f^2 - 12*B*C*a^5*b^3*c^2*d^10*f^2 - 12*B*C*a^3*b^5*c^10*
d^2*f^2 - 4*B*C*a^6*b^2*c^9*d^3*f^2 - 1468*A*C*a^4*b^4*c^6*d^6*f^2 + 996*A*
C*a^3*b^5*c^7*d^5*f^2 + 900*A*C*a^5*b^3*c^5*d^7*f^2 - 676*A*C*a^6*b^2*c^6*d
^6*f^2 - 660*A*C*a^2*b^6*c^6*d^6*f^2 + 636*A*C*a^3*b^5*c^5*d^7*f^2 + 540*A*
C*a^5*b^3*c^7*d^5*f^2 - 236*A*C*a^5*b^3*c^3*d^9*f^2 - 204*A*C*a^3*b^5*c^9*d
^3*f^2 + 156*A*C*a^2*b^6*c^10*d^2*f^2 + 132*A*C*a^6*b^2*c^2*d^10*f^2 - 72*A
*C*a^6*b^2*c^4*d^8*f^2 - 72*A*C*a^5*b^3*c^9*d^3*f^2 + 66*A*C*a^2*b^6*c^4*d^
8*f^2 + 54*A*C*a^4*b^4*c^10*d^2*f^2 + 54*A*C*a^4*b^4*c^2*d^10*f^2 - 48*A*C*
a^4*b^4*c^4*d^8*f^2 - 48*A*C*a^2*b^6*c^8*d^4*f^2 + 42*A*C*a^6*b^2*c^8*d^4*f

^2 - 40*A*C*a^3*b^5*c^3*d^9*f^2 - 36*A*C*a^4*b^4*c^8*d^4*f^2 + 24*A*C*a^2*b
^6*c^2*d^10*f^2 + 960*A*B*a^4*b^4*c^5*d^7*f^2 - 864*A*B*a^5*b^3*c^4*d^8*f^2
 + 756*A*B*a^3*b^5*c^8*d^4*f^2 - 744*A*B*a^4*b^4*c^7*d^5*f^2 - 528*A*B*a^3*
b^5*c^4*d^8*f^2 + 504*A*B*a^6*b^2*c^5*d^7*f^2 - 432*A*B*a^2*b^6*c^7*d^5*f^2
 + 432*A*B*a^2*b^6*c^5*d^7*f^2 + 348*A*B*a^5*b^3*c^8*d^4*f^2 - 312*A*B*a^6*
b^2*c^7*d^5*f^2 - 284*A*B*a^2*b^6*c^9*d^3*f^2 + 280*A*B*a^6*b^2*c^3*d^9*f^2
 + 264*A*B*a^4*b^4*c^3*d^9*f^2 - 240*A*B*a^3*b^5*c^6*d^6*f^2 - 172*A*B*a^4*
b^4*c^9*d^3*f^2 + 68*A*B*a^2*b^6*c^3*d^9*f^2 - 60*A*B*a^3*b^5*c^2*d^10*f^2
+ 24*A*B*a^5*b^3*c^6*d^6*f^2 - 24*A*B*a^5*b^3*c^2*d^10*f^2 + 12*A*B*a^3*b^5
*c^10*d^2*f^2 + 360*B*C*a^7*b*c^4*d^8*f^2 - 336*B*C*a*b^7*c^8*d^4*f^2 + 168
*B*C*a*b^7*c^6*d^6*f^2 - 136*B*C*a^7*b*c^6*d^6*f^2 + 36*B*C*a^6*b^2*c*d^11*
f^2 - 36*B*C*a^2*b^6*c^11*d*f^2 - 24*B*C*a^7*b*c^2*d^10*f^2 + 24*B*C*a*b^7*
c^10*d^2*f^2 - 12*B*C*a^4*b^4*c^11*d*f^2 + 12*B*C*a^4*b^4*c*d^11*f^2 + 12*B
*C*a*b^7*c^4*d^8*f^2 + 444*A*C*a*b^7*c^7*d^5*f^2 + 348*A*C*a^7*b*c^5*d^7*f^
2 - 164*A*C*a^7*b*c^3*d^9*f^2 - 132*A*C*a*b^7*c^9*d^3*f^2 + 84*A*C*a*b^7*c^
5*d^7*f^2 + 32*A*C*a*b^7*c^3*d^9*f^2 - 12*A*C*a^7*b*c^7*d^5*f^2 - 12*A*C*a^
5*b^3*c*d^11*f^2 - 12*A*C*a^3*b^5*c^11*d*f^2 - 360*A*B*a^7*b*c^4*d^8*f^2 +
288*A*B*a*b^7*c^8*d^4*f^2 - 288*A*B*a*b^7*c^6*d^6*f^2 - 144*A*B*a*b^7*c^4*d
^8*f^2 + 136*A*B*a^7*b*c^6*d^6*f^2 - 60*A*B*a*b^7*c^2*d^10*f^2 - 36*A*B*a*b
^7*c^10*d^2*f^2 + 24*A*B*a^7*b*c^2*d^10*f^2 - 24*A*B*a^6*b^2*c*d^11*f^2 + 1
2*A*B*a^4*b^4*c*d^11*f^2 + 12*A*B*a^2*b^6*c^11*d*f^2 + 12*A*B*a^2*b^6*c*d^1
1*f^2 + 80*B*C*b^8*c^9*d^3*f^2 - 24*B*C*b^8*c^7*d^5*f^2 - 90*A*C*b^8*c^8*d^
4*f^2 - 80*B*C*a^8*c^3*d^9*f^2 + 54*A*C*b^8*c^10*d^2*f^2 - 30*A*C*b^8*c^6*d
^6*f^2 + 24*B*C*a^8*c^5*d^7*f^2 - 12*A*C*b^8*c^4*d^8*f^2 - 112*A*B*b^8*c^9*
d^3*f^2 - 66*A*C*a^8*c^4*d^8*f^2 + 54*A*C*a^8*c^2*d^10*f^2 - 8*B*C*a^5*b^3*
d^12*f^2 - 8*B*C*a^3*b^5*d^12*f^2 + 4*A*B*b^8*c^3*d^9*f^2 + 2*A*C*a^8*c^6*d
^6*f^2 + 80*A*B*a^8*c^3*d^9*f^2 - 24*A*B*a^8*c^5*d^7*f^2 + 8*A*C*a^2*b^6*d^
12*f^2 - 4*B*C*a^3*b^5*c^12*f^2 + 4*A*C*a^4*b^4*d^12*f^2 - 2*A*C*a^6*b^2*d^
12*f^2 + 6*A*C*a^2*b^6*c^12*f^2 + 4*A*B*a^5*b^3*d^12*f^2 - 4*A*B*a^3*b^5*d^
12*f^2 + 726*C^2*a^4*b^4*c^6*d^6*f^2 - 402*C^2*a^5*b^3*c^5*d^7*f^2 - 402*C^
2*a^3*b^5*c^7*d^5*f^2 + 322*C^2*a^6*b^2*c^6*d^6*f^2 + 322*C^2*a^2*b^6*c^6*d
^6*f^2 - 222*C^2*a^5*b^3*c^7*d^5*f^2 - 222*C^2*a^3*b^5*c^5*d^7*f^2 + 134*C^
2*a^5*b^3*c^3*d^9*f^2 + 134*C^2*a^3*b^5*c^9*d^3*f^2 - 66*C^2*a^6*b^2*c^2*d^
10*f^2 - 66*C^2*a^2*b^6*c^10*d^2*f^2 + 52*C^2*a^5*b^3*c^9*d^3*f^2 + 52*C^2*
a^3*b^5*c^3*d^9*f^2 - 27*C^2*a^6*b^2*c^8*d^4*f^2 - 27*C^2*a^2*b^6*c^4*d^8*f
^2 + 24*C^2*a^6*b^2*c^4*d^8*f^2 + 24*C^2*a^4*b^4*c^8*d^4*f^2 + 24*C^2*a^4*b
^4*c^4*d^8*f^2 + 24*C^2*a^2*b^6*c^8*d^4*f^2 - 15*C^2*a^4*b^4*c^10*d^2*f^2 -
 15*C^2*a^4*b^4*c^2*d^10*f^2 - 570*B^2*a^4*b^4*c^6*d^6*f^2 + 366*B^2*a^3*b^
5*c^7*d^5*f^2 + 318*B^2*a^5*b^3*c^5*d^7*f^2 - 262*B^2*a^6*b^2*c^6*d^6*f^2 -
 222*B^2*a^2*b^6*c^6*d^6*f^2 - 210*B^2*a^5*b^3*c^3*d^9*f^2 + 186*B^2*a^5*b^
3*c^7*d^5*f^2 + 162*B^2*a^3*b^5*c^5*d^7*f^2 - 142*B^2*a^3*b^5*c^9*d^3*f^2 +
 132*B^2*a^4*b^4*c^4*d^8*f^2 + 117*B^2*a^2*b^6*c^4*d^8*f^2 + 102*B^2*a^6*b^
2*c^2*d^10*f^2 - 96*B^2*a^3*b^5*c^3*d^9*f^2 + 90*B^2*a^2*b^6*c^10*d^2*f^2 +
 81*B^2*a^4*b^4*c^2*d^10*f^2 - 56*B^2*a^5*b^3*c^9*d^3*f^2 + 48*B^2*a^6*b^2*
c^4*d^8*f^2 + 48*B^2*a^4*b^4*c^8*d^4*f^2 + 45*B^2*a^6*b^2*c^8*d^4*f^2 + 36*

B^2*a^2*b^6*c^8*d^4*f^2 + 36*B^2*a^2*b^6*c^2*d^10*f^2 + 33*B^2*a^4*b^4*c^10
*d^2*f^2 + 822*A^2*a^4*b^4*c^6*d^6*f^2 - 594*A^2*a^3*b^5*c^7*d^5*f^2 - 498*
A^2*a^5*b^3*c^5*d^7*f^2 + 498*A^2*a^2*b^6*c^6*d^6*f^2 - 414*A^2*a^3*b^5*c^5
*d^7*f^2 + 354*A^2*a^6*b^2*c^6*d^6*f^2 - 318*A^2*a^5*b^3*c^7*d^5*f^2 + 144*
A^2*a^2*b^6*c^8*d^4*f^2 + 102*A^2*a^5*b^3*c^3*d^9*f^2 + 84*A^2*a^4*b^4*c^4*
d^8*f^2 + 81*A^2*a^2*b^6*c^4*d^8*f^2 + 72*A^2*a^4*b^4*c^8*d^4*f^2 + 70*A^2*
a^3*b^5*c^9*d^3*f^2 - 66*A^2*a^6*b^2*c^2*d^10*f^2 + 48*A^2*a^6*b^2*c^4*d^8*
f^2 - 42*A^2*a^2*b^6*c^10*d^2*f^2 + 24*A^2*a^2*b^6*c^2*d^10*f^2 + 20*A^2*a^
5*b^3*c^9*d^3*f^2 - 15*A^2*a^6*b^2*c^8*d^4*f^2 - 15*A^2*a^4*b^4*c^10*d^2*f^
2 - 15*A^2*a^4*b^4*c^2*d^10*f^2 - 12*A^2*a^3*b^5*c^3*d^9*f^2 - 24*B*C*b^8*c
^11*d*f^2 + 24*B*C*a^8*c*d^11*f^2 + 12*A*B*b^8*c^11*d*f^2 - 8*B*C*a^7*b*d^1
2*f^2 - 24*A*B*a^8*c*d^11*f^2 + 4*B*C*a*b^7*c^12*f^2 + 8*A*B*a^7*b*d^12*f^2
 - 8*A*B*a*b^7*d^12*f^2 - 8*A*B*a*b^7*c^12*f^2 - 174*C^2*a^7*b*c^5*d^7*f^2
- 174*C^2*a*b^7*c^7*d^5*f^2 + 82*C^2*a^7*b*c^3*d^9*f^2 + 82*C^2*a*b^7*c^9*d
^3*f^2 + 6*C^2*a^7*b*c^7*d^5*f^2 + 6*C^2*a^5*b^3*c*d^11*f^2 + 6*C^2*a^3*b^5
*c^11*d*f^2 + 6*C^2*a*b^7*c^5*d^7*f^2 + 162*B^2*a*b^7*c^7*d^5*f^2 + 138*B^2
*a^7*b*c^5*d^7*f^2 - 118*B^2*a^7*b*c^3*d^9*f^2 - 86*B^2*a*b^7*c^9*d^3*f^2 -
 30*B^2*a^5*b^3*c*d^11*f^2 - 18*B^2*a^7*b*c^7*d^5*f^2 - 18*B^2*a*b^7*c^5*d^
7*f^2 - 12*B^2*a^3*b^5*c*d^11*f^2 - 6*B^2*a^3*b^5*c^11*d*f^2 - 4*B^2*a*b^7*
c^3*d^9*f^2 - 270*A^2*a*b^7*c^7*d^5*f^2 - 174*A^2*a^7*b*c^5*d^7*f^2 - 90*A^
2*a*b^7*c^5*d^7*f^2 + 82*A^2*a^7*b*c^3*d^9*f^2 + 50*A^2*a*b^7*c^9*d^3*f^2 -
 32*A^2*a*b^7*c^3*d^9*f^2 + 6*A^2*a^7*b*c^7*d^5*f^2 + 6*A^2*a^5*b^3*c*d^11*
f^2 + 6*A^2*a^3*b^5*c^11*d*f^2 + 6*C^2*a^7*b*c*d^11*f^2 + 6*C^2*a*b^7*c^11*
d*f^2 - 18*B^2*a^7*b*c*d^11*f^2 - 6*B^2*a*b^7*c^11*d*f^2 + 6*A^2*a^7*b*c*d^
11*f^2 + 6*A^2*a*b^7*c^11*d*f^2 - 6*A*C*a^8*d^12*f^2 - 2*A*C*b^8*c^12*f^2 +
 33*C^2*b^8*c^8*d^4*f^2 - 27*C^2*b^8*c^10*d^2*f^2 - C^2*b^8*c^6*d^6*f^2 + 3
3*C^2*a^8*c^4*d^8*f^2 + 33*B^2*b^8*c^10*d^2*f^2 - 27*C^2*a^8*c^2*d^10*f^2 -
 27*B^2*b^8*c^8*d^4*f^2 + 3*B^2*b^8*c^6*d^6*f^2 - C^2*a^8*c^6*d^6*f^2 + 117
*A^2*b^8*c^8*d^4*f^2 + 111*A^2*b^8*c^6*d^6*f^2 + 72*A^2*b^8*c^4*d^8*f^2 + 3
3*B^2*a^8*c^2*d^10*f^2 - 27*B^2*a^8*c^4*d^8*f^2 + 24*A^2*b^8*c^2*d^10*f^2 +
 4*C^2*a^4*b^4*d^12*f^2 + 3*C^2*a^6*b^2*d^12*f^2 + 3*B^2*a^8*c^6*d^6*f^2 -
3*A^2*b^8*c^10*d^2*f^2 + 33*A^2*a^8*c^4*d^8*f^2 - 27*A^2*a^8*c^2*d^10*f^2 +
 4*C^2*a^4*b^4*c^12*f^2 + 4*B^2*a^4*b^4*d^12*f^2 + 4*B^2*a^2*b^6*d^12*f^2 +
 3*C^2*a^2*b^6*c^12*f^2 + 3*B^2*a^6*b^2*d^12*f^2 - A^2*a^8*c^6*d^6*f^2 - 4*
A^2*a^4*b^4*d^12*f^2 + 3*B^2*a^2*b^6*c^12*f^2 - A^2*a^6*b^2*d^12*f^2 - A^2*
a^2*b^6*c^12*f^2 + 3*C^2*b^8*c^12*f^2 + 3*C^2*a^8*d^12*f^2 + 4*A^2*b^8*d^12
*f^2 - B^2*b^8*c^12*f^2 - B^2*a^8*d^12*f^2 + 3*A^2*b^8*c^12*f^2 + 3*A^2*a^8
*d^12*f^2 - 24*A*B*C*a*b^6*c*d^8*f + 342*A*B*C*a^2*b^5*c^4*d^5*f - 186*A*B*
C*a^3*b^4*c^5*d^4*f - 66*A*B*C*a^4*b^3*c^2*d^7*f + 48*A*B*C*a^2*b^5*c^2*d^7
*f + 42*A*B*C*a^2*b^5*c^6*d^3*f + 26*A*B*C*a^5*b^2*c^3*d^6*f + 24*A*B*C*a^4
*b^3*c^6*d^3*f - 18*A*B*C*a^4*b^3*c^4*d^5*f - 18*A*B*C*a^3*b^4*c^7*d^2*f -
8*A*B*C*a^3*b^4*c^3*d^6*f + 6*A*B*C*a^5*b^2*c^5*d^4*f - 128*A*B*C*a*b^6*c^3
*d^6*f + 126*A*B*C*a*b^6*c^7*d^2*f + 72*A*B*C*a^3*b^4*c*d^8*f - 36*A*B*C*a^
5*b^2*c*d^8*f - 36*A*B*C*a^2*b^5*c^8*d*f + 30*A*B*C*a^6*b*c^2*d^7*f - 12*A*
B*C*a^6*b*c^4*d^5*f - 12*A*B*C*a*b^6*c^5*d^4*f - 21*B^2*C*a*b^6*c^8*d*f - 3

*B^2*C*a^6*b*c*d^8*f + 21*A^2*C*a*b^6*c^8*d*f - 21*A*C^2*a*b^6*c^8*d*f - 9*
A^2*C*a^6*b*c*d^8*f + 9*A*C^2*a^6*b*c*d^8*f + 36*A^2*B*a*b^6*c*d^8*f + 21*A
*B^2*a*b^6*c^8*d*f + 3*A*B^2*a^6*b*c*d^8*f - 78*A*B*C*b^7*c^6*d^3*f + 24*A*
B*C*b^7*c^4*d^5*f + 2*A*B*C*a^7*c^3*d^6*f + 16*A*B*C*a^4*b^3*d^9*f - 16*A*B
*C*a^2*b^5*d^9*f - 237*B^2*C*a^3*b^4*c^4*d^5*f + 165*B*C^2*a^3*b^4*c^5*d^4*
f + 92*B^2*C*a^2*b^5*c^3*d^6*f - 81*B^2*C*a^2*b^5*c^7*d^2*f + 77*B^2*C*a^4*
b^3*c^3*d^6*f - 75*B*C^2*a^2*b^5*c^4*d^5*f + 69*B^2*C*a^4*b^3*c^5*d^4*f + 6
9*B*C^2*a^4*b^3*c^4*d^5*f - 68*B*C^2*a^3*b^4*c^3*d^6*f - 63*B^2*C*a^5*b^2*c
^4*d^5*f - 61*B*C^2*a^2*b^5*c^6*d^3*f + 57*B*C^2*a^4*b^3*c^2*d^7*f - 53*B*C
^2*a^5*b^2*c^3*d^6*f - 44*B*C^2*a^4*b^3*c^6*d^3*f - 36*B^2*C*a^3*b^4*c^2*d^
7*f + 35*B^2*C*a^3*b^4*c^6*d^3*f + 33*B^2*C*a^5*b^2*c^2*d^7*f - 33*B^2*C*a^
2*b^5*c^5*d^4*f + 33*B*C^2*a^3*b^4*c^7*d^2*f - 12*B^2*C*a^4*b^3*c^7*d^2*f +
 9*B*C^2*a^5*b^2*c^5*d^4*f + 4*B^2*C*a^5*b^2*c^6*d^3*f + 225*A^2*C*a^2*b^5*
c^5*d^4*f - 105*A*C^2*a^2*b^5*c^5*d^4*f - 99*A^2*C*a^3*b^4*c^4*d^5*f - 81*A
^2*C*a^5*b^2*c^4*d^5*f + 67*A^2*C*a^4*b^3*c^3*d^6*f - 59*A*C^2*a^4*b^3*c^3*
d^6*f + 57*A*C^2*a^5*b^2*c^2*d^7*f - 57*A*C^2*a^2*b^5*c^7*d^2*f + 51*A^2*C*
a^4*b^3*c^5*d^4*f + 48*A^2*C*a^3*b^4*c^2*d^7*f + 45*A*C^2*a^5*b^2*c^4*d^5*f
 - 35*A^2*C*a^3*b^4*c^6*d^3*f - 33*A^2*C*a^5*b^2*c^2*d^7*f + 33*A^2*C*a^2*b
^5*c^7*d^2*f + 33*A*C^2*a^4*b^3*c^5*d^4*f + 27*A*C^2*a^3*b^4*c^6*d^3*f - 24
*A*C^2*a^3*b^4*c^2*d^7*f + 24*A*C^2*a^2*b^5*c^3*d^6*f - 21*A*C^2*a^3*b^4*c^
4*d^5*f - 16*A^2*C*a^2*b^5*c^3*d^6*f - 243*A^2*B*a^2*b^5*c^4*d^5*f - 156*A*
B^2*a^2*b^5*c^3*d^6*f + 141*A*B^2*a^3*b^4*c^4*d^5*f + 108*A^2*B*a^3*b^4*c^3
*d^6*f - 105*A*B^2*a^4*b^3*c^3*d^6*f + 84*A*B^2*a^3*b^4*c^2*d^7*f + 81*A*B^
2*a^2*b^5*c^5*d^4*f - 51*A^2*B*a^4*b^3*c^4*d^5*f + 51*A^2*B*a^2*b^5*c^6*d^3
*f - 48*A^2*B*a^2*b^5*c^2*d^7*f + 45*A^2*B*a^3*b^4*c^5*d^4*f + 39*A*B^2*a^5
*b^2*c^4*d^5*f - 35*A*B^2*a^3*b^4*c^6*d^3*f + 33*A*B^2*a^2*b^5*c^7*d^2*f +
27*A^2*B*a^5*b^2*c^3*d^6*f - 21*A*B^2*a^4*b^3*c^5*d^4*f + 20*A^2*B*a^4*b^3*
c^6*d^3*f - 15*A^2*B*a^5*b^2*c^5*d^4*f - 15*A^2*B*a^3*b^4*c^7*d^2*f + 9*A^2
*B*a^4*b^3*c^2*d^7*f + 3*A*B^2*a^5*b^2*c^2*d^7*f + 18*A*B*C*b^7*c^8*d*f - 6
*A*B*C*a^7*c*d^8*f + 2*A*B*C*a^6*b*d^9*f - 6*A*B*C*a*b^6*c^9*f + 63*B^2*C*a
*b^6*c^6*d^3*f - 48*B^2*C*a^4*b^3*c*d^8*f + 42*B*C^2*a^2*b^5*c^8*d*f + 42*B
*C^2*a*b^6*c^5*d^4*f - 39*B*C^2*a*b^6*c^7*d^2*f + 30*B*C^2*a^5*b^2*c*d^8*f
- 24*B^2*C*a*b^6*c^4*d^5*f - 24*B*C^2*a^3*b^4*c*d^8*f + 17*B^2*C*a^6*b*c^3*
d^6*f - 15*B*C^2*a^6*b*c^2*d^7*f + 12*B^2*C*a^3*b^4*c^8*d*f + 12*B^2*C*a^2*
b^5*c*d^8*f + 6*B*C^2*a^6*b*c^4*d^5*f - 192*A^2*C*a*b^6*c^4*d^5*f - 99*A^2*
C*a*b^6*c^6*d^3*f + 84*A*C^2*a*b^6*c^4*d^5*f + 59*A*C^2*a*b^6*c^6*d^3*f + 5
1*A^2*C*a^6*b*c^3*d^6*f - 51*A*C^2*a^6*b*c^3*d^6*f - 36*A^2*C*a^2*b^5*c*d^8
*f - 24*A*C^2*a^4*b^3*c*d^8*f + 24*A*C^2*a^2*b^5*c*d^8*f + 12*A^2*C*a^4*b^3
*c*d^8*f + 12*A*C^2*a^3*b^4*c^8*d*f + 160*A^2*B*a*b^6*c^3*d^6*f - 99*A*B^2*
a*b^6*c^6*d^3*f - 87*A^2*B*a*b^6*c^7*d^2*f - 72*A*B^2*a*b^6*c^4*d^5*f - 48*
A*B^2*a^2*b^5*c*d^8*f - 36*A^2*B*a^3*b^4*c*d^8*f + 24*A*B^2*a^4*b^3*c*d^8*f
 - 17*A*B^2*a^6*b*c^3*d^6*f - 15*A^2*B*a^6*b*c^2*d^7*f + 12*A*B^2*a*b^6*c^2
*d^7*f + 6*A^2*B*a^6*b*c^4*d^5*f + 6*A^2*B*a^5*b^2*c*d^8*f + 6*A^2*B*a^2*b^
5*c^8*d*f - 6*A^2*B*a*b^6*c^5*d^4*f + 3*B^2*C*b^7*c^7*d^2*f - B*C^2*b^7*c^6
*d^3*f + 96*A^2*C*b^7*c^5*d^4*f - 39*A^2*C*b^7*c^7*d^2*f - 36*A*C^2*b^7*c^5

*d^4*f + 32*A^2*C*b^7*c^3*d^6*f + 15*A*C^2*b^7*c^7*d^2*f - 3*B^2*C*a^7*c^2*
d^7*f - B*C^2*a^7*c^3*d^6*f + 111*A^2*B*b^7*c^6*d^3*f - 39*A*B^2*b^7*c^7*d^
2*f + 24*A*B^2*b^7*c^5*d^4*f + 12*B^2*C*a^3*b^4*d^9*f - 12*B*C^2*a^4*b^3*d^
9*f - 9*A^2*C*a^7*c^2*d^7*f + 9*A*C^2*a^7*c^2*d^7*f - 4*A*B^2*b^7*c^3*d^6*f
 - 12*A^2*C*a^3*b^4*d^9*f - 8*A*C^2*a^5*b^2*d^9*f + 8*A*C^2*a^3*b^4*d^9*f +
 4*B^2*C*a^2*b^5*c^9*f + 4*A^2*C*a^5*b^2*d^9*f - 4*B*C^2*a^3*b^4*c^9*f + 3*
A*B^2*a^7*c^2*d^7*f - A^2*B*a^7*c^3*d^6*f + 12*A^2*B*a^2*b^5*d^9*f - 8*A*B^
2*a^3*b^4*d^9*f - 4*A^2*B*a^4*b^3*d^9*f + 4*A*C^2*a^2*b^5*c^9*f - 3*C^3*a^6
*b*c*d^8*f + 3*C^3*a*b^6*c^8*d*f + 3*A^3*a^6*b*c*d^8*f - 3*A^3*a*b^6*c^8*d*
f + 3*B*C^2*b^7*c^8*d*f + 12*A^2*C*b^7*c*d^8*f + 3*B*C^2*a^7*c*d^8*f - 9*A^
2*B*b^7*c^8*d*f - B*C^2*a^6*b*d^9*f + 4*A^2*C*a*b^6*d^9*f + 3*A^2*B*a^7*c*d
^8*f + 3*B*C^2*a*b^6*c^9*f + 8*A*B^2*a*b^6*d^9*f - A^2*B*a^6*b*d^9*f - A^2*
B*a*b^6*c^9*f - 39*C^3*a^4*b^3*c^5*d^4*f + 39*C^3*a^3*b^4*c^4*d^5*f - 27*C^
3*a^5*b^2*c^2*d^7*f + 27*C^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^4*b^3*c^3*d^6*f -
 17*C^3*a^3*b^4*c^6*d^3*f - 3*C^3*a^5*b^2*c^4*d^5*f + 3*C^3*a^2*b^5*c^5*d^4
*f - 63*B^3*a^3*b^4*c^5*d^4*f + 57*B^3*a^2*b^5*c^4*d^5*f - 51*B^3*a^4*b^3*c
^2*d^7*f + 48*B^3*a^3*b^4*c^3*d^6*f + 31*B^3*a^2*b^5*c^6*d^3*f + 27*B^3*a^5
*b^2*c^3*d^6*f + 16*B^3*a^4*b^3*c^6*d^3*f - 15*B^3*a^5*b^2*c^5*d^4*f - 12*B
^3*a^2*b^5*c^2*d^7*f + 9*B^3*a^4*b^3*c^4*d^5*f - 3*B^3*a^3*b^4*c^7*d^2*f -
123*A^3*a^2*b^5*c^5*d^4*f + 81*A^3*a^3*b^4*c^4*d^5*f - 45*A^3*a^4*b^3*c^5*d
^4*f + 39*A^3*a^5*b^2*c^4*d^5*f - 25*A^3*a^4*b^3*c^3*d^6*f + 25*A^3*a^3*b^4
*c^6*d^3*f - 24*A^3*a^3*b^4*c^2*d^7*f - 8*A^3*a^2*b^5*c^3*d^6*f + 3*A^3*a^5
*b^2*c^2*d^7*f - 3*A^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^6*b*c^3*d^6*f - 17*C^3*
a*b^6*c^6*d^3*f + 12*C^3*a^4*b^3*c*d^8*f - 12*C^3*a^3*b^4*c^8*d*f + 24*B^3*
a^3*b^4*c*d^8*f + 21*B^3*a*b^6*c^7*d^2*f - 18*B^3*a*b^6*c^5*d^4*f - 15*B^3*
a^6*b*c^2*d^7*f + 6*B^3*a^6*b*c^4*d^5*f + 6*B^3*a^5*b^2*c*d^8*f - 6*B^3*a^2
*b^5*c^8*d*f + 4*B^3*a*b^6*c^3*d^6*f + 108*A^3*a*b^6*c^4*d^5*f + 57*A^3*a*b
^6*c^6*d^3*f - 17*A^3*a^6*b*c^3*d^6*f + 12*A^3*a^2*b^5*c*d^8*f + 3*C^3*b^7*
c^7*d^2*f - 3*C^3*a^7*c^2*d^7*f - B^3*b^7*c^6*d^3*f - 60*A^3*b^7*c^5*d^4*f
- 32*A^3*b^7*c^3*d^6*f + 21*A^3*b^7*c^7*d^2*f + 4*C^3*a^5*b^2*d^9*f - B^3*a
^7*c^3*d^6*f - 4*C^3*a^2*b^5*c^9*f - 4*B^3*a^2*b^5*d^9*f + 3*A^3*a^7*c^2*d^
7*f + 4*A^3*a^3*b^4*d^9*f + 3*B^3*b^7*c^8*d*f - 12*A^3*b^7*c*d^8*f + 3*B^3*
a^7*c*d^8*f - B^3*a^6*b*d^9*f - 4*A^3*a*b^6*d^9*f - B^3*a*b^6*c^9*f - B^2*C
*b^7*c^9*f - 4*A^2*B*b^7*d^9*f + 3*A^2*C*a^7*d^9*f - 3*A*C^2*a^7*d^9*f - A*
C^2*b^7*c^9*f - A*B^2*a^7*d^9*f - C^3*b^7*c^9*f - A^3*a^7*d^9*f + B^2*C*a^7
*d^9*f + A^2*C*b^7*c^9*f + A*B^2*b^7*c^9*f + C^3*a^7*d^9*f + A^3*b^7*c^9*f
- 6*A*B^2*C*a*b^5*c^5*d - 21*A^2*B*C*a^2*b^4*c^3*d^3 + 21*A*B*C^2*a^2*b^4*c
^3*d^3 + 12*A*B^2*C*a^2*b^4*c^4*d^2 - 12*A*B^2*C*a^2*b^4*c^2*d^4 - 10*A*B^2
*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^3*b^3*c^4*d^2
+ 3*A^2*B*C*a^3*b^3*c^2*d^4 + 3*A*B^2*C*a^4*b^2*c^2*d^4 + 3*A*B*C^2*a^3*b^3
*c^2*d^4 + 2*A*B*C^2*a^4*b^2*c^3*d^3 - A^2*B*C*a^4*b^2*c^3*d^3 + 18*A^2*B*C
*a*b^5*c^2*d^4 + 10*A*B^2*C*a*b^5*c^3*d^3 + 9*A^2*B*C*a*b^5*c^4*d^2 - 9*A*B
*C^2*a*b^5*c^4*d^2 - 9*A*B*C^2*a*b^5*c^2*d^4 - 6*A^2*B*C*a^2*b^4*c*d^5 + 6*
A*B^2*C*a^3*b^3*c*d^5 - 6*A*B*C^2*a^4*b^2*c*d^5 + 6*A*B*C^2*a^2*b^4*c^5*d +
 3*A^2*B*C*a^4*b^2*c*d^5 - 3*A^2*B*C*a^2*b^4*c^5*d + 3*A*B*C^2*a^2*b^4*c*d^

5 + 3*B^3*C*a^4*b^2*c*d^5 - 3*B^3*C*a^2*b^4*c^5*d + 3*B^3*C*a*b^5*c^4*d^2 + 3*B^2*C^2*a*b^5*c^5*d + 3*B*C^3*a^4*b^2*c*d^5 - 3*B*C^3*a^2*b^4*c^5*d + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a*b^5*c^3*d^3 + 8*A*C^3*a*b^5*c^3*d^3 - 9*A^3*B*a*b^5*c^2*d^4 - 9*A*B^3*a*b^5*c^2*d^4 + 3*A^3*B*a^2*b^4*c*d^5 - 3*A^3*B*a*b^5*c^4*d^2 + 3*A^2*B^2*a*b^5*c^5*d + 3*A*B^3*a^2*b^4*c*d^5 - 3*A*B^3*a*b^5*c^4*d^2 - 3*A*B^2*C*b^6*c^4*d^2 - 2*A^2*B*C*b^6*c^3*d^3 + 5*A*B*C^2*a^3*b^3*d^6 - 4*A^2*B*C*a^3*b^3*d^6 - A*B^2*C*a^4*b^2*d^6 + 9*B^2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^2*b^4*c^4*d^2 + 6*B^2*C^2*a^2*b^4*c^2*d^4 - 3*B^2*C^2*a^4*b^2*c^2*d^4 + 24*A^2*C^2*a^3*b^3*c^3*d^3 - 15*A^2*C^2*a^2*b^4*c^4*d^2 - 9*A^2*C^2*a^4*b^2*c^2*d^4 + 3*A^2*C^2*a^2*b^4*c^2*d^4 + 9*A^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 6*A^2*B*C*b^6*c^5*d - 3*A*B*C^2*b^6*c^5*d + 4*A^2*B*C*a*b^5*d^6 - 2*A*B*C^2*a*b^5*d^6 + 2*A*B*C^2*a*b^5*c^6 - A^2*B*C*a*b^5*c^6 - 7*B^3*C*a^2*b^4*c^3*d^3 - 7*B*C^3*a^2*b^4*c^3*d^3 + 3*B^3*C*a^3*b^3*c^4*d^2 - 3*B^3*C*a^3*b^3*c^2*d^4 - 3*B^2*C^2*a^3*b^3*c*d^5 + 3*B*C^3*a^3*b^3*c^4*d^2 - 3*B*C^3*a^3*b^3*c^2*d^4 - B^3*C*a^4*b^2*c^3*d^3 - B^2*C^2*a*b^5*c^3*d^3 - B*C^3*a^4*b^2*c^3*d^3 - 24*A^2*C^2*a*b^5*c^3*d^3 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^2*b^4*c^4*d^2 + 9*A*C^3*a^4*b^2*c^2*d^4 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^2*b^4*c^4*d^2 - 6*A^3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^4*b^2*c^2*d^4 - 9*A^2*B^2*a*b^5*c^3*d^3 + 7*A^3*B*a^2*b^4*c^3*d^3 + 7*A*B^3*a^2*b^4*c^3*d^3 - 3*A^3*B*a^3*b^3*c^2*d^4 - 3*A^2*B^2*a^3*b^3*c*d^5 - 3*A*B^3*a^3*b^3*c^2*d^4 + 12*A^2*C^2*b^6*c^4*d^2 + 3*A^2*C^2*b^6*c^2*d^4 + 6*A^2*B^2*b^6*c^4*d^2 + 3*A^2*B^2*b^6*c^2*d^4 - 5*A^2*C^2*a^2*b^4*d^6 + 3*A^2*C^2*a^4*b^2*d^6 + A*B*C^2*b^6*c^3*d^3 - 3*B^4*a^3*b^3*c*d^5 - B^4*a*b^5*c^3*d^3 + A^2*B^2*a^3*b^3*c^3*d^3 - 8*A^4*a*b^5*c^3*d^3 - 15*A^3*C*b^6*c^4*d^2 - 6*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 - 2*B^3*C*a^3*b^3*d^6 - 2*B*C^3*a^3*b^3*d^6 + 4*A^3*C*a^2*b^4*d^6 - 3*A*C^3*a^4*b^2*d^6 + 2*A*C^3*a^2*b^4*d^6 - A^3*C*a^4*b^2*d^6 - 2*A*C^3*a^2*b^4*c^6 + 3*B^4*a*b^5*c^5*d - 3*A^3*B*b^6*c^5*d - 3*A*B^3*b^6*c^5*d - B^3*C*a*b^5*c^6 - B*C^3*a*b^5*c^6 - 2*A^3*B*a*b^5*d^6 - 2*A*B^3*a*b^5*d^6 + 8*C^4*a^3*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2 + 6*B^4*a^2*b^4*c^2*d^4 - 3*B^4*a^2*b^4*c^4*d^2 + 3*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*a^4*b^2*d^6 + B^2*C^2*a^2*b^4*d^6 + B^2*C^2*a^2*b^4*c^6 + A^2*C^2*a^2*b^4*c^6 - 2*A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B*a^3*b^3*d^6 + A*B^3*a^3*b^3*d^6 + 6*A^4*b^6*c^4*d^2 + 3*A^4*b^6*c^2*d^4 - A^4*a^2*b^4*d^6 - 2*A^2*C^2*b^6*c^6 + A*B^2*C*b^6*c^6 + B^4*a^3*b^3*c^3*d^3 + A^3*C*b^6*c^6 + A*C^3*b^6*c^6 + C^4*a^4*b^2*d^6 + C^4*a^2*b^4*c^6 + B^4*a^2*b^4*d^6 + A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k)*(root(480*a^9*b*c^7*d^11*f^4 + 480*a*b^9*c^11*d^7*f^4 + 360*a^9*b*c^9*d^9*f^4 + 360*a^9*b*c^5*d^13*f^4 + 360*a*b^9*c^13*d^5*f^4 + 360*a*b^9*c^9*d^9*f^4 + 144*a^9*b*c^11*d^7*f^4 + 144*a^9*b*c^3*d^15*f^4 + 144*a*b^9*c^15*d^3*f^4 + 144*a*b^9*c^7*d^11*f^4 + 48*a^7*b^3*c*d^17*f^4 + 48*a^3*b^7*c^17*d*f^4 + 24*a^9*b*c^13*d^5*f^4 + 24*a^5*b^5*c^17*d*f^4 + 24*a^5*b^5*c*d^17*f^4 + 24*a*b^9*c^5*d^13*f^4 + 24*a^9*b*c*d^17*f^4 + 24*a*b^9*c^17*d*f^4 + 3920*a^5*b^5*c^9*d^9*f^4 - 3360*a^6*b^4*c^8*d^10*f^4 - 3360*a^4*b^6*c^10*d^8*f^4 - 3024*a^6*b^4*c^10*d^8*f^4 + 3024*a^5*b^5*c^11*d^7*f^4 + 3024*a^5*b^5*c^7*d^11*f^4 - 3024*a^4*b^6*c^8*d^10*f^4

+ 2320*a^7*b^3*c^9*d^9*f^4 + 2320*a^3*b^7*c^9*d^9*f^4 - 2240*a^6*b^4*c^6*d
^12*f^4 - 2240*a^4*b^6*c^12*d^6*f^4 + 2160*a^7*b^3*c^7*d^11*f^4 + 2160*a^3*
b^7*c^11*d^7*f^4 - 1624*a^6*b^4*c^12*d^6*f^4 - 1624*a^4*b^6*c^6*d^12*f^4 +
1488*a^7*b^3*c^11*d^7*f^4 + 1488*a^3*b^7*c^7*d^11*f^4 + 1344*a^5*b^5*c^13*d
^5*f^4 + 1344*a^5*b^5*c^5*d^13*f^4 - 1320*a^8*b^2*c^8*d^10*f^4 - 1320*a^2*b
^8*c^10*d^8*f^4 + 1200*a^7*b^3*c^5*d^13*f^4 + 1200*a^3*b^7*c^13*d^5*f^4 - 1
060*a^8*b^2*c^6*d^12*f^4 - 1060*a^2*b^8*c^12*d^6*f^4 - 948*a^8*b^2*c^10*d^8
*f^4 - 948*a^2*b^8*c^8*d^10*f^4 - 840*a^6*b^4*c^4*d^14*f^4 - 840*a^4*b^6*c^
14*d^4*f^4 + 528*a^7*b^3*c^13*d^5*f^4 + 528*a^3*b^7*c^5*d^13*f^4 - 480*a^8*
b^2*c^4*d^14*f^4 - 480*a^6*b^4*c^14*d^4*f^4 - 480*a^4*b^6*c^4*d^14*f^4 - 48
0*a^2*b^8*c^14*d^4*f^4 - 368*a^8*b^2*c^12*d^6*f^4 + 368*a^7*b^3*c^3*d^15*f^
4 + 368*a^3*b^7*c^15*d^3*f^4 - 368*a^2*b^8*c^6*d^12*f^4 + 304*a^5*b^5*c^15*
d^3*f^4 + 304*a^5*b^5*c^3*d^15*f^4 - 144*a^6*b^4*c^2*d^16*f^4 - 144*a^4*b^6
*c^16*d^2*f^4 - 108*a^8*b^2*c^2*d^16*f^4 - 108*a^2*b^8*c^16*d^2*f^4 + 80*a^
7*b^3*c^15*d^3*f^4 + 80*a^3*b^7*c^3*d^15*f^4 - 60*a^8*b^2*c^14*d^4*f^4 - 60
*a^6*b^4*c^16*d^2*f^4 - 60*a^4*b^6*c^2*d^16*f^4 - 60*a^2*b^8*c^4*d^14*f^4 -
80*b^10*c^12*d^6*f^4 - 60*b^10*c^14*d^4*f^4 - 60*b^10*c^10*d^8*f^4 - 24*b^
10*c^16*d^2*f^4 - 24*b^10*c^8*d^10*f^4 - 4*b^10*c^6*d^12*f^4 - 80*a^10*c^6*
d^12*f^4 - 60*a^10*c^8*d^10*f^4 - 60*a^10*c^4*d^14*f^4 - 24*a^10*c^10*d^8*f
^4 - 24*a^10*c^2*d^16*f^4 - 4*a^10*c^12*d^6*f^4 - 8*a^8*b^2*d^18*f^4 - 4*a^
6*b^4*d^18*f^4 - 8*a^2*b^8*c^18*f^4 - 4*a^4*b^6*c^18*f^4 - 4*b^10*c^18*f^4
- 4*a^10*d^18*f^4 - 12*A*C*a^7*b*c*d^11*f^2 - 12*A*C*a*b^7*c^11*d*f^2 - 912
*B*C*a^4*b^4*c^5*d^7*f^2 + 792*B*C*a^5*b^3*c^4*d^8*f^2 - 792*B*C*a^3*b^5*c^
8*d^4*f^2 + 720*B*C*a^4*b^4*c^7*d^5*f^2 - 480*B*C*a^6*b^2*c^5*d^7*f^2 - 408
*B*C*a^2*b^6*c^5*d^7*f^2 + 384*B*C*a^2*b^6*c^7*d^5*f^2 - 336*B*C*a^5*b^3*c^
8*d^4*f^2 + 324*B*C*a^3*b^5*c^4*d^8*f^2 + 312*B*C*a^6*b^2*c^7*d^5*f^2 - 248
*B*C*a^6*b^2*c^3*d^9*f^2 + 216*B*C*a^2*b^6*c^9*d^3*f^2 - 196*B*C*a^4*b^4*c^
3*d^9*f^2 + 132*B*C*a^4*b^4*c^9*d^3*f^2 + 80*B*C*a^3*b^5*c^6*d^6*f^2 - 64*B
*C*a^5*b^3*c^6*d^6*f^2 - 36*B*C*a^3*b^5*c^2*d^10*f^2 - 28*B*C*a^2*b^6*c^3*d
^9*f^2 + 12*B*C*a^5*b^3*c^10*d^2*f^2 - 12*B*C*a^5*b^3*c^2*d^10*f^2 - 12*B*C
*a^3*b^5*c^10*d^2*f^2 - 4*B*C*a^6*b^2*c^9*d^3*f^2 - 1468*A*C*a^4*b^4*c^6*d^
6*f^2 + 996*A*C*a^3*b^5*c^7*d^5*f^2 + 900*A*C*a^5*b^3*c^5*d^7*f^2 - 676*A*C
*a^6*b^2*c^6*d^6*f^2 - 660*A*C*a^2*b^6*c^6*d^6*f^2 + 636*A*C*a^3*b^5*c^5*d^
7*f^2 + 540*A*C*a^5*b^3*c^7*d^5*f^2 - 236*A*C*a^5*b^3*c^3*d^9*f^2 - 204*A*C
*a^3*b^5*c^9*d^3*f^2 + 156*A*C*a^2*b^6*c^10*d^2*f^2 + 132*A*C*a^6*b^2*c^2*d
^10*f^2 - 72*A*C*a^6*b^2*c^4*d^8*f^2 - 72*A*C*a^5*b^3*c^9*d^3*f^2 + 66*A*C*
a^2*b^6*c^4*d^8*f^2 + 54*A*C*a^4*b^4*c^10*d^2*f^2 + 54*A*C*a^4*b^4*c^2*d^10
*f^2 - 48*A*C*a^4*b^4*c^4*d^8*f^2 - 48*A*C*a^2*b^6*c^8*d^4*f^2 + 42*A*C*a^6
*b^2*c^8*d^4*f^2 - 40*A*C*a^3*b^5*c^3*d^9*f^2 - 36*A*C*a^4*b^4*c^8*d^4*f^2
+ 24*A*C*a^2*b^6*c^2*d^10*f^2 + 960*A*B*a^4*b^4*c^5*d^7*f^2 - 864*A*B*a^5*b
^3*c^4*d^8*f^2 + 756*A*B*a^3*b^5*c^8*d^4*f^2 - 744*A*B*a^4*b^4*c^7*d^5*f^2
- 528*A*B*a^3*b^5*c^4*d^8*f^2 + 504*A*B*a^6*b^2*c^5*d^7*f^2 - 432*A*B*a^2*b
^6*c^7*d^5*f^2 + 432*A*B*a^2*b^6*c^5*d^7*f^2 + 348*A*B*a^5*b^3*c^8*d^4*f^2
- 312*A*B*a^6*b^2*c^7*d^5*f^2 - 284*A*B*a^2*b^6*c^9*d^3*f^2 + 280*A*B*a^6*b
^2*c^3*d^9*f^2 + 264*A*B*a^4*b^4*c^3*d^9*f^2 - 240*A*B*a^3*b^5*c^6*d^6*f^2

- 172*A*B*a^4*b^4*c^9*d^3*f^2 + 68*A*B*a^2*b^6*c^3*d^9*f^2 - 60*A*B*a^3*b^5
*c^2*d^10*f^2 + 24*A*B*a^5*b^3*c^6*d^6*f^2 - 24*A*B*a^5*b^3*c^2*d^10*f^2 +
12*A*B*a^3*b^5*c^10*d^2*f^2 + 360*B*C*a^7*b*c^4*d^8*f^2 - 336*B*C*a*b^7*c^8
*d^4*f^2 + 168*B*C*a*b^7*c^6*d^6*f^2 - 136*B*C*a^7*b*c^6*d^6*f^2 + 36*B*C*a
^6*b^2*c*d^11*f^2 - 36*B*C*a^2*b^6*c^11*d*f^2 - 24*B*C*a^7*b*c^2*d^10*f^2 +
24*B*C*a*b^7*c^10*d^2*f^2 - 12*B*C*a^4*b^4*c^11*d*f^2 + 12*B*C*a^4*b^4*c*d
^11*f^2 + 12*B*C*a*b^7*c^4*d^8*f^2 + 444*A*C*a*b^7*c^7*d^5*f^2 + 348*A*C*a^
7*b*c^5*d^7*f^2 - 164*A*C*a^7*b*c^3*d^9*f^2 - 132*A*C*a*b^7*c^9*d^3*f^2 + 8
4*A*C*a*b^7*c^5*d^7*f^2 + 32*A*C*a*b^7*c^3*d^9*f^2 - 12*A*C*a^7*b*c^7*d^5*f
^2 - 12*A*C*a^5*b^3*c*d^11*f^2 - 12*A*C*a^3*b^5*c^11*d*f^2 - 360*A*B*a^7*b*
c^4*d^8*f^2 + 288*A*B*a*b^7*c^8*d^4*f^2 - 288*A*B*a*b^7*c^6*d^6*f^2 - 144*A
*B*a*b^7*c^4*d^8*f^2 + 136*A*B*a^7*b*c^6*d^6*f^2 - 60*A*B*a*b^7*c^2*d^10*f^
2 - 36*A*B*a*b^7*c^10*d^2*f^2 + 24*A*B*a^7*b*c^2*d^10*f^2 - 24*A*B*a^6*b^2*
c*d^11*f^2 + 12*A*B*a^4*b^4*c*d^11*f^2 + 12*A*B*a^2*b^6*c^11*d*f^2 + 12*A*B
*a^2*b^6*c*d^11*f^2 + 80*B*C*b^8*c^9*d^3*f^2 - 24*B*C*b^8*c^7*d^5*f^2 - 90*
A*C*b^8*c^8*d^4*f^2 - 80*B*C*a^8*c^3*d^9*f^2 + 54*A*C*b^8*c^10*d^2*f^2 - 30
*A*C*b^8*c^6*d^6*f^2 + 24*B*C*a^8*c^5*d^7*f^2 - 12*A*C*b^8*c^4*d^8*f^2 - 11
2*A*B*b^8*c^9*d^3*f^2 - 66*A*C*a^8*c^4*d^8*f^2 + 54*A*C*a^8*c^2*d^10*f^2 -
8*B*C*a^5*b^3*d^12*f^2 - 8*B*C*a^3*b^5*d^12*f^2 + 4*A*B*b^8*c^3*d^9*f^2 + 2
*A*C*a^8*c^6*d^6*f^2 + 80*A*B*a^8*c^3*d^9*f^2 - 24*A*B*a^8*c^5*d^7*f^2 + 8*
A*C*a^2*b^6*d^12*f^2 - 4*B*C*a^3*b^5*c^12*f^2 + 4*A*C*a^4*b^4*d^12*f^2 - 2*
A*C*a^6*b^2*d^12*f^2 + 6*A*C*a^2*b^6*c^12*f^2 + 4*A*B*a^5*b^3*d^12*f^2 - 4*
A*B*a^3*b^5*d^12*f^2 + 726*C^2*a^4*b^4*c^6*d^6*f^2 - 402*C^2*a^5*b^3*c^5*d^
7*f^2 - 402*C^2*a^3*b^5*c^7*d^5*f^2 + 322*C^2*a^6*b^2*c^6*d^6*f^2 + 322*C^2
*a^2*b^6*c^6*d^6*f^2 - 222*C^2*a^5*b^3*c^7*d^5*f^2 - 222*C^2*a^3*b^5*c^5*d^
7*f^2 + 134*C^2*a^5*b^3*c^3*d^9*f^2 + 134*C^2*a^3*b^5*c^9*d^3*f^2 - 66*C^2*
a^6*b^2*c^2*d^10*f^2 - 66*C^2*a^2*b^6*c^10*d^2*f^2 + 52*C^2*a^5*b^3*c^9*d^3
*f^2 + 52*C^2*a^3*b^5*c^3*d^9*f^2 - 27*C^2*a^6*b^2*c^8*d^4*f^2 - 27*C^2*a^2
*b^6*c^4*d^8*f^2 + 24*C^2*a^6*b^2*c^4*d^8*f^2 + 24*C^2*a^4*b^4*c^8*d^4*f^2
+ 24*C^2*a^4*b^4*c^4*d^8*f^2 + 24*C^2*a^2*b^6*c^8*d^4*f^2 - 15*C^2*a^4*b^4*
c^10*d^2*f^2 - 15*C^2*a^4*b^4*c^2*d^10*f^2 - 570*B^2*a^4*b^4*c^6*d^6*f^2 +
366*B^2*a^3*b^5*c^7*d^5*f^2 + 318*B^2*a^5*b^3*c^5*d^7*f^2 - 262*B^2*a^6*b^2
*c^6*d^6*f^2 - 222*B^2*a^2*b^6*c^6*d^6*f^2 - 210*B^2*a^5*b^3*c^3*d^9*f^2 +
186*B^2*a^5*b^3*c^7*d^5*f^2 + 162*B^2*a^3*b^5*c^5*d^7*f^2 - 142*B^2*a^3*b^5
*c^9*d^3*f^2 + 132*B^2*a^4*b^4*c^4*d^8*f^2 + 117*B^2*a^2*b^6*c^4*d^8*f^2 +
102*B^2*a^6*b^2*c^2*d^10*f^2 - 96*B^2*a^3*b^5*c^3*d^9*f^2 + 90*B^2*a^2*b^6*
c^10*d^2*f^2 + 81*B^2*a^4*b^4*c^2*d^10*f^2 - 56*B^2*a^5*b^3*c^9*d^3*f^2 + 4
8*B^2*a^6*b^2*c^4*d^8*f^2 + 48*B^2*a^4*b^4*c^8*d^4*f^2 + 45*B^2*a^6*b^2*c^8
*d^4*f^2 + 36*B^2*a^2*b^6*c^8*d^4*f^2 + 36*B^2*a^2*b^6*c^2*d^10*f^2 + 33*B^
2*a^4*b^4*c^10*d^2*f^2 + 822*A^2*a^4*b^4*c^6*d^6*f^2 - 594*A^2*a^3*b^5*c^7*
d^5*f^2 - 498*A^2*a^5*b^3*c^5*d^7*f^2 + 498*A^2*a^2*b^6*c^6*d^6*f^2 - 414*A
^2*a^3*b^5*c^5*d^7*f^2 + 354*A^2*a^6*b^2*c^6*d^6*f^2 - 318*A^2*a^5*b^3*c^7*
d^5*f^2 + 144*A^2*a^2*b^6*c^8*d^4*f^2 + 102*A^2*a^5*b^3*c^3*d^9*f^2 + 84*A^
2*a^4*b^4*c^4*d^8*f^2 + 81*A^2*a^2*b^6*c^4*d^8*f^2 + 72*A^2*a^4*b^4*c^8*d^4
*f^2 + 70*A^2*a^3*b^5*c^9*d^3*f^2 - 66*A^2*a^6*b^2*c^2*d^10*f^2 + 48*A^2*a^

6*b^2*c^4*d^8*f^2 - 42*A^2*a^2*b^6*c^10*d^2*f^2 + 24*A^2*a^2*b^6*c^2*d^10*f
^2 + 20*A^2*a^5*b^3*c^9*d^3*f^2 - 15*A^2*a^6*b^2*c^8*d^4*f^2 - 15*A^2*a^4*b
^4*c^10*d^2*f^2 - 15*A^2*a^4*b^4*c^2*d^10*f^2 - 12*A^2*a^3*b^5*c^3*d^9*f^2
- 24*B*C*b^8*c^11*d*f^2 + 24*B*C*a^8*c*d^11*f^2 + 12*A*B*b^8*c^11*d*f^2 - 8
*B*C*a^7*b*d^12*f^2 - 24*A*B*a^8*c*d^11*f^2 + 4*B*C*a*b^7*c^12*f^2 + 8*A*B*
a^7*b*d^12*f^2 - 8*A*B*a*b^7*d^12*f^2 - 8*A*B*a*b^7*c^12*f^2 - 174*C^2*a^7*
b*c^5*d^7*f^2 - 174*C^2*a*b^7*c^7*d^5*f^2 + 82*C^2*a^7*b*c^3*d^9*f^2 + 82*C
^2*a*b^7*c^9*d^3*f^2 + 6*C^2*a^7*b*c^7*d^5*f^2 + 6*C^2*a^5*b^3*c*d^11*f^2 +
6*C^2*a^3*b^5*c^11*d*f^2 + 6*C^2*a*b^7*c^5*d^7*f^2 + 162*B^2*a*b^7*c^7*d^5
*f^2 + 138*B^2*a^7*b*c^5*d^7*f^2 - 118*B^2*a^7*b*c^3*d^9*f^2 - 86*B^2*a*b^7
*c^9*d^3*f^2 - 30*B^2*a^5*b^3*c*d^11*f^2 - 18*B^2*a^7*b*c^7*d^5*f^2 - 18*B^
2*a*b^7*c^5*d^7*f^2 - 12*B^2*a^3*b^5*c*d^11*f^2 - 6*B^2*a^3*b^5*c^11*d*f^2
- 4*B^2*a*b^7*c^3*d^9*f^2 - 270*A^2*a*b^7*c^7*d^5*f^2 - 174*A^2*a^7*b*c^5*d
^7*f^2 - 90*A^2*a*b^7*c^5*d^7*f^2 + 82*A^2*a^7*b*c^3*d^9*f^2 + 50*A^2*a*b^7
*c^9*d^3*f^2 - 32*A^2*a*b^7*c^3*d^9*f^2 + 6*A^2*a^7*b*c^7*d^5*f^2 + 6*A^2*a
^5*b^3*c*d^11*f^2 + 6*A^2*a^3*b^5*c^11*d*f^2 + 6*C^2*a^7*b*c*d^11*f^2 + 6*C
^2*a*b^7*c^11*d*f^2 - 18*B^2*a^7*b*c*d^11*f^2 - 6*B^2*a*b^7*c^11*d*f^2 + 6*
A^2*a^7*b*c*d^11*f^2 + 6*A^2*a*b^7*c^11*d*f^2 - 6*A*C*a^8*d^12*f^2 - 2*A*C*
b^8*c^12*f^2 + 33*C^2*b^8*c^8*d^4*f^2 - 27*C^2*b^8*c^10*d^2*f^2 - C^2*b^8*c
^6*d^6*f^2 + 33*C^2*a^8*c^4*d^8*f^2 + 33*B^2*b^8*c^10*d^2*f^2 - 27*C^2*a^8*
c^2*d^10*f^2 - 27*B^2*b^8*c^8*d^4*f^2 + 3*B^2*b^8*c^6*d^6*f^2 - C^2*a^8*c^6
*d^6*f^2 + 117*A^2*b^8*c^8*d^4*f^2 + 111*A^2*b^8*c^6*d^6*f^2 + 72*A^2*b^8*c
^4*d^8*f^2 + 33*B^2*a^8*c^2*d^10*f^2 - 27*B^2*a^8*c^4*d^8*f^2 + 24*A^2*b^8*
c^2*d^10*f^2 + 4*C^2*a^4*b^4*d^12*f^2 + 3*C^2*a^6*b^2*d^12*f^2 + 3*B^2*a^8*
c^6*d^6*f^2 - 3*A^2*b^8*c^10*d^2*f^2 + 33*A^2*a^8*c^4*d^8*f^2 - 27*A^2*a^8*
c^2*d^10*f^2 + 4*C^2*a^4*b^4*c^12*f^2 + 4*B^2*a^4*b^4*d^12*f^2 + 4*B^2*a^2*
b^6*d^12*f^2 + 3*C^2*a^2*b^6*c^12*f^2 + 3*B^2*a^6*b^2*d^12*f^2 - A^2*a^8*c^
6*d^6*f^2 - 4*A^2*a^4*b^4*d^12*f^2 + 3*B^2*a^2*b^6*c^12*f^2 - A^2*a^6*b^2*d
^12*f^2 - A^2*a^2*b^6*c^12*f^2 + 3*C^2*b^8*c^12*f^2 + 3*C^2*a^8*d^12*f^2 +
4*A^2*b^8*d^12*f^2 - B^2*b^8*c^12*f^2 - B^2*a^8*d^12*f^2 + 3*A^2*b^8*c^12*f
^2 + 3*A^2*a^8*d^12*f^2 - 24*A*B*C*a*b^6*c*d^8*f + 342*A*B*C*a^2*b^5*c^4*d^
5*f - 186*A*B*C*a^3*b^4*c^5*d^4*f - 66*A*B*C*a^4*b^3*c^2*d^7*f + 48*A*B*C*a
^2*b^5*c^2*d^7*f + 42*A*B*C*a^2*b^5*c^6*d^3*f + 26*A*B*C*a^5*b^2*c^3*d^6*f
+ 24*A*B*C*a^4*b^3*c^6*d^3*f - 18*A*B*C*a^4*b^3*c^4*d^5*f - 18*A*B*C*a^3*b^
4*c^7*d^2*f - 8*A*B*C*a^3*b^4*c^3*d^6*f + 6*A*B*C*a^5*b^2*c^5*d^4*f - 128*A
*B*C*a*b^6*c^3*d^6*f + 126*A*B*C*a*b^6*c^7*d^2*f + 72*A*B*C*a^3*b^4*c*d^8*f
- 36*A*B*C*a^5*b^2*c*d^8*f - 36*A*B*C*a^2*b^5*c^8*d*f + 30*A*B*C*a^6*b*c^2
*d^7*f - 12*A*B*C*a^6*b*c^4*d^5*f - 12*A*B*C*a*b^6*c^5*d^4*f - 21*B^2*C*a*b
^6*c^8*d*f - 3*B^2*C*a^6*b*c*d^8*f + 21*A^2*C*a*b^6*c^8*d*f - 21*A*C^2*a*b^
6*c^8*d*f - 9*A^2*C*a^6*b*c*d^8*f + 9*A*C^2*a^6*b*c*d^8*f + 36*A^2*B*a*b^6*
c*d^8*f + 21*A*B^2*a*b^6*c^8*d*f + 3*A*B^2*a^6*b*c*d^8*f - 78*A*B*C*b^7*c^6
*d^3*f + 24*A*B*C*b^7*c^4*d^5*f + 2*A*B*C*a^7*c^3*d^6*f + 16*A*B*C*a^4*b^3*
d^9*f - 16*A*B*C*a^2*b^5*d^9*f - 237*B^2*C*a^3*b^4*c^4*d^5*f + 165*B*C^2*a^
3*b^4*c^5*d^4*f + 92*B^2*C*a^2*b^5*c^3*d^6*f - 81*B^2*C*a^2*b^5*c^7*d^2*f +
77*B^2*C*a^4*b^3*c^3*d^6*f - 75*B*C^2*a^2*b^5*c^4*d^5*f + 69*B^2*C*a^4*b^3

*c^5*d^4*f + 69*B*C^2*a^4*b^3*c^4*d^5*f - 68*B*C^2*a^3*b^4*c^3*d^6*f - 63*B
^2*C*a^5*b^2*c^4*d^5*f - 61*B*C^2*a^2*b^5*c^6*d^3*f + 57*B*C^2*a^4*b^3*c^2*
d^7*f - 53*B*C^2*a^5*b^2*c^3*d^6*f - 44*B*C^2*a^4*b^3*c^6*d^3*f - 36*B^2*C*
a^3*b^4*c^2*d^7*f + 35*B^2*C*a^3*b^4*c^6*d^3*f + 33*B^2*C*a^5*b^2*c^2*d^7*f
 - 33*B^2*C*a^2*b^5*c^5*d^4*f + 33*B*C^2*a^3*b^4*c^7*d^2*f - 12*B^2*C*a^4*b
^3*c^7*d^2*f + 9*B*C^2*a^5*b^2*c^5*d^4*f + 4*B^2*C*a^5*b^2*c^6*d^3*f + 225*
A^2*C*a^2*b^5*c^5*d^4*f - 105*A*C^2*a^2*b^5*c^5*d^4*f - 99*A^2*C*a^3*b^4*c^
4*d^5*f - 81*A^2*C*a^5*b^2*c^4*d^5*f + 67*A^2*C*a^4*b^3*c^3*d^6*f - 59*A*C^
2*a^4*b^3*c^3*d^6*f + 57*A*C^2*a^5*b^2*c^2*d^7*f - 57*A*C^2*a^2*b^5*c^7*d^2
*f + 51*A^2*C*a^4*b^3*c^5*d^4*f + 48*A^2*C*a^3*b^4*c^2*d^7*f + 45*A*C^2*a^5
*b^2*c^4*d^5*f - 35*A^2*C*a^3*b^4*c^6*d^3*f - 33*A^2*C*a^5*b^2*c^2*d^7*f +
33*A^2*C*a^2*b^5*c^7*d^2*f + 33*A*C^2*a^4*b^3*c^5*d^4*f + 27*A*C^2*a^3*b^4*
c^6*d^3*f - 24*A*C^2*a^3*b^4*c^2*d^7*f + 24*A*C^2*a^2*b^5*c^3*d^6*f - 21*A*
C^2*a^3*b^4*c^4*d^5*f - 16*A^2*C*a^2*b^5*c^3*d^6*f - 243*A^2*B*a^2*b^5*c^4*
d^5*f - 156*A*B^2*a^2*b^5*c^3*d^6*f + 141*A*B^2*a^3*b^4*c^4*d^5*f + 108*A^2
*B*a^3*b^4*c^3*d^6*f - 105*A*B^2*a^4*b^3*c^3*d^6*f + 84*A*B^2*a^3*b^4*c^2*d
^7*f + 81*A*B^2*a^2*b^5*c^5*d^4*f - 51*A^2*B*a^4*b^3*c^4*d^5*f + 51*A^2*B*a
^2*b^5*c^6*d^3*f - 48*A^2*B*a^2*b^5*c^2*d^7*f + 45*A^2*B*a^3*b^4*c^5*d^4*f
+ 39*A*B^2*a^5*b^2*c^4*d^5*f - 35*A*B^2*a^3*b^4*c^6*d^3*f + 33*A*B^2*a^2*b^
5*c^7*d^2*f + 27*A^2*B*a^5*b^2*c^3*d^6*f - 21*A*B^2*a^4*b^3*c^5*d^4*f + 20*
A^2*B*a^4*b^3*c^6*d^3*f - 15*A^2*B*a^5*b^2*c^5*d^4*f - 15*A^2*B*a^3*b^4*c^7
*d^2*f + 9*A^2*B*a^4*b^3*c^2*d^7*f + 3*A*B^2*a^5*b^2*c^2*d^7*f + 18*A*B*C*b
^7*c^8*d*f - 6*A*B*C*a^7*c*d^8*f + 2*A*B*C*a^6*b*d^9*f - 6*A*B*C*a*b^6*c^9*
f + 63*B^2*C*a*b^6*c^6*d^3*f - 48*B^2*C*a^4*b^3*c*d^8*f + 42*B*C^2*a^2*b^5*
c^8*d*f + 42*B*C^2*a*b^6*c^5*d^4*f - 39*B*C^2*a*b^6*c^7*d^2*f + 30*B*C^2*a^
5*b^2*c*d^8*f - 24*B^2*C*a*b^6*c^4*d^5*f - 24*B*C^2*a^3*b^4*c*d^8*f + 17*B^
2*C*a^6*b*c^3*d^6*f - 15*B*C^2*a^6*b*c^2*d^7*f + 12*B^2*C*a^3*b^4*c^8*d*f +
 12*B^2*C*a^2*b^5*c*d^8*f + 6*B*C^2*a^6*b*c^4*d^5*f - 192*A^2*C*a*b^6*c^4*d
^5*f - 99*A^2*C*a*b^6*c^6*d^3*f + 84*A*C^2*a*b^6*c^4*d^5*f + 59*A*C^2*a*b^6
*c^6*d^3*f + 51*A^2*C*a^6*b*c^3*d^6*f - 51*A*C^2*a^6*b*c^3*d^6*f - 36*A^2*C
*a^2*b^5*c*d^8*f - 24*A*C^2*a^4*b^3*c*d^8*f + 24*A*C^2*a^2*b^5*c*d^8*f + 12
*A^2*C*a^4*b^3*c*d^8*f + 12*A*C^2*a^3*b^4*c^8*d*f + 160*A^2*B*a*b^6*c^3*d^6
*f - 99*A*B^2*a*b^6*c^6*d^3*f - 87*A^2*B*a*b^6*c^7*d^2*f - 72*A*B^2*a*b^6*c
^4*d^5*f - 48*A*B^2*a^2*b^5*c*d^8*f - 36*A^2*B*a^3*b^4*c*d^8*f + 24*A*B^2*a
^4*b^3*c*d^8*f - 17*A*B^2*a^6*b*c^3*d^6*f - 15*A^2*B*a^6*b*c^2*d^7*f + 12*A
*B^2*a*b^6*c^2*d^7*f + 6*A^2*B*a^6*b*c^4*d^5*f + 6*A^2*B*a^5*b^2*c*d^8*f +
6*A^2*B*a^2*b^5*c^8*d*f - 6*A^2*B*a*b^6*c^5*d^4*f + 3*B^2*C*b^7*c^7*d^2*f -
 B*C^2*b^7*c^6*d^3*f + 96*A^2*C*b^7*c^5*d^4*f - 39*A^2*C*b^7*c^7*d^2*f - 36
*A*C^2*b^7*c^5*d^4*f + 32*A^2*C*b^7*c^3*d^6*f + 15*A*C^2*b^7*c^7*d^2*f - 3*
B^2*C*a^7*c^2*d^7*f - B*C^2*a^7*c^3*d^6*f + 111*A^2*B*b^7*c^6*d^3*f - 39*A*
B^2*b^7*c^7*d^2*f + 24*A*B^2*b^7*c^5*d^4*f + 12*B^2*C*a^3*b^4*d^9*f - 12*B*
C^2*a^4*b^3*d^9*f - 9*A^2*C*a^7*c^2*d^7*f + 9*A*C^2*a^7*c^2*d^7*f - 4*A*B^2
*b^7*c^3*d^6*f - 12*A^2*C*a^3*b^4*d^9*f - 8*A*C^2*a^5*b^2*d^9*f + 8*A*C^2*a
^3*b^4*d^9*f + 4*B^2*C*a^2*b^5*c^9*f + 4*A^2*C*a^5*b^2*d^9*f - 4*B*C^2*a^3*
b^4*c^9*f + 3*A*B^2*a^7*c^2*d^7*f - A^2*B*a^7*c^3*d^6*f + 12*A^2*B*a^2*b^5*

d^9*f - 8*A*B^2*a^3*b^4*d^9*f - 4*A^2*B*a^4*b^3*d^9*f + 4*A*C^2*a^2*b^5*c^9
*f - 3*C^3*a^6*b*c*d^8*f + 3*C^3*a*b^6*c^8*d*f + 3*A^3*a^6*b*c*d^8*f - 3*A^
3*a*b^6*c^8*d*f + 3*B*C^2*b^7*c^8*d*f + 12*A^2*C*b^7*c*d^8*f + 3*B*C^2*a^7*
c*d^8*f - 9*A^2*B*b^7*c^8*d*f - B*C^2*a^6*b*d^9*f + 4*A^2*C*a*b^6*d^9*f + 3
*A^2*B*a^7*c*d^8*f + 3*B*C^2*a*b^6*c^9*f + 8*A*B^2*a*b^6*d^9*f - A^2*B*a^6*
b*d^9*f - A^2*B*a*b^6*c^9*f - 39*C^3*a^4*b^3*c^5*d^4*f + 39*C^3*a^3*b^4*c^4
*d^5*f - 27*C^3*a^5*b^2*c^2*d^7*f + 27*C^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^4*b
^3*c^3*d^6*f - 17*C^3*a^3*b^4*c^6*d^3*f - 3*C^3*a^5*b^2*c^4*d^5*f + 3*C^3*a
^2*b^5*c^5*d^4*f - 63*B^3*a^3*b^4*c^5*d^4*f + 57*B^3*a^2*b^5*c^4*d^5*f - 51
*B^3*a^4*b^3*c^2*d^7*f + 48*B^3*a^3*b^4*c^3*d^6*f + 31*B^3*a^2*b^5*c^6*d^3*
f + 27*B^3*a^5*b^2*c^3*d^6*f + 16*B^3*a^4*b^3*c^6*d^3*f - 15*B^3*a^5*b^2*c^
5*d^4*f - 12*B^3*a^2*b^5*c^2*d^7*f + 9*B^3*a^4*b^3*c^4*d^5*f - 3*B^3*a^3*b^
4*c^7*d^2*f - 123*A^3*a^2*b^5*c^5*d^4*f + 81*A^3*a^3*b^4*c^4*d^5*f - 45*A^3
*a^4*b^3*c^5*d^4*f + 39*A^3*a^5*b^2*c^4*d^5*f - 25*A^3*a^4*b^3*c^3*d^6*f +
25*A^3*a^3*b^4*c^6*d^3*f - 24*A^3*a^3*b^4*c^2*d^7*f - 8*A^3*a^2*b^5*c^3*d^6
*f + 3*A^3*a^5*b^2*c^2*d^7*f - 3*A^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^6*b*c^3*d
^6*f - 17*C^3*a*b^6*c^6*d^3*f + 12*C^3*a^4*b^3*c*d^8*f - 12*C^3*a^3*b^4*c^8
*d*f + 24*B^3*a^3*b^4*c*d^8*f + 21*B^3*a*b^6*c^7*d^2*f - 18*B^3*a*b^6*c^5*d
^4*f - 15*B^3*a^6*b*c^2*d^7*f + 6*B^3*a^6*b*c^4*d^5*f + 6*B^3*a^5*b^2*c*d^8
*f - 6*B^3*a^2*b^5*c^8*d*f + 4*B^3*a*b^6*c^3*d^6*f + 108*A^3*a*b^6*c^4*d^5*
f + 57*A^3*a*b^6*c^6*d^3*f - 17*A^3*a^6*b*c^3*d^6*f + 12*A^3*a^2*b^5*c*d^8*
f + 3*C^3*b^7*c^7*d^2*f - 3*C^3*a^7*c^2*d^7*f - B^3*b^7*c^6*d^3*f - 60*A^3*
b^7*c^5*d^4*f - 32*A^3*b^7*c^3*d^6*f + 21*A^3*b^7*c^7*d^2*f + 4*C^3*a^5*b^2
*d^9*f - B^3*a^7*c^3*d^6*f - 4*C^3*a^2*b^5*c^9*f - 4*B^3*a^2*b^5*d^9*f + 3*
A^3*a^7*c^2*d^7*f + 4*A^3*a^3*b^4*d^9*f + 3*B^3*b^7*c^8*d*f - 12*A^3*b^7*c*
d^8*f + 3*B^3*a^7*c*d^8*f - B^3*a^6*b*d^9*f - 4*A^3*a*b^6*d^9*f - B^3*a*b^6
*c^9*f - B^2*C*b^7*c^9*f - 4*A^2*B*b^7*d^9*f + 3*A^2*C*a^7*d^9*f - 3*A*C^2*
a^7*d^9*f - A*C^2*b^7*c^9*f - A*B^2*a^7*d^9*f - C^3*b^7*c^9*f - A^3*a^7*d^9
*f + B^2*C*a^7*d^9*f + A^2*C*b^7*c^9*f + A*B^2*b^7*c^9*f + C^3*a^7*d^9*f +
A^3*b^7*c^9*f - 6*A*B^2*C*a*b^5*c^5*d - 21*A^2*B*C*a^2*b^4*c^3*d^3 + 21*A*B
*C^2*a^2*b^4*c^3*d^3 + 12*A*B^2*C*a^2*b^4*c^4*d^2 - 12*A*B^2*C*a^2*b^4*c^2*
d^4 - 10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^
3*b^3*c^4*d^2 + 3*A^2*B*C*a^3*b^3*c^2*d^4 + 3*A*B^2*C*a^4*b^2*c^2*d^4 + 3*A
*B*C^2*a^3*b^3*c^2*d^4 + 2*A*B*C^2*a^4*b^2*c^3*d^3 - A^2*B*C*a^4*b^2*c^3*d^
3 + 18*A^2*B*C*a*b^5*c^2*d^4 + 10*A*B^2*C*a*b^5*c^3*d^3 + 9*A^2*B*C*a*b^5*c
^4*d^2 - 9*A*B*C^2*a*b^5*c^4*d^2 - 9*A*B*C^2*a*b^5*c^2*d^4 - 6*A^2*B*C*a^2*
b^4*c*d^5 + 6*A*B^2*C*a^3*b^3*c*d^5 - 6*A*B*C^2*a^4*b^2*c*d^5 + 6*A*B*C^2*a
^2*b^4*c^5*d + 3*A^2*B*C*a^4*b^2*c*d^5 - 3*A^2*B*C*a^2*b^4*c^5*d + 3*A*B*C^
2*a^2*b^4*c*d^5 + 3*B^3*C*a^4*b^2*c*d^5 - 3*B^3*C*a^2*b^4*c^5*d + 3*B^3*C*a
*b^5*c^4*d^2 + 3*B^2*C^2*a*b^5*c^5*d + 3*B*C^3*a^4*b^2*c*d^5 - 3*B*C^3*a^2*
b^4*c^5*d + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a*b^5*c^3*d^3 + 8*A*C^3*a*b^5*
c^3*d^3 - 9*A^3*B*a*b^5*c^2*d^4 - 9*A*B^3*a*b^5*c^2*d^4 + 3*A^3*B*a^2*b^4*c
*d^5 - 3*A^3*B*a*b^5*c^4*d^2 + 3*A^2*B^2*a*b^5*c^5*d + 3*A*B^3*a^2*b^4*c*d^
5 - 3*A*B^3*a*b^5*c^4*d^2 - 3*A*B^2*C*b^6*c^4*d^2 - 2*A^2*B*C*b^6*c^3*d^3 +
5*A*B*C^2*a^3*b^3*d^6 - 4*A^2*B*C*a^3*b^3*d^6 - A*B^2*C*a^4*b^2*d^6 + 9*B^

2*C^2*a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^2*b^4*c^4*d^2 + 6*B^2*C^2*a^2*b^4*c^2*d
^4 - 3*B^2*C^2*a^4*b^2*c^2*d^4 + 24*A^2*C^2*a^3*b^3*c^3*d^3 - 15*A^2*C^2*a^
2*b^4*c^4*d^2 - 9*A^2*C^2*a^4*b^2*c^2*d^4 + 3*A^2*C^2*a^2*b^4*c^2*d^4 + 9*A
^2*B^2*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 6*A^2*B*C*b^6*c^5*d -
3*A*B*C^2*b^6*c^5*d + 4*A^2*B*C*a*b^5*d^6 - 2*A*B*C^2*a*b^5*d^6 + 2*A*B*C^2
*a*b^5*c^6 - A^2*B*C*a*b^5*c^6 - 7*B^3*C*a^2*b^4*c^3*d^3 - 7*B*C^3*a^2*b^4*
c^3*d^3 + 3*B^3*C*a^3*b^3*c^4*d^2 - 3*B^3*C*a^3*b^3*c^2*d^4 - 3*B^2*C^2*a^3
*b^3*c*d^5 + 3*B*C^3*a^3*b^3*c^4*d^2 - 3*B*C^3*a^3*b^3*c^2*d^4 - B^3*C*a^4*
b^2*c^3*d^3 - B^2*C^2*a*b^5*c^3*d^3 - B*C^3*a^4*b^2*c^3*d^3 - 24*A^2*C^2*a*
b^5*c^3*d^3 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^2*b^4*c^4*d^2 + 9*A*C^3
*a^4*b^2*c^2*d^4 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^2*b^4*c^4*d^2 - 6*A^
3*C*a^2*b^4*c^2*d^4 + 3*A^3*C*a^4*b^2*c^2*d^4 - 9*A^2*B^2*a*b^5*c^3*d^3 + 7
*A^3*B*a^2*b^4*c^3*d^3 + 7*A*B^3*a^2*b^4*c^3*d^3 - 3*A^3*B*a^3*b^3*c^2*d^4
- 3*A^2*B^2*a^3*b^3*c*d^5 - 3*A*B^3*a^3*b^3*c^2*d^4 + 12*A^2*C^2*b^6*c^4*d^
2 + 3*A^2*C^2*b^6*c^2*d^4 + 6*A^2*B^2*b^6*c^4*d^2 + 3*A^2*B^2*b^6*c^2*d^4 -
5*A^2*C^2*a^2*b^4*d^6 + 3*A^2*C^2*a^4*b^2*d^6 + A*B*C^2*b^6*c^3*d^3 - 3*B^
4*a^3*b^3*c*d^5 - B^4*a*b^5*c^3*d^3 + A^2*B^2*a^3*b^3*c^3*d^3 - 8*A^4*a*b^5
*c^3*d^3 - 15*A^3*C*b^6*c^4*d^2 - 6*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2
- 2*B^3*C*a^3*b^3*d^6 - 2*B*C^3*a^3*b^3*d^6 + 4*A^3*C*a^2*b^4*d^6 - 3*A*C^
3*a^4*b^2*d^6 + 2*A*C^3*a^2*b^4*d^6 - A^3*C*a^4*b^2*d^6 - 2*A*C^3*a^2*b^4*c
^6 + 3*B^4*a*b^5*c^5*d - 3*A^3*B*b^6*c^5*d - 3*A*B^3*b^6*c^5*d - B^3*C*a*b^
5*c^6 - B*C^3*a*b^5*c^6 - 2*A^3*B*a*b^5*d^6 - 2*A*B^3*a*b^5*d^6 + 8*C^4*a^3
*b^3*c^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2 + 6*B^4*a^2*b^
4*c^2*d^4 - 3*B^4*a^2*b^4*c^4*d^2 + 3*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*a^4*b^2
*d^6 + B^2*C^2*a^2*b^4*d^6 + B^2*C^2*a^2*b^4*c^6 + A^2*C^2*a^2*b^4*c^6 - 2*
A^3*C*b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B*a^3*b^3*d^6 +
A*B^3*a^3*b^3*d^6 + 6*A^4*b^6*c^4*d^2 + 3*A^4*b^6*c^2*d^4 - A^4*a^2*b^4*d^
6 - 2*A^2*C^2*b^6*c^6 + A*B^2*C*b^6*c^6 + B^4*a^3*b^3*c^3*d^3 + A^3*C*b^6*c
^6 + A*C^3*b^6*c^6 + C^4*a^4*b^2*d^6 + C^4*a^2*b^4*c^6 + B^4*a^2*b^4*d^6 +
A^2*C^2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k)*((4*a^5*b^4*d^17 - 4
*a^7*b^2*d^17 + 4*b^9*c^5*d^12 + 12*b^9*c^7*d^10 + 8*b^9*c^9*d^8 - 8*b^9*c^
11*d^6 - 12*b^9*c^13*d^4 - 4*b^9*c^15*d^2 - 12*a*b^8*c^4*d^13 - 20*a*b^8*c^
6*d^11 + 48*a*b^8*c^8*d^9 + 152*a*b^8*c^10*d^7 + 148*a*b^8*c^12*d^5 + 60*a*
b^8*c^14*d^3 - 12*a^4*b^5*c*d^16 + 28*a^6*b^3*c*d^16 + 32*a^8*b*c^3*d^14 +
48*a^8*b*c^5*d^12 + 32*a^8*b*c^7*d^10 + 8*a^8*b*c^9*d^8 + 8*a^2*b^7*c^3*d^1
4 - 28*a^2*b^7*c^5*d^12 - 228*a^2*b^7*c^7*d^10 - 472*a^2*b^7*c^9*d^8 - 448*
a^2*b^7*c^11*d^6 - 204*a^2*b^7*c^13*d^4 - 36*a^2*b^7*c^15*d^2 + 8*a^3*b^6*c
^2*d^15 + 68*a^3*b^6*c^4*d^13 + 252*a^3*b^6*c^6*d^11 + 488*a^3*b^6*c^8*d^9
+ 512*a^3*b^6*c^10*d^7 + 276*a^3*b^6*c^12*d^5 + 60*a^3*b^6*c^14*d^3 - 12*a^
4*b^5*c^3*d^14 + 40*a^4*b^5*c^5*d^12 + 40*a^4*b^5*c^7*d^10 - 60*a^4*b^5*c^9
*d^8 - 92*a^4*b^5*c^11*d^6 - 32*a^4*b^5*c^13*d^4 - 44*a^5*b^4*c^2*d^15 - 24
8*a^5*b^4*c^4*d^13 - 472*a^5*b^4*c^6*d^11 - 428*a^5*b^4*c^8*d^9 - 188*a^5*b
^4*c^10*d^7 - 32*a^5*b^4*c^12*d^5 + 172*a^6*b^3*c^3*d^14 + 408*a^6*b^3*c^5*
d^12 + 472*a^6*b^3*c^7*d^10 + 268*a^6*b^3*c^9*d^8 + 60*a^6*b^3*c^11*d^6 - 5
2*a^7*b^2*c^2*d^15 - 168*a^7*b^2*c^4*d^13 - 232*a^7*b^2*c^6*d^11 - 148*a^7*

b^2*c^8*d^9 - 36*a^7*b^2*c^10*d^7 + 8*a*b^8*c^16*d + 8*a^8*b*c*d^16)/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11) + (tan(e + f*x)*(6*a^8*b*d^17 + 6*b^9*c^16*d + 8*a^4*b^5*d^17 + 6*a^6*b^3*d^17 + 8*b^9*c^4*d^13 + 38*b^9*c^6*d^11 + 78*b^9*c^8*d^9 + 92*b^9*c^10*d^7 + 68*b^9*c^12*d^5 + 30*b^9*c^14*d^3 - 32*a*b^8*c^3*d^14 - 148*a*b^8*c^5*d^12 - 292*a*b^8*c^7*d^10 - 328*a*b^8*c^9*d^8 - 232*a*b^8*c^11*d^6 - 100*a*b^8*c^13*d^4 - 20*a*b^8*c^15*d^2 - 2*a^2*b^7*c^16*d - 32*a^3*b^6*c*d^16 - 20*a^5*b^4*c*d^16 - 20*a^7*b^2*c*d^16 + 22*a^8*b*c^2*d^15 + 28*a^8*b*c^4*d^13 + 12*a^8*b*c^6*d^11 - 2*a^8*b*c^8*d^9 - 2*a^8*b*c^10*d^7 + 48*a^2*b^7*c^2*d^15 + 218*a^2*b^7*c^4*d^13 + 400*a^2*b^7*c^6*d^11 + 378*a^2*b^7*c^8*d^9 + 192*a^2*b^7*c^10*d^7 + 46*a^2*b^7*c^12*d^5 - 152*a^3*b^6*c^3*d^14 - 236*a^3*b^6*c^5*d^12 - 52*a^3*b^6*c^7*d^10 + 232*a^3*b^6*c^9*d^8 + 256*a^3*b^6*c^11*d^6 + 100*a^3*b^6*c^13*d^4 + 12*a^3*b^6*c^15*d^2 + 58*a^4*b^5*c^2*d^15 + 60*a^4*b^5*c^4*d^13 - 210*a^4*b^5*c^6*d^11 - 560*a^4*b^5*c^8*d^9 - 522*a^4*b^5*c^10*d^7 - 212*a^4*b^5*c^12*d^5 - 30*a^4*b^5*c^14*d^3 - 28*a^5*b^4*c^3*d^14 + 128*a^5*b^4*c^5*d^12 + 392*a^5*b^4*c^7*d^10 + 428*a^5*b^4*c^9*d^8 + 212*a^5*b^4*c^11*d^6 + 40*a^5*b^4*c^13*d^4 + 32*a^6*b^3*c^2*d^15 + 38*a^6*b^3*c^4*d^13 - 48*a^6*b^3*c^6*d^11 - 142*a^6*b^3*c^8*d^9 - 112*a^6*b^3*c^10*d^7 - 30*a^6*b^3*c^12*d^5 - 68*a^7*b^2*c^3*d^14 - 72*a^7*b^2*c^5*d^12 - 8*a^7*b^2*c^7*d^10 + 28*a^7*b^2*c^9*d^8 + 12*a^7*b^2*c^11*d^6))/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11)) + (B*a^7*b*d^14 - B*b^8*c^13*d - 4*A*a^2*b^6*d^14 + 4*A*a^4*b^4*d^14 - 3*A*a^6*b^2*d^14 + 4*B*a^3*b^5*d^14 - 4*B*a^5*b^3*d^14 - 4*A*b^8*c^2*d^12 - 16*A*b^8*c^4*d^10 - 35*A*b^8*c^6*d^8 - 33*A*b^8*c^8*d^6 - 5*A*b^8*c^10*d^4 + 5*A*b^8*c^12*d^2 - 4*C*a^4*b^4*d^14 + 3*C*a^6*b^2*d^14 - 4*B*b^8*c^5*d^9 + 3*B*b^8*c^7*d^7 + 17*B*b^8*c^9*d^5 + 9*B*b^8*c^11*d^3 + 11*C*b^8*c^6*d^8 + 17*C*b^8*c^8*d^6 + C*b^8*c^10*d^4 - 5*C*b^8*c^12*d^2 + 40*A*a*b^7*c^3*d^11 + 122*A*a*b^7*c^5*d^9 + 175*A*a*b^7*c^7*d^7 + 105*A*a*b^7*c^9*d^5 + 21*A*a*b^7*c^11*d^3 - 6*A*a^5*b^3*c*d^13 + 3*A*a^7*b*c^3*d^11 + 3*A*a^7*b*c^5*d^9 + A*a^7*b*c^7*d^7 + 4*B*a*b^7*c^2*d^12 + 32*B*a*b^7*c^4*d^10 + 31*B*a*b^7*c^6*d^8 - 27*B*a*b^7*c^8*d^6 - 39*B*a*b^7*c^10*d^4 - 9*B*a*b^7*c^12*d^2 - 8*B*a^2*b^6*c*d^13 - 4*B*a^4*b^4*c*d^13 + 5*B*a^6*b^2*c*d^13 + 3*B*a^7*b*c^2*d^12 + 3*B*a^7*b*c^4*d^10 + B*a^7*b*c^6*d^8 - 38*C*a*b^7*c^5*d^9 - 79*C*a*b^7*c^7*d^7 - 41*C*a*b^7*c^9*d^5 + 3*C*a*b^7*c^11*d^3 + 8*C*a^3*b^5*c*d^13 + 10*C*a^5*b^3*c*d^13 - 3*C*a^7*b*c^3*d^11 - 3*C*a^7*b*c^5*d^9 - C*a^7*b*c^7*d^7 - 28*A*a^2*b^6*c^2*d^12 - 117*A*a^2*b^6*c^4* d^10 - 245*A*a^2*b^6*c^6*d^8 - 237*A*a^2*b^6*c^8*d^6 - 91*A*a^2*b^6*c^10*d^4 - 6*A*a^2*b^6*c^12*d^2 - 4*A*a^3*b^5*c^3*d^11 + 67*A*a^3*b^5*c^5*d^9 + 161*A*a^3*b^5*c^7*d^7 + 105*A*a^3*b^5*c^9*d^5 + 15*A*a^3*b^5*c^11*d^3 + 43*A*a^4*b^4*c^2*d^12 + 69*A*a^4*b^4*c^4*d^10 + 5*A*a^4*b^4*c^6*d^8 - 45*A*a^4*b^4*c^8*d^6 - 20*A*a^4*b^4*c^10*d^4 - 35*A*a^5*b^3*c^3*d^11 - 37*A*a^5*b^3*c^5*d^9 + 7*A*a^5*b^3*c^7*d^7 + 15*A*a^5*b^3*c^9*d^5 + A*a^6*b^2*c^2*d^12 + 5*A*a^6*b^2*c^4*d^10 - 5*A*a^6*b^2*c^6*d^8 - 6*A*a^6*b^2*c^8*d^6 - 64*B*a^2*b^6*c^3*d^11 - 145*B*a^2*b^6*c^5*d^9 - 115*B*a^2*b^6*c^7*d^7 - 11*B*a^2*b^6*c^9*d^5 + 15*B*a^2*b^6*c^11*d^3 + 44*B*a^3*b^5*c^2*d^12 + 187*B*a^3*b^5*c^4*d^10 + 273*B*a^3*b^5*c^6*d^8 + 141*B*a^3*b^5*c^8*d^6 + 15*B*a^3*b^5*c^10*d^4 - 71*B*a^4*b^4*c^3*d^11 - 173*B*a^4*b^4*c^5*d^9 - 149*B*a^4*b^4*c^7*d^7 - 43*B*a^4*b^4*c^9*d^5 - 11*B*a^5*b^3*c^2*d^12 + 23*B*a^5*b^3*c^4*d^10 + 63*B*a^5*b^3*c^6*d^8 + 33*B*a^5*b^3*c^8*d^6 - B*a^6*b^2*c^3*d^11 - 17*B*a^6*b^2*c^5*d^9 - 11*B*a^6*b^2*c^7*d^7 - 4*C*a^2*b^6*c^2*d^12 + 25*C*a^2*b^6*c^4*d^10 + 117*C*a^2*b^6*c^6*d^8 + 145*C*a^2*b^6*c^8*d^6 + 59*C*a^2*b^6*c^10*d^4 + 2*C*a^2*b^6*c^12*d^2 + 36*C*a^3*b^5*c^3*d^11 - 19*C*a^3*b^5*c^5*d^9 - 129*C*a^3*b^5*c^7*d^7 - 97*C*a^3*b^5*c^9*d^5 - 15*C*a^3*b^5*c^11*d^3 - 47*C*a^4*b^4*c^2*d^12 - 85*C*a^4*b^4*c^4*d^10 - 29*C*a^4*b^4*c^6*d^8 + 29*C*a^4*b^4*c^8*d^6 + 16*C*a^4*b^4*c^10*d^4 + 51*C*a^5*b^3*c^3*d^11 + 61*C*a^5*b^3*c^5*d^9 + 9*C*a^5*b^3*c^7*d^7 - 11*C*a^5*b^3*c^9*d^5 - C*a^6*b^2*c^2*d^12 - 5*C*a^6*b^2*c^4*d^10 + 5*C*a^6*b^2*c^6*d^8 + 6*C*a^6*b^2*c^8*d^6 + 8*A*a*b^7*c*d^13 + A*a*b^7*c^13*d + A*a^7*b*c*d^13 + 3*C*a*b^7*c^13*d - C*a^7*b*c*d^13)/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11) + (tan(e + f*x)*(3*A*b^8*c^13*d - 3*A*a^7*b*d^14 + 3*C*a^7*b*d^14 + C*b^8*c^13*d + 8*A*a^3*b^5*d^14 - 8*A*a^5*b^3*d^14 - 12*B*a^4*b^4*d^14 - B*a^6*b^2*d^14 + 8*A*b^8*c^3*d^11 + 24*A*b^8*c^5*d^9 + 51*A*b^8*c^7*d^7 + 65*A*b^8*c^9*d^5 + 33*A*b^8*c^11*d^3 + 12*C*a^5*b^3*d^14 - 4*B*b^8*c^4*d^10 + 7*B*b^8*c^6*d^8 + 21*B*b^8*c^8*d^6 + 5*B*b^8*c^10*d^4 - 5*B*b^8*c^12*d^2 + 12*C*b^8*c^5*d^9 + 13*C*b^8*c^7*d^7 - 9*C*b^8*c^9*d^5 - 9*C*b^8*c^11*d^3 - 8*A*a*b^7*c^2*d^12 + 8*A*a*b^7*c^4*d^10 + 3*A*a*b^7*c^6*d^8 - 63*A*a*b^7*c^8*d^6 - 63*A*a*b^7*c^10*d^4 - 13*A*a*b^7*c^12*d^2 - 8*A*a^2*b^6*c*d^13 + 8*A*a^4*b^4*c*d^13 + 13*A*a^6*b^2*c*d^13 - A*a^7*b*c^2*d^12 + 7*A*a^7*b*c^4*d^10 + 5*A*a^7*b*c^6*d^8 + 8*B*a*b^7*c^3*d^11 - 50*B*a*b^7*c^5*d^9 - 143*B*a*b^7*c^7*d^7 - 105*B*a*b^7*c^9*d^5 - 21*B*a*b^7*c^11*d^3 + 24*B*a^3*b^5*c*d^13 + 30*B*a^5*b^3*c*d^13 + 13*B*a^7*b*c^3*d^11 + 5*B*a^7*b*c^5*d^9 - B*a^7*b*c^7*d^7 - 44*C*a*b^7*c^4*d^10 - 67*C*a*b^7*c^6*d^8 + 7*C*a*b^7*c^8*d^6 + 39*C*a*b^7*c^10*d^4 + 9*C*a*b^7*c^12*d^2 - 12*C*a^4*b^4*c*d^13 - 13*C*a^6*b^2*c*d^13 + C*a^7*b*c^2*d^12 - 7*C*a^7*b*c^4*d^10 - 5*C*a^7*b*c^6*d^8 - 96*A*a^2*b^6*c^3*d^11 - 233*A*a^2*b^6*c^5*d^9 - 195*A*a^2*b^6*c^7*d^7 - 35*A*a^2*b^6*c^9*d^5 + 15*A*a^2*b^6*c^11*d^3 + 64*A*a^3*b^5*c^2*d^12 + 263*A*a^3*b^5*c^4*d^

```
10 + 381*A*a^3*b^5*c^6*d^8 + 189*A*a^3*b^5*c^8*d^6 + 15*A*a^3*b^5*c^10*d^4
 - 87*A*a^4*b^4*c^3*d^11 - 253*A*a^4*b^4*c^5*d^9 - 213*A*a^4*b^4*c^7*d^7 - 5
5*A*a^4*b^4*c^9*d^5 - 7*A*a^5*b^3*c^2*d^12 + 67*A*a^5*b^3*c^4*d^10 + 123*A*
a^5*b^3*c^6*d^8 + 57*A*a^5*b^3*c^8*d^6 - A*a^6*b^2*c^3*d^11 - 41*A*a^6*b^2*
c^5*d^9 - 27*A*a^6*b^2*c^7*d^7 - 16*B*a^2*b^6*c^2*d^12 + 17*B*a^2*b^6*c^4*d
^10 + 161*B*a^2*b^6*c^6*d^8 + 213*B*a^2*b^6*c^8*d^6 + 91*B*a^2*b^6*c^10*d^4
 + 6*B*a^2*b^6*c^12*d^2 + 116*B*a^3*b^5*c^3*d^11 + 85*B*a^3*b^5*c^5*d^9 - 9
7*B*a^3*b^5*c^7*d^7 - 105*B*a^3*b^5*c^9*d^5 - 15*B*a^3*b^5*c^11*d^3 - 119*B
*a^4*b^4*c^2*d^12 - 209*B*a^4*b^4*c^4*d^10 - 89*B*a^4*b^4*c^6*d^8 + 33*B*a^
4*b^4*c^8*d^6 + 20*B*a^4*b^4*c^10*d^4 + 115*B*a^5*b^3*c^3*d^11 + 125*B*a^5*
b^3*c^5*d^9 + 25*B*a^5*b^3*c^7*d^7 - 15*B*a^5*b^3*c^9*d^5 - 37*B*a^6*b^2*c^
2*d^12 - 65*B*a^6*b^2*c^4*d^10 - 23*B*a^6*b^2*c^6*d^8 + 6*B*a^6*b^2*c^8*d^6
 + 64*C*a^2*b^6*c^3*d^11 + 185*C*a^2*b^6*c^5*d^9 + 163*C*a^2*b^6*c^7*d^7 +
27*C*a^2*b^6*c^9*d^5 - 15*C*a^2*b^6*c^11*d^3 - 32*C*a^3*b^5*c^2*d^12 - 215*
C*a^3*b^5*c^4*d^10 - 349*C*a^3*b^5*c^6*d^8 - 181*C*a^3*b^5*c^8*d^6 - 15*C*a
^3*b^5*c^10*d^4 + 71*C*a^4*b^4*c^3*d^11 + 229*C*a^4*b^4*c^5*d^9 + 197*C*a^4
*b^4*c^7*d^7 + 51*C*a^4*b^4*c^9*d^5 + 23*C*a^5*b^3*c^2*d^12 - 43*C*a^5*b^3*
c^4*d^10 - 107*C*a^5*b^3*c^6*d^8 - 53*C*a^5*b^3*c^8*d^6 + C*a^6*b^2*c^3*d^1
1 + 41*C*a^6*b^2*c^5*d^9 + 27*C*a^6*b^2*c^7*d^7 - B*a*b^7*c^13*d + 7*B*a^7*
b*c*d^13))/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^
6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c
^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*
c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*
c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 + 24
*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11)) -
 (4*A^2*a^3*b^4*d^11 - A^2*a^5*b^2*d^11 - B^2*a^5*b^2*d^11 - 28*A^2*b^7*c^3
*d^8 - 45*A^2*b^7*c^5*d^6 - 24*A^2*b^7*c^7*d^4 + A^2*b^7*c^9*d^2 - C^2*a^5*
b^2*d^11 - B^2*b^7*c^5*d^6 - 3*B^2*b^7*c^9*d^2 - C^2*b^7*c^5*d^6 - 4*C^2*b^
7*c^7*d^4 + C^2*b^7*c^9*d^2 - 4*A^2*a*b^6*d^11 - 4*A^2*b^7*c*d^10 + 14*A^2*
a^2*b^5*c^3*d^8 - 154*A^2*a^2*b^5*c^5*d^6 + 28*A^2*a^2*b^5*c^7*d^4 - 26*A^2
*a^3*b^4*c^2*d^9 + 72*A^2*a^3*b^4*c^4*d^7 - 42*A^2*a^3*b^4*c^6*d^5 - 24*A^2
*a^4*b^3*c^3*d^8 + 33*A^2*a^4*b^3*c^5*d^6 + 10*A^2*a^5*b^2*c^2*d^9 - 13*A^2
*a^5*b^2*c^4*d^7 - 46*B^2*a^2*b^5*c^3*d^8 + 102*B^2*a^2*b^5*c^5*d^6 - 52*B^
2*a^2*b^5*c^7*d^4 + 34*B^2*a^3*b^4*c^2*d^9 - 68*B^2*a^3*b^4*c^4*d^7 + 42*B^
2*a^3*b^4*c^6*d^5 + 36*B^2*a^4*b^3*c^3*d^8 - 27*B^2*a^4*b^3*c^5*d^6 - 14*B^
2*a^5*b^2*c^2*d^9 + 11*B^2*a^5*b^2*c^4*d^7 + 10*C^2*a^2*b^5*c^3*d^8 - 134*C
^2*a^2*b^5*c^5*d^6 + 48*C^2*a^2*b^5*c^7*d^4 + 4*C^2*a^2*b^5*c^9*d^2 - 22*C^
2*a^3*b^4*c^2*d^9 + 92*C^2*a^3*b^4*c^4*d^7 - 30*C^2*a^3*b^4*c^6*d^5 - 24*C^
2*a^4*b^3*c^3*d^8 + 33*C^2*a^4*b^3*c^5*d^6 + 10*C^2*a^5*b^2*c^2*d^9 - 13*C^
2*a^5*b^2*c^4*d^7 + 4*A*B*a^2*b^5*d^11 - 4*A*C*a^3*b^4*d^11 + 2*A*C*a^5*b^2
*d^11 - 4*A*B*b^7*c^2*d^9 + 4*A*B*b^7*c^4*d^7 + 19*A*B*b^7*c^6*d^5 + 18*A*B
*b^7*c^8*d^3 + 12*A*C*b^7*c^3*d^8 + 22*A*C*b^7*c^5*d^6 + 12*A*C*b^7*c^7*d^4
 - 6*A*C*b^7*c^9*d^2 + B*C*b^7*c^6*d^5 - 6*B*C*b^7*c^8*d^3 - 2*A^2*a^6*b*c*
d^10 + 2*B^2*a^6*b*c*d^10 + 4*C^2*a*b^6*c^10*d - 2*C^2*a^6*b*c*d^10 + 8*A^2
*a*b^6*c^2*d^9 + 63*A^2*a*b^6*c^4*d^7 + 130*A^2*a*b^6*c^6*d^5 - 9*A^2*a*b^6
```

*c^8*d^3 + 8*A^2*a^2*b^5*c*d^10 + 3*A^2*a^4*b^3*c*d^10 + 2*A^2*a^6*b*c^3*d^8 + 4*B^2*a*b^6*c^2*d^9 + 3*B^2*a*b^6*c^4*d^7 - 50*B^2*a*b^6*c^6*d^5 + 39*B^2*a*b^6*c^8*d^3 - 12*B^2*a^2*b^5*c*d^10 + 3*B^2*a^4*b^3*c*d^10 - 2*B^2*a^6*b*c^3*d^8 + 3*C^2*a*b^6*c^4*d^7 + 54*C^2*a*b^6*c^6*d^5 - 33*C^2*a*b^6*c^8*d^3 + 3*C^2*a^4*b^3*c*d^10 + 2*C^2*a^6*b*c^3*d^8 - A*B*a^6*b*d^11 - A*B*b^7*c^10*d + B*C*a^6*b*d^11 + B*C*b^7*c^10*d + 16*A*B*a*b^6*c*d^10 + 4*A*C*a^6*b*c*d^10 + 56*A*B*a*b^6*c^3*d^8 + 70*A*B*a*b^6*c^5*d^6 - 140*A*B*a*b^6*c^7*d^4 + 6*A*B*a*b^6*c^9*d^2 - 24*A*B*a^3*b^4*c*d^10 + 6*A*B*a^5*b^2*c*d^10 + 6*A*B*a^6*b*c^2*d^9 - A*B*a^6*b*c^4*d^7 - 20*A*C*a*b^6*c^2*d^9 - 74*A*C*a*b^6*c^4*d^7 - 176*A*C*a*b^6*c^6*d^5 + 54*A*C*a*b^6*c^8*d^3 - 4*A*C*a^2*b^5*c*d^10 - 6*A*C*a^4*b^3*c*d^10 - 4*A*C*a^6*b*c^3*d^8 - 12*B*C*a*b^6*c^3*d^8 - 50*B*C*a*b^6*c^5*d^6 + 112*B*C*a*b^6*c^7*d^4 - 26*B*C*a*b^6*c^9*d^2 + 12*B*C*a^3*b^4*c*d^10 - 6*B*C*a^5*b^2*c*d^10 - 6*B*C*a^6*b*c^2*d^9 + B*C*a^6*b*c^4*d^7 - 20*A*B*a^2*b^5*c^2*d^9 - 195*A*B*a^2*b^5*c^4*d^7 + 190*A*B*a^2*b^5*c^6*d^5 - 15*A*B*a^2*b^5*c^8*d^3 + 100*A*B*a^3*b^4*c^3*d^8 - 144*A*B*a^3*b^4*c^5*d^6 + 20*A*B*a^3*b^4*c^7*d^4 - 15*A*B*a^4*b^3*c^2*d^9 + 90*A*B*a^4*b^3*c^4*d^7 - 15*A*B*a^4*b^3*c^6*d^5 - 36*A*B*a^5*b^2*c^3*d^8 + 6*A*B*a^5*b^2*c^5*d^6 - 8*A*C*a^2*b^5*c^3*d^8 + 312*A*C*a^2*b^5*c^5*d^6 - 60*A*C*a^2*b^5*c^7*d^4 + 48*A*C*a^3*b^4*c^2*d^9 - 164*A*C*a^3*b^4*c^4*d^7 + 72*A*C*a^3*b^4*c^6*d^5 + 48*A*C*a^4*b^3*c^3*d^8 - 66*A*C*a^4*b^3*c^5*d^6 - 20*A*C*a^5*b^2*c^2*d^9 + 26*A*C*a^5*b^2*c^4*d^7 + 16*B*C*a^2*b^5*c^2*d^9 + 175*B*C*a^2*b^5*c^4*d^7 - 202*B*C*a^2*b^5*c^6*d^5 + 15*B*C*a^2*b^5*c^8*d^3 - 120*B*C*a^3*b^4*c^3*d^8 + 140*B*C*a^3*b^4*c^5*d^6 - 16*B*C*a^3*b^4*c^7*d^4 + 15*B*C*a^4*b^3*c^2*d^9 - 90*B*C*a^4*b^3*c^4*d^7 + 15*B*C*a^4*b^3*c^6*d^5 + 36*B*C*a^5*b^2*c^3*d^8 - 6*B*C*a^5*b^2*c^5*d^6)/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11) + (tan(e + f*x)*(2*A^2*b^7*d^11 - 6*A^2*a^2*b^5*d^11 + 2*A^2*a^4*b^3*d^11 + 2*B^2*a^2*b^5*d^11 + 2*B^2*a^4*b^3*d^11 + 6*A^2*b^7*c^2*d^9 - 12*A^2*b^7*c^4*d^7 - 66*A^2*b^7*c^6*d^5 + 18*A^2*b^7*c^8*d^3 + 4*C^2*a^4*b^3*d^11 - 2*B^2*b^7*c^4*d^7 + 29*B^2*b^7*c^6*d^5 - 36*B^2*b^7*c^8*d^3 + 2*C^2*b^7*c^4*d^7 - 32*C^2*b^7*c^6*d^5 + 30*C^2*b^7*c^8*d^3 + B^2*a^6*b*d^11 + B^2*b^7*c^10*d - 4*C^2*b^7*c^10*d + 38*A^2*a^2*b^5*c^2*d^9 - 2*A^2*a^2*b^5*c^4*d^7 + 78*A^2*a^2*b^5*c^6*d^5 - 16*A^2*a^3*b^4*c^3*d^8 - 88*A^2*a^3*b^4*c^5*d^6 + 4*A^2*a^4*b^3*c^2*d^9 + 62*A^2*a^4*b^3*c^4*d^7 - 24*A^2*a^5*b^2*c^3*d^8 - 8*B^2*a^2*b^5*c^2*d^9 + 83*B^2*a^2*b^5*c^4*d^7 - 22*B^2*a^2*b^5*c^6*d^5 + 9*B^2*a^2*b^5*c^8*d^3 - 46*B^2*a^3*b^4*c^3*d^8 + 30*B^2*a^3*b^4*c^5*d^6 - 18*B^2*a^3*b^4*c^7*d^4 + 19*B^2*a^4*b^3*c^2*d^9 - 28*B^2*a^4*b^3*c^4*d^7 + 15*B^2*a^4*b^3*c^6*d^5 + 12*B^2*a^5*b^2*c^3*d^8 - 6*B^2*a^5*b^2*c^5*d^6 + 12*C^2*a^2*b^5*c^2*d^9 - 82*C^2*a^2*b^5*c^4*d^7 + 22*C^2*a^2*b^5*c^6*d^5 - 6*C^2*a^2*b^5*c^8*d^3 - 56*C^2*a^3*b^4*c^5*d^6 + 16*C^2*a^3*b^4*c^7*d^4 + 2*C^2*a^4*b^3*c^2*d^9 + 52*C^2*a^4*b^3*c^4*d^7 - 6*C^

2*a^4*b^3*c^6*d^5 - 24*C^2*a^5*b^2*c^3*d^8 + 2*A*B*a^3*b^4*d^11 + 4*A*C*a^2
*b^5*d^11 - 6*A*C*a^4*b^3*d^11 - 6*A*B*b^7*c^3*d^8 - 18*A*B*b^7*c^5*d^6 + 1
14*A*B*b^7*c^7*d^4 - 10*A*B*b^7*c^9*d^2 - 4*B*C*a^3*b^4*d^11 + 14*A*C*b^7*c
^4*d^7 + 94*A*C*b^7*c^6*d^5 - 54*A*C*b^7*c^8*d^3 + 24*B*C*b^7*c^5*d^6 - 84*
B*C*b^7*c^7*d^4 + 28*B*C*b^7*c^9*d^2 - 8*A^2*a*b^6*c*d^10 - 40*A^2*a*b^6*c^
3*d^8 + 72*A^2*a*b^6*c^5*d^6 - 48*A^2*a*b^6*c^7*d^4 - 8*A^2*a^3*b^4*c*d^10
+ 4*A^2*a^6*b*c^2*d^9 + 14*B^2*a*b^6*c^3*d^8 - 100*B^2*a*b^6*c^5*d^6 + 38*B
^2*a*b^6*c^7*d^4 - 14*B^2*a^3*b^4*c*d^10 - 6*B^2*a^5*b^2*c*d^10 - 2*B^2*a^6
*b*c^2*d^9 + B^2*a^6*b*c^4*d^7 - 8*C^2*a*b^6*c^3*d^8 + 104*C^2*a*b^6*c^5*d^
6 - 48*C^2*a*b^6*c^7*d^4 - 8*C^2*a*b^6*c^9*d^2 + 2*C^2*a^2*b^5*c^10*d - 8*C
^2*a^3*b^4*c*d^10 + 4*C^2*a^6*b*c^2*d^9 - 4*A*B*a*b^6*d^11 + 2*A*C*b^7*c^10
*d + 4*A*B*a^6*b*c*d^10 - 2*B*C*a*b^6*c^10*d - 4*B*C*a^6*b*c*d^10 - 10*A*B*
a*b^6*c^2*d^9 + 114*A*B*a*b^6*c^4*d^7 - 166*A*B*a*b^6*c^6*d^5 + 18*A*B*a*b^
6*c^8*d^3 + 30*A*B*a^2*b^5*c*d^10 - 4*A*B*a^6*b*c^3*d^8 + 16*A*C*a*b^6*c^3*
d^8 - 224*A*C*a*b^6*c^5*d^6 + 64*A*C*a*b^6*c^7*d^4 + 16*A*C*a^3*b^4*c*d^10
- 8*A*C*a^6*b*c^2*d^9 - 106*B*C*a*b^6*c^4*d^7 + 194*B*C*a*b^6*c^6*d^5 - 6*B
*C*a*b^6*c^8*d^3 + 6*B*C*a^4*b^3*c*d^10 + 4*B*C*a^6*b*c^3*d^8 - 54*A*B*a^2*
b^5*c^3*d^8 + 118*A*B*a^2*b^5*c^5*d^6 - 46*A*B*a^2*b^5*c^7*d^4 - 2*A*B*a^3*
b^4*c^2*d^9 - 90*A*B*a^3*b^4*c^4*d^7 + 74*A*B*a^3*b^4*c^6*d^5 + 60*A*B*a^4*
b^3*c^3*d^8 - 60*A*B*a^4*b^3*c^5*d^6 - 24*A*B*a^5*b^2*c^2*d^9 + 24*A*B*a^5*
b^2*c^4*d^7 - 56*A*C*a^2*b^5*c^2*d^9 + 80*A*C*a^2*b^5*c^4*d^7 - 96*A*C*a^2*
b^5*c^6*d^5 + 12*A*C*a^2*b^5*c^8*d^3 + 16*A*C*a^3*b^4*c^3*d^8 + 144*A*C*a^3
*b^4*c^5*d^6 - 16*A*C*a^3*b^4*c^7*d^4 - 6*A*C*a^4*b^3*c^2*d^9 - 114*A*C*a^4
*b^3*c^4*d^7 + 6*A*C*a^4*b^3*c^6*d^5 + 48*A*C*a^5*b^2*c^3*d^8 + 106*B*C*a^2
*b^5*c^3*d^8 - 110*B*C*a^2*b^5*c^5*d^6 + 26*B*C*a^2*b^5*c^7*d^4 - 6*B*C*a^2
*b^5*c^9*d^2 - 14*B*C*a^3*b^4*c^2*d^9 + 70*B*C*a^3*b^4*c^4*d^7 - 74*B*C*a^3
*b^4*c^6*d^5 + 6*B*C*a^3*b^4*c^8*d^3 - 50*B*C*a^4*b^3*c^3*d^8 + 62*B*C*a^4*
b^3*c^5*d^6 - 2*B*C*a^4*b^3*c^7*d^4 + 24*B*C*a^5*b^2*c^2*d^9 - 24*B*C*a^5*b
^2*c^4*d^7))/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*
c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4
*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^
3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*
b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 +
24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11))
- (A^3*a^2*b^4*d^8 - A^3*b^6*d^8 - 4*A^3*b^6*c^2*d^6 - 7*A^3*b^6*c^4*d^4 +
A^2*C*b^6*d^8 - 3*A^3*a^2*b^4*c^2*d^6 - B^3*a^2*b^4*c^3*d^5 - C^3*a^2*b^4*
c^2*d^6 + 7*C^3*a^2*b^4*c^4*d^4 - 2*C^3*a^3*b^3*c^3*d^5 + A^2*B*a*b^5*d^8 +
A^2*B*b^6*c*d^7 + A^3*a*b^5*c*d^7 + C^3*a*b^5*c^7*d + A*C^2*a^2*b^4*d^8 -
2*A^2*C*a^2*b^4*d^8 - A*B^2*b^6*c^2*d^6 - 3*A*B^2*b^6*c^6*d^2 - B*C^2*a^3*b
^3*d^8 + 2*A^2*B*b^6*c^3*d^5 + 9*A^2*B*b^6*c^5*d^3 + B^2*C*a^2*b^4*d^8 - A*
C^2*b^6*c^2*d^6 - 4*A*C^2*b^6*c^4*d^4 + A*C^2*b^6*c^6*d^2 + 5*A^2*C*b^6*c^2
*d^6 + 11*A^2*C*b^6*c^4*d^4 - A^2*C*b^6*c^6*d^2 + 9*A^3*a*b^5*c^3*d^5 + B^3
*a*b^5*c^2*d^6 + B^3*a*b^5*c^4*d^4 - B^3*a^2*b^4*c*d^7 - 3*C^3*a*b^5*c^5*d^
3 + 2*C^3*a^3*b^3*c*d^7 - 2*A*B*C*a*b^5*d^8 + A*B*C*b^6*c^7*d + 3*A*B^2*a^2
*b^4*c^2*d^6 - A*B^2*a^2*b^4*c^4*d^4 + 3*A^2*B*a^2*b^4*c^3*d^5 - A*C^2*a^2*

b^4*c^2*d^6 - 14*A*C^2*a^2*b^4*c^4*d^4 + 4*A*C^2*a^3*b^3*c^3*d^5 + 5*A^2*C*
a^2*b^4*c^2*d^6 + 7*A^2*C*a^2*b^4*c^4*d^4 - 2*A^2*C*a^3*b^3*c^3*d^5 - 15*B*
C^2*a^2*b^4*c^3*d^5 + 3*B*C^2*a^2*b^4*c^5*d^3 + 6*B*C^2*a^3*b^3*c^2*d^6 - B
*C^2*a^3*b^3*c^4*d^4 + 5*B^2*C*a^2*b^4*c^2*d^6 - 4*B^2*C*a^2*b^4*c^4*d^4 +
2*B^2*C*a^3*b^3*c^3*d^5 + A*B*C*a^3*b^3*d^8 + A*B*C*b^6*c^3*d^5 - 6*A*B*C*b
^6*c^5*d^3 + 2*A*C^2*a*b^5*c*d^7 - A*C^2*a*b^5*c^7*d - 3*A^2*C*a*b^5*c*d^7
- 5*A*B^2*a*b^5*c^3*d^5 + 3*A*B^2*a*b^5*c^5*d^3 + 7*A^2*B*a*b^5*c^2*d^6 - 1
0*A^2*B*a*b^5*c^4*d^4 - 5*A^2*B*a^2*b^4*c*d^7 + 12*A*C^2*a*b^5*c^3*d^5 + 9*
A*C^2*a*b^5*c^5*d^3 - 4*A*C^2*a^3*b^3*c*d^7 - 21*A^2*C*a*b^5*c^3*d^5 - 6*A^
2*C*a*b^5*c^5*d^3 + 2*A^2*C*a^3*b^3*c*d^7 + B*C^2*a*b^5*c^2*d^6 + 5*B*C^2*a
*b^5*c^4*d^4 - 4*B*C^2*a*b^5*c^6*d^2 - 2*B*C^2*a^2*b^4*c*d^7 - B^2*C*a*b^5*
c^3*d^5 + 3*B^2*C*a*b^5*c^5*d^3 - 2*B^2*C*a^3*b^3*c*d^7 + 12*A*B*C*a^2*b^4*
c^3*d^5 - 3*A*B*C*a^2*b^4*c^5*d^3 - 6*A*B*C*a^3*b^3*c^2*d^6 + A*B*C*a^3*b^3
*c^4*d^4 - 11*A*B*C*a*b^5*c^2*d^6 + 2*A*B*C*a*b^5*c^4*d^4 + 3*A*B*C*a*b^5*c
^6*d^2 + 7*A*B*C*a^2*b^4*c*d^7)/(a^4*d^12 + b^4*c^12 + 4*a^4*c^2*d^10 + 6*a
^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*b^4*c^6*d^6 + 6*
b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c^5*d^7 - 24*a*b^
3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3
*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36
*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d
 - 4*a^3*b*c*d^11) - (tan(e + f*x)*(B^3*b^6*c^4*d^4 - A^3*b^6*c^3*d^5 - B^3
*a^2*b^4*d^8 - 3*B^3*b^6*c^6*d^2 - 3*C^3*b^6*c^5*d^3 - A^2*B*b^6*d^8 + A^3*
a*b^5*d^8 - A^3*b^6*c*d^7 + C^3*b^6*c^7*d + 2*B^3*a^2*b^4*c^2*d^6 - B^3*a^2
*b^4*c^4*d^4 - 12*C^3*a^2*b^4*c^3*d^5 + 4*C^3*a^3*b^3*c^2*d^6 + 2*A*B^2*a*b
^5*d^8 - A^2*C*a*b^5*d^8 - A*C^2*b^6*c^7*d + A^2*C*b^6*c*d^7 + B^2*C*b^6*c^
7*d + A*B^2*b^6*c^3*d^5 + 9*A*B^2*b^6*c^5*d^3 - 3*A^2*B*b^6*c^2*d^6 - 6*A^2
*B*b^6*c^4*d^4 + B^2*C*a^3*b^3*d^8 + 2*A*C^2*b^6*c^3*d^5 + 9*A*C^2*b^6*c^5*
d^3 - A^2*C*b^6*c^3*d^5 - 6*A^2*C*b^6*c^5*d^3 + B*C^2*b^6*c^4*d^4 - 3*B*C^2
*b^6*c^6*d^2 - 3*B^2*C*b^6*c^5*d^3 + A^3*a*b^5*c^2*d^6 - 5*B^3*a*b^5*c^3*d^
5 + 3*B^3*a*b^5*c^5*d^3 + 11*C^3*a*b^5*c^4*d^4 - C^3*a*b^5*c^6*d^2 + 4*A*B^
2*a^2*b^4*c^3*d^5 - 4*A^2*B*a^2*b^4*c^2*d^6 + 24*A*C^2*a^2*b^4*c^3*d^5 - 8*
A*C^2*a^3*b^3*c^2*d^6 - 12*A^2*C*a^2*b^4*c^3*d^5 + 4*A^2*C*a^3*b^3*c^2*d^6
+ 8*B*C^2*a^2*b^4*c^2*d^6 - 12*B*C^2*a^2*b^4*c^4*d^4 + 4*B*C^2*a^3*b^3*c^3*
d^5 + 2*B^2*C*a^2*b^4*c^3*d^5 - 3*B^2*C*a^2*b^4*c^5*d^3 - 2*B^2*C*a^3*b^3*c
^2*d^6 + B^2*C*a^3*b^3*c^4*d^4 + 2*A*B*C*b^6*c^4*d^4 + 2*A*B*C*b^6*c^6*d^2
+ A^2*B*a*b^5*c*d^7 - B*C^2*a*b^5*c^7*d + 7*A*B^2*a*b^5*c^2*d^6 - 11*A*B^2*
a*b^5*c^4*d^4 - 4*A*B^2*a^2*b^4*c*d^7 + 9*A^2*B*a*b^5*c^3*d^5 - 2*A*C^2*a*b
^5*c^2*d^6 - 25*A*C^2*a*b^5*c^4*d^4 + A*C^2*a*b^5*c^6*d^2 + A^2*C*a*b^5*c^2
*d^6 + 14*A^2*C*a*b^5*c^4*d^4 - 6*B*C^2*a*b^5*c^3*d^5 + 9*B*C^2*a*b^5*c^5*d
^3 - 4*B*C^2*a^3*b^3*c*d^7 + 7*B^2*C*a*b^5*c^4*d^4 + 3*B^2*C*a*b^5*c^6*d^2
+ B^2*C*a^2*b^4*c*d^7 - 4*A*B*C*a^2*b^4*c^2*d^6 + 12*A*B*C*a^2*b^4*c^4*d^4
- 4*A*B*C*a^3*b^3*c^3*d^5 - 2*A*B*C*a*b^5*c*d^7 - 6*A*B*C*a*b^5*c^3*d^5 - 1
2*A*B*C*a*b^5*c^5*d^3 + 4*A*B*C*a^3*b^3*c*d^7))/(a^4*d^12 + b^4*c^12 + 4*a^
4*c^2*d^10 + 6*a^4*c^4*d^8 + 4*a^4*c^6*d^6 + a^4*c^8*d^4 + b^4*c^4*d^8 + 4*
b^4*c^6*d^6 + 6*b^4*c^8*d^4 + 4*b^4*c^10*d^2 - 4*a*b^3*c^3*d^9 - 16*a*b^3*c

^5*d^7 - 24*a*b^3*c^7*d^5 - 16*a*b^3*c^9*d^3 - 16*a^3*b*c^3*d^9 - 24*a^3*b*c^5*d^7 - 16*a^3*b*c^7*d^5 - 4*a^3*b*c^9*d^3 + 6*a^2*b^2*c^2*d^10 + 24*a^2*b^2*c^4*d^8 + 36*a^2*b^2*c^6*d^6 + 24*a^2*b^2*c^8*d^4 + 6*a^2*b^2*c^10*d^2 - 4*a*b^3*c^11*d - 4*a^3*b*c*d^11))*root(480*a^9*b*c^7*d^11*f^4 + 480*a*b^9*c^11*d^7*f^4 + 360*a^9*b*c^9*d^9*f^4 + 360*a^9*b*c^5*d^13*f^4 + 360*a*b^9*c^13*d^5*f^4 + 360*a*b^9*c^9*d^9*f^4 + 144*a^9*b*c^11*d^7*f^4 + 144*a^9*b*c^3*d^15*f^4 + 144*a*b^9*c^15*d^3*f^4 + 144*a*b^9*c^7*d^11*f^4 + 48*a^7*b^3*c*d^17*f^4 + 48*a^3*b^7*c^17*d*f^4 + 24*a^9*b*c^13*d^5*f^4 + 24*a^5*b^5*c^17*d*f^4 + 24*a^5*b^5*c*d^17*f^4 + 24*a*b^9*c^5*d^13*f^4 + 24*a^9*b*c*d^17*f^4 + 24*a*b^9*c^17*d*f^4 + 3920*a^5*b^5*c^9*d^9*f^4 - 3360*a^6*b^4*c^8*d^10*f^4 - 3360*a^4*b^6*c^10*d^8*f^4 - 3024*a^6*b^4*c^10*d^8*f^4 + 3024*a^5*b^5*c^11*d^7*f^4 + 3024*a^5*b^5*c^7*d^11*f^4 - 3024*a^4*b^6*c^8*d^10*f^4 + 2320*a^7*b^3*c^9*d^9*f^4 + 2320*a^3*b^7*c^9*d^9*f^4 - 2240*a^6*b^4*c^6*d^12*f^4 - 2240*a^4*b^6*c^12*d^6*f^4 + 2160*a^7*b^3*c^7*d^11*f^4 + 2160*a^3*b^7*c^11*d^7*f^4 - 1624*a^6*b^4*c^12*d^6*f^4 - 1624*a^4*b^6*c^6*d^12*f^4 + 1488*a^7*b^3*c^11*d^7*f^4 + 1488*a^3*b^7*c^7*d^11*f^4 + 1344*a^5*b^5*c^13*d^5*f^4 + 1344*a^5*b^5*c^5*d^13*f^4 - 1320*a^8*b^2*c^8*d^10*f^4 - 1320*a^2*b^8*c^10*d^8*f^4 + 1200*a^7*b^3*c^5*d^13*f^4 + 1200*a^3*b^7*c^13*d^5*f^4 - 1060*a^8*b^2*c^6*d^12*f^4 - 1060*a^2*b^8*c^12*d^6*f^4 - 948*a^8*b^2*c^10*d^8*f^4 - 948*a^2*b^8*c^8*d^10*f^4 - 840*a^6*b^4*c^4*d^14*f^4 - 840*a^4*b^6*c^14*d^4*f^4 + 528*a^7*b^3*c^13*d^5*f^4 + 528*a^3*b^7*c^5*d^13*f^4 - 480*a^8*b^2*c^4*d^14*f^4 - 480*a^6*b^4*c^14*d^4*f^4 - 480*a^4*b^6*c^4*d^14*f^4 - 480*a^2*b^8*c^14*d^4*f^4 - 368*a^8*b^2*c^12*d^6*f^4 + 368*a^7*b^3*c^3*d^15*f^4 + 368*a^3*b^7*c^15*d^3*f^4 - 368*a^2*b^8*c^6*d^12*f^4 + 304*a^5*b^5*c^15*d^3*f^4 + 304*a^5*b^5*c^3*d^15*f^4 - 144*a^6*b^4*c^2*d^16*f^4 - 144*a^4*b^6*c^16*d^2*f^4 - 108*a^8*b^2*c^2*d^16*f^4 - 108*a^2*b^8*c^16*d^2*f^4 + 80*a^7*b^3*c^15*d^3*f^4 + 80*a^3*b^7*c^3*d^15*f^4 - 60*a^8*b^2*c^14*d^4*f^4 - 60*a^6*b^4*c^16*d^2*f^4 - 60*a^4*b^6*c^2*d^16*f^4 - 60*a^2*b^8*c^4*d^14*f^4 - 80*b^10*c^12*d^6*f^4 - 60*b^10*c^14*d^4*f^4 - 60*b^10*c^10*d^8*f^4 - 24*b^10*c^16*d^2*f^4 - 24*b^10*c^8*d^10*f^4 - 4*b^10*c^6*d^12*f^4 - 80*a^10*c^6*d^12*f^4 - 60*a^10*c^8*d^10*f^4 - 60*a^10*c^4*d^14*f^4 - 24*a^10*c^10*d^8*f^4 - 24*a^10*c^2*d^16*f^4 - 4*a^10*c^12*d^6*f^4 - 8*a^8*b^2*d^18*f^4 - 4*a^6*b^4*d^18*f^4 - 8*a^2*b^8*c^18*f^4 - 4*a^4*b^6*c^18*f^4 - 4*b^10*c^18*f^4 - 4*a^10*d^18*f^4 - 12*A*C*a^7*b*c*d^11*f^2 - 12*A*C*a*b^7*c^11*d*f^2 - 912*B*C*a^4*b^4*c^5*d^7*f^2 + 792*B*C*a^5*b^3*c^4*d^8*f^2 - 792*B*C*a^3*b^5*c^8*d^4*f^2 + 720*B*C*a^4*b^4*c^7*d^5*f^2 - 480*B*C*a^6*b^2*c^5*d^7*f^2 - 408*B*C*a^2*b^6*c^5*d^7*f^2 + 384*B*C*a^2*b^6*c^7*d^5*f^2 - 336*B*C*a^5*b^3*c^8*d^4*f^2 + 324*B*C*a^3*b^5*c^4*d^8*f^2 + 312*B*C*a^6*b^2*c^7*d^5*f^2 - 248*B*C*a^6*b^2*c^3*d^9*f^2 + 216*B*C*a^2*b^6*c^9*d^3*f^2 - 196*B*C*a^4*b^4*c^3*d^9*f^2 + 132*B*C*a^4*b^4*c^9*d^3*f^2 + 80*B*C*a^3*b^5*c^6*d^6*f^2 - 64*B*C*a^5*b^3*c^6*d^6*f^2 - 36*B*C*a^3*b^5*c^2*d^10*f^2 - 28*B*C*a^2*b^6*c^3*d^9*f^2 + 12*B*C*a^5*b^3*c^10*d^2*f^2 - 12*B*C*a^5*b^3*c^2*d^10*f^2 - 12*B*C*a^3*b^5*c^10*d^2*f^2 - 4*B*C*a^6*b^2*c^9*d^3*f^2 - 1468*A*C*a^4*b^4*c^6*d^6*f^2 + 996*A*C*a^3*b^5*c^7*d^5*f^2 + 900*A*C*a^5*b^3*c^5*d^7*f^2 - 676*A*C*a^6*b^2*c^6*d^6*f^2 - 660*A*C*a^2*b^6*c^6*d^6*f^2 + 636*A*C*a^3*b^5*c^5*d^7*f^2

+ 540*A*C*a^5*b^3*c^7*d^5*f^2 - 236*A*C*a^5*b^3*c^3*d^9*f^2 - 204*A*C*a^3*b
^5*c^9*d^3*f^2 + 156*A*C*a^2*b^6*c^10*d^2*f^2 + 132*A*C*a^6*b^2*c^2*d^10*f^
2 - 72*A*C*a^6*b^2*c^4*d^8*f^2 - 72*A*C*a^5*b^3*c^9*d^3*f^2 + 66*A*C*a^2*b^
6*c^4*d^8*f^2 + 54*A*C*a^4*b^4*c^10*d^2*f^2 + 54*A*C*a^4*b^4*c^2*d^10*f^2 -
48*A*C*a^4*b^4*c^4*d^8*f^2 - 48*A*C*a^2*b^6*c^8*d^4*f^2 + 42*A*C*a^6*b^2*c
^8*d^4*f^2 - 40*A*C*a^3*b^5*c^3*d^9*f^2 - 36*A*C*a^4*b^4*c^8*d^4*f^2 + 24*A
*C*a^2*b^6*c^2*d^10*f^2 + 960*A*B*a^4*b^4*c^5*d^7*f^2 - 864*A*B*a^5*b^3*c^4
*d^8*f^2 + 756*A*B*a^3*b^5*c^8*d^4*f^2 - 744*A*B*a^4*b^4*c^7*d^5*f^2 - 528*
A*B*a^3*b^5*c^4*d^8*f^2 + 504*A*B*a^6*b^2*c^5*d^7*f^2 - 432*A*B*a^2*b^6*c^7
*d^5*f^2 + 432*A*B*a^2*b^6*c^5*d^7*f^2 + 348*A*B*a^5*b^3*c^8*d^4*f^2 - 312*
A*B*a^6*b^2*c^7*d^5*f^2 - 284*A*B*a^2*b^6*c^9*d^3*f^2 + 280*A*B*a^6*b^2*c^3
*d^9*f^2 + 264*A*B*a^4*b^4*c^3*d^9*f^2 - 240*A*B*a^3*b^5*c^6*d^6*f^2 - 172*
A*B*a^4*b^4*c^9*d^3*f^2 + 68*A*B*a^2*b^6*c^3*d^9*f^2 - 60*A*B*a^3*b^5*c^2*d
^10*f^2 + 24*A*B*a^5*b^3*c^6*d^6*f^2 - 24*A*B*a^5*b^3*c^2*d^10*f^2 + 12*A*B
*a^3*b^5*c^10*d^2*f^2 + 360*B*C*a^7*b*c^4*d^8*f^2 - 336*B*C*a*b^7*c^8*d^4*f
^2 + 168*B*C*a*b^7*c^6*d^6*f^2 - 136*B*C*a^7*b*c^6*d^6*f^2 + 36*B*C*a^6*b^2
*c*d^11*f^2 - 36*B*C*a^2*b^6*c^11*d*f^2 - 24*B*C*a^7*b*c^2*d^10*f^2 + 24*B*
C*a*b^7*c^10*d^2*f^2 - 12*B*C*a^4*b^4*c^11*d*f^2 + 12*B*C*a^4*b^4*c*d^11*f^
2 + 12*B*C*a*b^7*c^4*d^8*f^2 + 444*A*C*a*b^7*c^7*d^5*f^2 + 348*A*C*a^7*b*c^
5*d^7*f^2 - 164*A*C*a^7*b*c^3*d^9*f^2 - 132*A*C*a*b^7*c^9*d^3*f^2 + 84*A*C*
a*b^7*c^5*d^7*f^2 + 32*A*C*a*b^7*c^3*d^9*f^2 - 12*A*C*a^7*b*c^7*d^5*f^2 - 1
2*A*C*a^5*b^3*c*d^11*f^2 - 12*A*C*a^3*b^5*c^11*d*f^2 - 360*A*B*a^7*b*c^4*d^
8*f^2 + 288*A*B*a*b^7*c^8*d^4*f^2 - 288*A*B*a*b^7*c^6*d^6*f^2 - 144*A*B*a*b
^7*c^4*d^8*f^2 + 136*A*B*a^7*b*c^6*d^6*f^2 - 60*A*B*a*b^7*c^2*d^10*f^2 - 36
*A*B*a*b^7*c^10*d^2*f^2 + 24*A*B*a^7*b*c^2*d^10*f^2 - 24*A*B*a^6*b^2*c*d^11
*f^2 + 12*A*B*a^4*b^4*c*d^11*f^2 + 12*A*B*a^2*b^6*c^11*d*f^2 + 12*A*B*a^2*b
^6*c*d^11*f^2 + 80*B*C*b^8*c^9*d^3*f^2 - 24*B*C*b^8*c^7*d^5*f^2 - 90*A*C*b^
8*c^8*d^4*f^2 - 80*B*C*a^8*c^3*d^9*f^2 + 54*A*C*b^8*c^10*d^2*f^2 - 30*A*C*b
^8*c^6*d^6*f^2 + 24*B*C*a^8*c^5*d^7*f^2 - 12*A*C*b^8*c^4*d^8*f^2 - 112*A*B*
b^8*c^9*d^3*f^2 - 66*A*C*a^8*c^4*d^8*f^2 + 54*A*C*a^8*c^2*d^10*f^2 - 8*B*C*
a^5*b^3*d^12*f^2 - 8*B*C*a^3*b^5*d^12*f^2 + 4*A*B*b^8*c^3*d^9*f^2 + 2*A*C*a
^8*c^6*d^6*f^2 + 80*A*B*a^8*c^3*d^9*f^2 - 24*A*B*a^8*c^5*d^7*f^2 + 8*A*C*a^
2*b^6*d^12*f^2 - 4*B*C*a^3*b^5*c^12*f^2 + 4*A*C*a^4*b^4*d^12*f^2 - 2*A*C*a^
6*b^2*d^12*f^2 + 6*A*C*a^2*b^6*c^12*f^2 + 4*A*B*a^5*b^3*d^12*f^2 - 4*A*B*a^
3*b^5*d^12*f^2 + 726*C^2*a^4*b^4*c^6*d^6*f^2 - 402*C^2*a^5*b^3*c^5*d^7*f^2
- 402*C^2*a^3*b^5*c^7*d^5*f^2 + 322*C^2*a^6*b^2*c^6*d^6*f^2 + 322*C^2*a^2*b
^6*c^6*d^6*f^2 - 222*C^2*a^5*b^3*c^7*d^5*f^2 - 222*C^2*a^3*b^5*c^5*d^7*f^2
+ 134*C^2*a^5*b^3*c^3*d^9*f^2 + 134*C^2*a^3*b^5*c^9*d^3*f^2 - 66*C^2*a^6*b^
2*c^2*d^10*f^2 - 66*C^2*a^2*b^6*c^10*d^2*f^2 + 52*C^2*a^5*b^3*c^9*d^3*f^2 +
52*C^2*a^3*b^5*c^3*d^9*f^2 - 27*C^2*a^6*b^2*c^8*d^4*f^2 - 27*C^2*a^2*b^6*c
^4*d^8*f^2 + 24*C^2*a^6*b^2*c^4*d^8*f^2 + 24*C^2*a^4*b^4*c^8*d^4*f^2 + 24*C
^2*a^4*b^4*c^4*d^8*f^2 + 24*C^2*a^2*b^6*c^8*d^4*f^2 - 15*C^2*a^4*b^4*c^10*d
^2*f^2 - 15*C^2*a^4*b^4*c^2*d^10*f^2 - 570*B^2*a^4*b^4*c^6*d^6*f^2 + 366*B^
2*a^3*b^5*c^7*d^5*f^2 + 318*B^2*a^5*b^3*c^5*d^7*f^2 - 262*B^2*a^6*b^2*c^6*d
^6*f^2 - 222*B^2*a^2*b^6*c^6*d^6*f^2 - 210*B^2*a^5*b^3*c^3*d^9*f^2 + 186*B^

2*a^5*b^3*c^7*d^5*f^2 + 162*B^2*a^3*b^5*c^5*d^7*f^2 - 142*B^2*a^3*b^5*c^9*d
^3*f^2 + 132*B^2*a^4*b^4*c^4*d^8*f^2 + 117*B^2*a^2*b^6*c^4*d^8*f^2 + 102*B^
2*a^6*b^2*c^2*d^10*f^2 - 96*B^2*a^3*b^5*c^3*d^9*f^2 + 90*B^2*a^2*b^6*c^10*d
^2*f^2 + 81*B^2*a^4*b^4*c^2*d^10*f^2 - 56*B^2*a^5*b^3*c^9*d^3*f^2 + 48*B^2*
a^6*b^2*c^4*d^8*f^2 + 48*B^2*a^4*b^4*c^8*d^4*f^2 + 45*B^2*a^6*b^2*c^8*d^4*f
^2 + 36*B^2*a^2*b^6*c^8*d^4*f^2 + 36*B^2*a^2*b^6*c^2*d^10*f^2 + 33*B^2*a^4*
b^4*c^10*d^2*f^2 + 822*A^2*a^4*b^4*c^6*d^6*f^2 - 594*A^2*a^3*b^5*c^7*d^5*f^
2 - 498*A^2*a^5*b^3*c^5*d^7*f^2 + 498*A^2*a^2*b^6*c^6*d^6*f^2 - 414*A^2*a^3
*b^5*c^5*d^7*f^2 + 354*A^2*a^6*b^2*c^6*d^6*f^2 - 318*A^2*a^5*b^3*c^7*d^5*f^
2 + 144*A^2*a^2*b^6*c^8*d^4*f^2 + 102*A^2*a^5*b^3*c^3*d^9*f^2 + 84*A^2*a^4*
b^4*c^4*d^8*f^2 + 81*A^2*a^2*b^6*c^4*d^8*f^2 + 72*A^2*a^4*b^4*c^8*d^4*f^2 +
70*A^2*a^3*b^5*c^9*d^3*f^2 - 66*A^2*a^6*b^2*c^2*d^10*f^2 + 48*A^2*a^6*b^2*
c^4*d^8*f^2 - 42*A^2*a^2*b^6*c^10*d^2*f^2 + 24*A^2*a^2*b^6*c^2*d^10*f^2 + 2
0*A^2*a^5*b^3*c^9*d^3*f^2 - 15*A^2*a^6*b^2*c^8*d^4*f^2 - 15*A^2*a^4*b^4*c^1
0*d^2*f^2 - 15*A^2*a^4*b^4*c^2*d^10*f^2 - 12*A^2*a^3*b^5*c^3*d^9*f^2 - 24*B
*C*b^8*c^11*d*f^2 + 24*B*C*a^8*c*d^11*f^2 + 12*A*B*b^8*c^11*d*f^2 - 8*B*C*a
^7*b*d^12*f^2 - 24*A*B*a^8*c*d^11*f^2 + 4*B*C*a*b^7*c^12*f^2 + 8*A*B*a^7*b*
d^12*f^2 - 8*A*B*a*b^7*d^12*f^2 - 8*A*B*a*b^7*c^12*f^2 - 174*C^2*a^7*b*c^5*
d^7*f^2 - 174*C^2*a*b^7*c^7*d^5*f^2 + 82*C^2*a^7*b*c^3*d^9*f^2 + 82*C^2*a*b
^7*c^9*d^3*f^2 + 6*C^2*a^7*b*c^7*d^5*f^2 + 6*C^2*a^5*b^3*c*d^11*f^2 + 6*C^2
*a^3*b^5*c^11*d*f^2 + 6*C^2*a*b^7*c^5*d^7*f^2 + 162*B^2*a*b^7*c^7*d^5*f^2 +
138*B^2*a^7*b*c^5*d^7*f^2 - 118*B^2*a^7*b*c^3*d^9*f^2 - 86*B^2*a*b^7*c^9*d
^3*f^2 - 30*B^2*a^5*b^3*c*d^11*f^2 - 18*B^2*a^7*b*c^7*d^5*f^2 - 18*B^2*a*b^
7*c^5*d^7*f^2 - 12*B^2*a^3*b^5*c*d^11*f^2 - 6*B^2*a^3*b^5*c^11*d*f^2 - 4*B^
2*a*b^7*c^3*d^9*f^2 - 270*A^2*a*b^7*c^7*d^5*f^2 - 174*A^2*a^7*b*c^5*d^7*f^2
- 90*A^2*a*b^7*c^5*d^7*f^2 + 82*A^2*a^7*b*c^3*d^9*f^2 + 50*A^2*a*b^7*c^9*d
^3*f^2 - 32*A^2*a*b^7*c^3*d^9*f^2 + 6*A^2*a^7*b*c^7*d^5*f^2 + 6*A^2*a^5*b^3
*c*d^11*f^2 + 6*A^2*a^3*b^5*c^11*d*f^2 + 6*C^2*a^7*b*c*d^11*f^2 + 6*C^2*a*b
^7*c^11*d*f^2 - 18*B^2*a^7*b*c*d^11*f^2 - 6*B^2*a*b^7*c^11*d*f^2 + 6*A^2*a^
7*b*c*d^11*f^2 + 6*A^2*a*b^7*c^11*d*f^2 - 6*A*C*a^8*d^12*f^2 - 2*A*C*b^8*c^
12*f^2 + 33*C^2*b^8*c^8*d^4*f^2 - 27*C^2*b^8*c^10*d^2*f^2 - C^2*b^8*c^6*d^6
*f^2 + 33*C^2*a^8*c^4*d^8*f^2 + 33*B^2*b^8*c^10*d^2*f^2 - 27*C^2*a^8*c^2*d^
10*f^2 - 27*B^2*b^8*c^8*d^4*f^2 + 3*B^2*b^8*c^6*d^6*f^2 - C^2*a^8*c^6*d^6*f
^2 + 117*A^2*b^8*c^8*d^4*f^2 + 111*A^2*b^8*c^6*d^6*f^2 + 72*A^2*b^8*c^4*d^8
*f^2 + 33*B^2*a^8*c^2*d^10*f^2 - 27*B^2*a^8*c^4*d^8*f^2 + 24*A^2*b^8*c^2*d^
10*f^2 + 4*C^2*a^4*b^4*d^12*f^2 + 3*C^2*a^6*b^2*d^12*f^2 + 3*B^2*a^8*c^6*d^
6*f^2 - 3*A^2*b^8*c^10*d^2*f^2 + 33*A^2*a^8*c^4*d^8*f^2 - 27*A^2*a^8*c^2*d^
10*f^2 + 4*C^2*a^4*b^4*c^12*f^2 + 4*B^2*a^4*b^4*d^12*f^2 + 4*B^2*a^2*b^6*d^
12*f^2 + 3*C^2*a^2*b^6*c^12*f^2 + 3*B^2*a^6*b^2*d^12*f^2 - A^2*a^8*c^6*d^6*
f^2 - 4*A^2*a^4*b^4*d^12*f^2 + 3*B^2*a^2*b^6*c^12*f^2 - A^2*a^6*b^2*d^12*f^
2 - A^2*a^2*b^6*c^12*f^2 + 3*C^2*b^8*c^12*f^2 + 3*C^2*a^8*d^12*f^2 + 4*A^2*
b^8*d^12*f^2 - B^2*b^8*c^12*f^2 - B^2*a^8*d^12*f^2 + 3*A^2*b^8*c^12*f^2 + 3
*A^2*a^8*d^12*f^2 - 24*A*B*C*a*b^6*c*d^8*f + 342*A*B*C*a^2*b^5*c^4*d^5*f -
186*A*B*C*a^3*b^4*c^5*d^4*f - 66*A*B*C*a^4*b^3*c^2*d^7*f + 48*A*B*C*a^2*b^5
*c^2*d^7*f + 42*A*B*C*a^2*b^5*c^6*d^3*f + 26*A*B*C*a^5*b^2*c^3*d^6*f + 24*A

*B*C*a^4*b^3*c^6*d^3*f - 18*A*B*C*a^4*b^3*c^4*d^5*f - 18*A*B*C*a^3*b^4*c^7*
d^2*f - 8*A*B*C*a^3*b^4*c^3*d^6*f + 6*A*B*C*a^5*b^2*c^5*d^4*f - 128*A*B*C*a
*b^6*c^3*d^6*f + 126*A*B*C*a*b^6*c^7*d^2*f + 72*A*B*C*a^3*b^4*c*d^8*f - 36*
A*B*C*a^5*b^2*c*d^8*f - 36*A*B*C*a^2*b^5*c^8*d*f + 30*A*B*C*a^6*b*c^2*d^7*f
 - 12*A*B*C*a^6*b*c^4*d^5*f - 12*A*B*C*a*b^6*c^5*d^4*f - 21*B^2*C*a*b^6*c^8
*d*f - 3*B^2*C*a^6*b*c*d^8*f + 21*A^2*C*a*b^6*c^8*d*f - 21*A*C^2*a*b^6*c^8*
d*f - 9*A^2*C*a^6*b*c*d^8*f + 9*A*C^2*a^6*b*c*d^8*f + 36*A^2*B*a*b^6*c*d^8*
f + 21*A*B^2*a*b^6*c^8*d*f + 3*A*B^2*a^6*b*c*d^8*f - 78*A*B*C*b^7*c^6*d^3*f
 + 24*A*B*C*b^7*c^4*d^5*f + 2*A*B*C*a^7*c^3*d^6*f + 16*A*B*C*a^4*b^3*d^9*f
- 16*A*B*C*a^2*b^5*d^9*f - 237*B^2*C*a^3*b^4*c^4*d^5*f + 165*B*C^2*a^3*b^4*
c^5*d^4*f + 92*B^2*C*a^2*b^5*c^3*d^6*f - 81*B^2*C*a^2*b^5*c^7*d^2*f + 77*B^
2*C*a^4*b^3*c^3*d^6*f - 75*B*C^2*a^2*b^5*c^4*d^5*f + 69*B^2*C*a^4*b^3*c^5*d
^4*f + 69*B*C^2*a^4*b^3*c^4*d^5*f - 68*B*C^2*a^3*b^4*c^3*d^6*f - 63*B^2*C*a
^5*b^2*c^4*d^5*f - 61*B*C^2*a^2*b^5*c^6*d^3*f + 57*B*C^2*a^4*b^3*c^2*d^7*f
- 53*B*C^2*a^5*b^2*c^3*d^6*f - 44*B*C^2*a^4*b^3*c^6*d^3*f - 36*B^2*C*a^3*b^
4*c^2*d^7*f + 35*B^2*C*a^3*b^4*c^6*d^3*f + 33*B^2*C*a^5*b^2*c^2*d^7*f - 33*
B^2*C*a^2*b^5*c^5*d^4*f + 33*B*C^2*a^3*b^4*c^7*d^2*f - 12*B^2*C*a^4*b^3*c^7
*d^2*f + 9*B*C^2*a^5*b^2*c^5*d^4*f + 4*B^2*C*a^5*b^2*c^6*d^3*f + 225*A^2*C*
a^2*b^5*c^5*d^4*f - 105*A*C^2*a^2*b^5*c^5*d^4*f - 99*A^2*C*a^3*b^4*c^4*d^5*
f - 81*A^2*C*a^5*b^2*c^4*d^5*f + 67*A^2*C*a^4*b^3*c^3*d^6*f - 59*A*C^2*a^4*
b^3*c^3*d^6*f + 57*A*C^2*a^5*b^2*c^2*d^7*f - 57*A*C^2*a^2*b^5*c^7*d^2*f + 5
1*A^2*C*a^4*b^3*c^5*d^4*f + 48*A^2*C*a^3*b^4*c^2*d^7*f + 45*A*C^2*a^5*b^2*c
^4*d^5*f - 35*A^2*C*a^3*b^4*c^6*d^3*f - 33*A^2*C*a^5*b^2*c^2*d^7*f + 33*A^2
*C*a^2*b^5*c^7*d^2*f + 33*A*C^2*a^4*b^3*c^5*d^4*f + 27*A*C^2*a^3*b^4*c^6*d^
3*f - 24*A*C^2*a^3*b^4*c^2*d^7*f + 24*A*C^2*a^2*b^5*c^3*d^6*f - 21*A*C^2*a^
3*b^4*c^4*d^5*f - 16*A^2*C*a^2*b^5*c^3*d^6*f - 243*A^2*B*a^2*b^5*c^4*d^5*f
- 156*A*B^2*a^2*b^5*c^3*d^6*f + 141*A*B^2*a^3*b^4*c^4*d^5*f + 108*A^2*B*a^3
*b^4*c^3*d^6*f - 105*A*B^2*a^4*b^3*c^3*d^6*f + 84*A*B^2*a^3*b^4*c^2*d^7*f +
 81*A*B^2*a^2*b^5*c^5*d^4*f - 51*A^2*B*a^4*b^3*c^4*d^5*f + 51*A^2*B*a^2*b^5
*c^6*d^3*f - 48*A^2*B*a^2*b^5*c^2*d^7*f + 45*A^2*B*a^3*b^4*c^5*d^4*f + 39*A
*B^2*a^5*b^2*c^4*d^5*f - 35*A*B^2*a^3*b^4*c^6*d^3*f + 33*A*B^2*a^2*b^5*c^7*
d^2*f + 27*A^2*B*a^5*b^2*c^3*d^6*f - 21*A*B^2*a^4*b^3*c^5*d^4*f + 20*A^2*B*
a^4*b^3*c^6*d^3*f - 15*A^2*B*a^5*b^2*c^5*d^4*f - 15*A^2*B*a^3*b^4*c^7*d^2*f
 + 9*A^2*B*a^4*b^3*c^2*d^7*f + 3*A*B^2*a^5*b^2*c^2*d^7*f + 18*A*B*C*b^7*c^8
*d*f - 6*A*B*C*a^7*c*d^8*f + 2*A*B*C*a^6*b*d^9*f - 6*A*B*C*a*b^6*c^9*f + 63
*B^2*C*a*b^6*c^6*d^3*f - 48*B^2*C*a^4*b^3*c*d^8*f + 42*B*C^2*a^2*b^5*c^8*d*
f + 42*B*C^2*a*b^6*c^5*d^4*f - 39*B*C^2*a*b^6*c^7*d^2*f + 30*B*C^2*a^5*b^2*
c*d^8*f - 24*B^2*C*a*b^6*c^4*d^5*f - 24*B*C^2*a^3*b^4*c*d^8*f + 17*B^2*C*a^
6*b*c^3*d^6*f - 15*B*C^2*a^6*b*c^2*d^7*f + 12*B^2*C*a^3*b^4*c^8*d*f + 12*B^
2*C*a^2*b^5*c*d^8*f + 6*B*C^2*a^6*b*c^4*d^5*f - 192*A^2*C*a*b^6*c^4*d^5*f -
 99*A^2*C*a*b^6*c^6*d^3*f + 84*A*C^2*a*b^6*c^4*d^5*f + 59*A*C^2*a*b^6*c^6*d
^3*f + 51*A^2*C*a^6*b*c^3*d^6*f - 51*A*C^2*a^6*b*c^3*d^6*f - 36*A^2*C*a^2*b
^5*c*d^8*f - 24*A*C^2*a^4*b^3*c*d^8*f + 24*A*C^2*a^2*b^5*c*d^8*f + 12*A^2*C
*a^4*b^3*c*d^8*f + 12*A*C^2*a^3*b^4*c^8*d*f + 160*A^2*B*a*b^6*c^3*d^6*f - 9
9*A*B^2*a*b^6*c^6*d^3*f - 87*A^2*B*a*b^6*c^7*d^2*f - 72*A*B^2*a*b^6*c^4*d^5

*f - 48*A*B^2*a^2*b^5*c*d^8*f - 36*A^2*B*a^3*b^4*c*d^8*f + 24*A*B^2*a^4*b^3
*c*d^8*f - 17*A*B^2*a^6*b*c^3*d^6*f - 15*A^2*B*a^6*b*c^2*d^7*f + 12*A*B^2*a
*b^6*c^2*d^7*f + 6*A^2*B*a^6*b*c^4*d^5*f + 6*A^2*B*a^5*b^2*c*d^8*f + 6*A^2*
B*a^2*b^5*c^8*d*f - 6*A^2*B*a*b^6*c^5*d^4*f + 3*B^2*C*b^7*c^7*d^2*f - B*C^2
*b^7*c^6*d^3*f + 96*A^2*C*b^7*c^5*d^4*f - 39*A^2*C*b^7*c^7*d^2*f - 36*A*C^2
*b^7*c^5*d^4*f + 32*A^2*C*b^7*c^3*d^6*f + 15*A*C^2*b^7*c^7*d^2*f - 3*B^2*C*
a^7*c^2*d^7*f - B*C^2*a^7*c^3*d^6*f + 111*A^2*B*b^7*c^6*d^3*f - 39*A*B^2*b^
7*c^7*d^2*f + 24*A*B^2*b^7*c^5*d^4*f + 12*B^2*C*a^3*b^4*d^9*f - 12*B*C^2*a^
4*b^3*d^9*f - 9*A^2*C*a^7*c^2*d^7*f + 9*A*C^2*a^7*c^2*d^7*f - 4*A*B^2*b^7*c
^3*d^6*f - 12*A^2*C*a^3*b^4*d^9*f - 8*A*C^2*a^5*b^2*d^9*f + 8*A*C^2*a^3*b^4
*d^9*f + 4*B^2*C*a^2*b^5*c^9*f + 4*A^2*C*a^5*b^2*d^9*f - 4*B*C^2*a^3*b^4*c^
9*f + 3*A*B^2*a^7*c^2*d^7*f - A^2*B*a^7*c^3*d^6*f + 12*A^2*B*a^2*b^5*d^9*f
- 8*A*B^2*a^3*b^4*d^9*f - 4*A^2*B*a^4*b^3*d^9*f + 4*A*C^2*a^2*b^5*c^9*f - 3
*C^3*a^6*b*c*d^8*f + 3*C^3*a*b^6*c^8*d*f + 3*A^3*a^6*b*c*d^8*f - 3*A^3*a*b^
6*c^8*d*f + 3*B*C^2*b^7*c^8*d*f + 12*A^2*C*b^7*c*d^8*f + 3*B*C^2*a^7*c*d^8*
f - 9*A^2*B*b^7*c^8*d*f - B*C^2*a^6*b*d^9*f + 4*A^2*C*a*b^6*d^9*f + 3*A^2*B
*a^7*c*d^8*f + 3*B*C^2*a*b^6*c^9*f + 8*A*B^2*a*b^6*d^9*f - A^2*B*a^6*b*d^9*
f - A^2*B*a*b^6*c^9*f - 39*C^3*a^4*b^3*c^5*d^4*f + 39*C^3*a^3*b^4*c^4*d^5*f
 - 27*C^3*a^5*b^2*c^2*d^7*f + 27*C^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^4*b^3*c^3
*d^6*f - 17*C^3*a^3*b^4*c^6*d^3*f - 3*C^3*a^5*b^2*c^4*d^5*f + 3*C^3*a^2*b^5
*c^5*d^4*f - 63*B^3*a^3*b^4*c^5*d^4*f + 57*B^3*a^2*b^5*c^4*d^5*f - 51*B^3*a
^4*b^3*c^2*d^7*f + 48*B^3*a^3*b^4*c^3*d^6*f + 31*B^3*a^2*b^5*c^6*d^3*f + 27
*B^3*a^5*b^2*c^3*d^6*f + 16*B^3*a^4*b^3*c^6*d^3*f - 15*B^3*a^5*b^2*c^5*d^4*
f - 12*B^3*a^2*b^5*c^2*d^7*f + 9*B^3*a^4*b^3*c^4*d^5*f - 3*B^3*a^3*b^4*c^7*
d^2*f - 123*A^3*a^2*b^5*c^5*d^4*f + 81*A^3*a^3*b^4*c^4*d^5*f - 45*A^3*a^4*b
^3*c^5*d^4*f + 39*A^3*a^5*b^2*c^4*d^5*f - 25*A^3*a^4*b^3*c^3*d^6*f + 25*A^3
*a^3*b^4*c^6*d^3*f - 24*A^3*a^3*b^4*c^2*d^7*f - 8*A^3*a^2*b^5*c^3*d^6*f + 3
*A^3*a^5*b^2*c^2*d^7*f - 3*A^3*a^2*b^5*c^7*d^2*f + 17*C^3*a^6*b*c^3*d^6*f -
 17*C^3*a*b^6*c^6*d^3*f + 12*C^3*a^4*b^3*c*d^8*f - 12*C^3*a^3*b^4*c^8*d*f +
 24*B^3*a^3*b^4*c*d^8*f + 21*B^3*a*b^6*c^7*d^2*f - 18*B^3*a*b^6*c^5*d^4*f -
 15*B^3*a^6*b*c^2*d^7*f + 6*B^3*a^6*b*c^4*d^5*f + 6*B^3*a^5*b^2*c*d^8*f - 6
*B^3*a^2*b^5*c^8*d*f + 4*B^3*a*b^6*c^3*d^6*f + 108*A^3*a*b^6*c^4*d^5*f + 57
*A^3*a*b^6*c^6*d^3*f - 17*A^3*a^6*b*c^3*d^6*f + 12*A^3*a^2*b^5*c*d^8*f + 3*
C^3*b^7*c^7*d^2*f - 3*C^3*a^7*c^2*d^7*f - B^3*b^7*c^6*d^3*f - 60*A^3*b^7*c^
5*d^4*f - 32*A^3*b^7*c^3*d^6*f + 21*A^3*b^7*c^7*d^2*f + 4*C^3*a^5*b^2*d^9*f
 - B^3*a^7*c^3*d^6*f - 4*C^3*a^2*b^5*c^9*f - 4*B^3*a^2*b^5*d^9*f + 3*A^3*a^
7*c^2*d^7*f + 4*A^3*a^3*b^4*d^9*f + 3*B^3*b^7*c^8*d*f - 12*A^3*b^7*c*d^8*f
+ 3*B^3*a^7*c*d^8*f - B^3*a^6*b*d^9*f - 4*A^3*a*b^6*d^9*f - B^3*a*b^6*c^9*f
 - B^2*C*b^7*c^9*f - 4*A^2*B*b^7*d^9*f + 3*A^2*C*a^7*d^9*f - 3*A*C^2*a^7*d^
9*f - A*C^2*b^7*c^9*f - A*B^2*a^7*d^9*f - C^3*b^7*c^9*f - A^3*a^7*d^9*f + B
^2*C*a^7*d^9*f + A^2*C*b^7*c^9*f + A*B^2*b^7*c^9*f + C^3*a^7*d^9*f + A^3*b^
7*c^9*f - 6*A*B^2*C*a*b^5*c^5*d - 21*A^2*B*C*a^2*b^4*c^3*d^3 + 21*A*B*C^2*a
^2*b^4*c^3*d^3 + 12*A*B^2*C*a^2*b^4*c^4*d^2 - 12*A*B^2*C*a^2*b^4*c^2*d^4 -
10*A*B^2*C*a^3*b^3*c^3*d^3 - 6*A*B*C^2*a^3*b^3*c^4*d^2 + 3*A^2*B*C*a^3*b^3*
c^4*d^2 + 3*A^2*B*C*a^3*b^3*c^2*d^4 + 3*A*B^2*C*a^4*b^2*c^2*d^4 + 3*A*B*C^2

```
*a^3*b^3*c^2*d^4 + 2*A*B*C^2*a^4*b^2*c^3*d^3 - A^2*B*C*a^4*b^2*c^3*d^3 + 18
*A^2*B*C*a*b^5*c^2*d^4 + 10*A*B^2*C*a*b^5*c^3*d^3 + 9*A^2*B*C*a*b^5*c^4*d^2
 - 9*A*B*C^2*a*b^5*c^4*d^2 - 9*A*B*C^2*a*b^5*c^2*d^4 - 6*A^2*B*C*a^2*b^4*c*
d^5 + 6*A*B^2*C*a^3*b^3*c*d^5 - 6*A*B*C^2*a^4*b^2*c*d^5 + 6*A*B*C^2*a^2*b^4
*c^5*d + 3*A^2*B*C*a^4*b^2*c*d^5 - 3*A^2*B*C*a^2*b^4*c^5*d + 3*A*B*C^2*a^2*
b^4*c*d^5 + 3*B^3*C*a^4*b^2*c*d^5 - 3*B^3*C*a^2*b^4*c^5*d + 3*B^3*C*a*b^5*c
^4*d^2 + 3*B^2*C^2*a*b^5*c^5*d + 3*B*C^3*a^4*b^2*c*d^5 - 3*B*C^3*a^2*b^4*c^
5*d + 3*B*C^3*a*b^5*c^4*d^2 + 24*A^3*C*a*b^5*c^3*d^3 + 8*A*C^3*a*b^5*c^3*d^
3 - 9*A^3*B*a*b^5*c^2*d^4 - 9*A*B^3*a*b^5*c^2*d^4 + 3*A^3*B*a^2*b^4*c*d^5 -
 3*A^3*B*a*b^5*c^4*d^2 + 3*A^2*B^2*a*b^5*c^5*d + 3*A*B^3*a^2*b^4*c*d^5 - 3*
A*B^3*a*b^5*c^4*d^2 - 3*A*B^2*C*b^6*c^4*d^2 - 2*A^2*B*C*b^6*c^3*d^3 + 5*A*B
*C^2*a^3*b^3*d^6 - 4*A^2*B*C*a^3*b^3*d^6 - A*B^2*C*a^4*b^2*d^6 + 9*B^2*C^2*
a^3*b^3*c^3*d^3 - 6*B^2*C^2*a^2*b^4*c^4*d^2 + 6*B^2*C^2*a^2*b^4*c^2*d^4 - 3
*B^2*C^2*a^4*b^2*c^2*d^4 + 24*A^2*C^2*a^3*b^3*c^3*d^3 - 15*A^2*C^2*a^2*b^4*
c^4*d^2 - 9*A^2*C^2*a^4*b^2*c^2*d^4 + 3*A^2*C^2*a^2*b^4*c^2*d^4 + 9*A^2*B^2
*a^2*b^4*c^2*d^4 - 3*A^2*B^2*a^2*b^4*c^4*d^2 + 6*A^2*B*C*b^6*c^5*d - 3*A*B*
C^2*b^6*c^5*d + 4*A^2*B*C*a*b^5*d^6 - 2*A*B*C^2*a*b^5*d^6 + 2*A*B*C^2*a*b^5
*c^6 - A^2*B*C*a*b^5*c^6 - 7*B^3*C*a^2*b^4*c^3*d^3 - 7*B*C^3*a^2*b^4*c^3*d^
3 + 3*B^3*C*a^3*b^3*c^4*d^2 - 3*B^3*C*a^3*b^3*c^2*d^4 - 3*B^2*C^2*a^3*b^3*c
*d^5 + 3*B*C^3*a^3*b^3*c^4*d^2 - 3*B*C^3*a^3*b^3*c^2*d^4 - B^3*C*a^4*b^2*c^
3*d^3 - B^2*C^2*a*b^5*c^3*d^3 - B*C^3*a^4*b^2*c^3*d^3 - 24*A^2*C^2*a*b^5*c^
3*d^3 - 24*A*C^3*a^3*b^3*c^3*d^3 + 12*A*C^3*a^2*b^4*c^4*d^2 + 9*A*C^3*a^4*b
^2*c^2*d^4 - 8*A^3*C*a^3*b^3*c^3*d^3 + 6*A^3*C*a^2*b^4*c^4*d^2 - 6*A^3*C*a^
2*b^4*c^2*d^4 + 3*A^3*C*a^4*b^2*c^2*d^4 - 9*A^2*B^2*a*b^5*c^3*d^3 + 7*A^3*B
*a^2*b^4*c^3*d^3 + 7*A*B^3*a^2*b^4*c^3*d^3 - 3*A^3*B*a^3*b^3*c^2*d^4 - 3*A^
2*B^2*a^3*b^3*c*d^5 - 3*A*B^3*a^3*b^3*c^2*d^4 + 12*A^2*C^2*b^6*c^4*d^2 + 3*
A^2*C^2*b^6*c^2*d^4 + 6*A^2*B^2*b^6*c^4*d^2 + 3*A^2*B^2*b^6*c^2*d^4 - 5*A^2
*C^2*a^2*b^4*d^6 + 3*A^2*C^2*a^4*b^2*d^6 + A*B*C^2*b^6*c^3*d^3 - 3*B^4*a^3*
b^3*c*d^5 - B^4*a*b^5*c^3*d^3 + A^2*B^2*a^3*b^3*c^3*d^3 - 8*A^4*a*b^5*c^3*d
^3 - 15*A^3*C*b^6*c^4*d^2 - 6*A^3*C*b^6*c^2*d^4 - 3*A*C^3*b^6*c^4*d^2 - 2*B
^3*C*a^3*b^3*d^6 - 2*B*C^3*a^3*b^3*d^6 + 4*A^3*C*a^2*b^4*d^6 - 3*A*C^3*a^4*
b^2*d^6 + 2*A*C^3*a^2*b^4*d^6 - A^3*C*a^4*b^2*d^6 - 2*A*C^3*a^2*b^4*c^6 + 3
*B^4*a*b^5*c^5*d - 3*A^3*B*b^6*c^5*d - 3*A*B^3*b^6*c^5*d - B^3*C*a*b^5*c^6
 - B*C^3*a*b^5*c^6 - 2*A^3*B*a*b^5*d^6 - 2*A*B^3*a*b^5*d^6 + 8*C^4*a^3*b^3*c
^3*d^3 - 3*C^4*a^4*b^2*c^2*d^4 - 3*C^4*a^2*b^4*c^4*d^2 + 6*B^4*a^2*b^4*c^2*
d^4 - 3*B^4*a^2*b^4*c^4*d^2 + 3*A^4*a^2*b^4*c^2*d^4 + B^2*C^2*a^4*b^2*d^6 +
 B^2*C^2*a^2*b^4*d^6 + B^2*C^2*a^2*b^4*c^6 + A^2*C^2*a^2*b^4*c^6 - 2*A^3*C*
b^6*d^6 + A^3*B*b^6*c^3*d^3 + A*B^3*b^6*c^3*d^3 + A^3*B*a^3*b^3*d^6 + A*B^3
*a^3*b^3*d^6 + 6*A^4*b^6*c^4*d^2 + 3*A^4*b^6*c^2*d^4 - A^4*a^2*b^4*d^6 - 2*
A^2*C^2*b^6*c^6 + A*B^2*C*b^6*c^6 + B^4*a^3*b^3*c^3*d^3 + A^3*C*b^6*c^6 + A
*C^3*b^6*c^6 + C^4*a^4*b^2*d^6 + C^4*a^2*b^4*c^6 + B^4*a^2*b^4*d^6 + A^2*C^
2*b^6*d^6 + A^2*B^2*b^6*d^6 + A^4*b^6*d^6, f, k), k, 1, 4) - ((A*a*d^5 - 3*
C*b*c^5 - 3*A*b*c*d^4 + B*a*c*d^4 + 5*B*b*c^4*d + C*a*c^4*d + 5*A*a*c^2*d^3
 - 7*A*b*c^3*d^2 - 3*B*a*c^3*d^2 + B*b*c^2*d^3 - 3*C*a*c^2*d^3 + C*b*c^3*d^
2)/(2*(a^2*d^2 + b^2*c^2 - 2*a*b*c*d)*(c^4 + d^4 + 2*c^2*d^2)) - (tan(e + f
```

```
*x)*(A*b*d^5 - B*a*d^5 - 2*A*a*c*d^4 + 2*C*a*c*d^4 + C*b*c^4*d + 3*A*b*c^2*
d^3 + B*a*c^2*d^3 - 2*B*b*c^3*d^2 - C*b*c^2*d^3))/((a^2*d^2 + b^2*c^2 - 2*a
*b*c*d)*(c^4 + d^4 + 2*c^2*d^2)))/(c^2 + d^2*tan(e + f*x)^2 + 2*c*d*tan(e +
 f*x)))/f
```

sympy **[F(-2)]** time = 0.00, size = 0, normalized size = 0.00

Exception raised: NotImplementedError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)
)**3,x)
```

```
[Out] Exception raised: NotImplementedError
```

3.89 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}\,dx$

Optimal. Leaf size=861

$$\frac{\left(-3Cda^4+4bBda^3-b^2(Bc+(5A+C)d)a^2+2b^3(Ac-Cc+Bd)a+b^4(Bc-3Ad)\right)\log(a\cos(e+fx)+b\sin(e}{\left(a^2+b^2\right)^2(bc-ad)^4f}$$

```
[Out] -(b^2*(A*c^3-3*A*c*d^2+3*B*c^2*d-B*d^3-C*c^3+3*C*c*d^2)+a^2*(c^3*C-3*B*c^2*
d-3*c*C*d^2+B*d^3-A*(c^3-3*c*d^2))+2*a*b*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^
2)))*x/(a^2+b^2)^2/(c^2+d^2)^3+b^2*(4*a^3*b*B*d-3*a^4*C*d+b^4*(-3*A*d+B*c)+
2*a*b^3*(A*c+B*d-C*c)-a^2*b^2*(B*c+(5*A+C)*d))*ln(a*cos(f*x+e)+b*sin(f*x+e)
)/(a^2+b^2)^2/(-a*d+b*c)^4/f+d*(b^2*(3*c^6*C-6*B*c^5*d+c^4*(10*A-C)*d^2-3*B
*c^3*d^3+9*A*c^2*d^4-B*c*d^5+3*A*d^6)+a^2*d^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3
*c*d^2))-2*a*b*d^2*(c*(A-C)*d*(5*c^2+d^2)-B*(2*c^4-3*c^2*d^2-d^4)))*ln(c*co
s(f*x+e)+d*sin(f*x+e))/(-a*d+b*c)^4/(c^2+d^2)^3/f-1/2*d*(b^2*c*(-B*d+C*c)-2
*a*b*B*(c^2+d^2)+a^2*(-B*c*d+3*C*c^2+2*C*d^2)+A*(a^2*d^2+b^2*(2*c^2+3*d^2))
)/(a^2+b^2)/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^2+(-A*b^2+a*(B*b-C*a)
)/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))^2-d*(b^3*c*(-3*B
*c^2*d-B*d^3+2*C*c^3)+a^2*b*(-3*B*c^3*d-B*c*d^3+3*C*c^4+2*C*c^2*d^2+C*d^4)+
a^3*d^2*(2*c*C*d+B*(c^2-d^2))+a*b^2*(2*c*C*d^3-B*(c^4+c^2*d^2+2*d^4))-A*(2*
a^3*c*d^3+2*a*b^2*c*d^3-2*a^2*b*d^2*(2*c^2+d^2)-b^3*(c^4+6*c^2*d^2+3*d^4)))
/(a^2+b^2)/(-a*d+b*c)^3/(c^2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] time = 4.28, antiderivative size = 860, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 3, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.067, Rules used = {3649, 3651, 3530}

$$\frac{\left(-3Cda^4+4bBda^3-b^2(Bc+(5A+C)d)a^2+2b^3(Ac-Cc+Bd)a+b^4(Bc-3Ad)\right)\log(a\cos(e+fx)+b\sin(e}{\left(a^2+b^2\right)^2(bc-ad)^4f}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(c + d*
Tan[e + f*x])^3),x]
```

```
[Out] -(((b^2*(A*c^3 - c^3*C + 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 - B*d^3) + a^2*(
c^3*C - 3*B*c^2*d - 3*c*C*d^2 + B*d^3 - A*(c^3 - 3*c*d^2)) + 2*a*b*((A - C)
*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)))*x)/((a^2 + b^2)^2*(c^2 + d^2)^3)) +
(b^2*(4*a^3*b*B*d - 3*a^4*C*d + b^4*(B*c - 3*A*d) + 2*a*b^3*(A*c - c*C + B*
d) - a^2*b^2*(B*c + (5*A + C)*d))*Log[a*Cos[e + f*x] + b*Sin[e + f*x]])/((a
^2 + b^2)^2*(b*c - a*d)^4*f) + (d*(b^2*(3*c^6*C - 6*B*c^5*d + c^4*(10*A - C
)*d^2 - 3*B*c^3*d^3 + 9*A*c^2*d^4 - B*c*d^5 + 3*A*d^6) + a^2*d^3*((A - C)*d
```

```
*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) - 2*a*b*d^2*(c*(A - C)*d*(5*c^2 + d^2)
- B*(2*c^4 - 3*c^2*d^2 - d^4)))*Log[c*Cos[e + f*x] + d*Sin[e + f*x]])/((b*c
 - a*d)^4*(c^2 + d^2)^3*f) - (d*(a^2*A*d^2 + b^2*c*(c*C - B*d) - 2*a*b*B*(c
^2 + d^2) + A*b^2*(2*c^2 + 3*d^2) + a^2*(3*c^2*C - B*c*d + 2*C*d^2)))/(2*(a
^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^2) - (A*b^2 - a*
(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])*(c + d*Tan[e +
 f*x])^2) - (d*(b^3*c*(2*c^3*C - 3*B*c^2*d - B*d^3) + a^2*b*(3*c^4*C - 3*B*
c^3*d + 2*c^2*C*d^2 - B*c*d^3 + C*d^4) + a^3*d^2*(2*c*C*d + B*(c^2 - d^2))
+ a*b^2*(2*c*C*d^3 - B*(c^4 + c^2*d^2 + 2*d^4)) - A*(2*a^3*c*d^3 + 2*a*b^2*
c*d^3 - 2*a^2*b*d^2*(2*c^2 + d^2) - b^3*(c^4 + 6*c^2*d^2 + 3*d^4))))/((a^2
+ b^2)*(b*c - a*d)^3*(c^2 + d^2)^2*f*(c + d*Tan[e + f*x]))
```

Rule 3530

```
Int[((c_) + (d_.)*tan[(e_.) + (f_.)*(x_)])/((a_) + (b_.)*tan[(e_.) + (f_.)*
(x_)]), x_Symbol] :> Simp[(c*Log[RemoveContent[a*Cos[e + f*x] + b*Sin[e + f
*x], x]])/(b*f), x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] &&
 NeQ[a^2 + b^2, 0] && EqQ[a*c + b*d, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3651

```
Int[((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^
2)/(((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])), x_Symbol] :> Simp[((a*(A*c - c*C + B*d) + b*(B*c - A*d + C*d))*x)
/((a^2 + b^2)*(c^2 + d^2)), x] + (Dist[(A*b^2 - a*b*B + a^2*C)/((b*c - a*d)
*(a^2 + b^2)), Int[(b - a*Tan[e + f*x])/(a + b*Tan[e + f*x]), x], x] - Dist
[(c^2*C - B*c*d + A*d^2)/((b*c - a*d)*(c^2 + d^2)), Int[(d - c*Tan[e + f*x]
)/(c + d*Tan[e + f*x]), x], x]) /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] &&
 NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^3}\,dx = -\frac{Ab^2-a(bB-aC)}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^2} -$$

$$= -\frac{d\left(a^2Ad^2+b^2c(cC-Bd)-2abB\left(c^2+d^2\right)+Ab^2\left(2c^2+3d^2\right)\right.}{2\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f(c+d\tan}$$

$$= -\frac{d\left(a^2Ad^2+b^2c(cC-Bd)-2abB\left(c^2+d^2\right)+Ab^2\left(2c^2+3d^2\right)\right.}{2\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f(c+d\tan}$$

$$= -\frac{\left(b^2\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3\right)+a^2\left(c^3C-\right.\right.}{}$$

$$= -\frac{\left(b^2\left(Ac^3-c^3C+3Bc^2d-3Acd^2+3cCd^2-Bd^3\right)+a^2\left(c^3C-\right.\right.}{}$$

Mathematica [B] time = 8.74, size = 1732, normalized size = 2.01

$$-\frac{Ab^2-a(bB-aC)}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^2} - \frac{-\frac{d^2\left(3Adb^2-aA(bc-ad)-(bB-aC)(bc+2ad)\right)-c\left((Ab-Cb-aB)d(bc-\right.}{2(ad-bc)\left(c^2+d^2\right)f(c+d\tan(e+fx))^2}}{}$$

Warning: Unable to verify antiderivative.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^3),x]
```

```
[Out] -((A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])*(
c + d*Tan[e + f*x])^2)) - (-1/2*(-(c*(-3*c*(A*b^2 - a*(b*B - a*C))*d + (A*b
 - a*B - b*C)*d*(b*c - a*d))) + d^2*(3*A*b^2*d - a*A*(b*c - a*d) - (b*B - a
*C)*(b*c + 2*a*d)))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^2) -
 (-(((-(((b*c - a*d)^3*(-(b^2*(2*a*A*b*c^3 - a^2*B*c^3 + b^2*B*c^3 - 2*a*b*c
^3*C + 3*a^2*A*c^2*d - 3*A*b^2*c^2*d + 6*a*b*B*c^2*d - 3*a^2*c^2*C*d + 3*b^
2*c^2*C*d - 6*a*A*b*c*d^2 + 3*a^2*B*c*d^2 - 3*b^2*B*c*d^2 + 6*a*b*c*C*d^2 -
 a^2*A*d^3 + A*b^2*d^3 - 2*a*b*B*d^3 + a^2*C*d^3 - b^2*C*d^3)) + Sqrt[-b^2]
*(-(a^2*A*b*c^3) + A*b^3*c^3 - 2*a*b^2*B*c^3 + a^2*b*c^3*C - b^3*c^3*C + 6*
a*A*b^2*c^2*d - 3*a^2*b*B*c^2*d + 3*b^3*B*c^2*d - 6*a*b^2*c^2*C*d + 3*a^2*A
*b*c*d^2 - 3*A*b^3*c*d^2 + 6*a*b^2*B*c*d^2 - 3*a^2*b*c*C*d^2 + 3*b^3*c*C*d^
```

```
2 - 2*a*A*b^2*d^3 + a^2*b*B*d^3 - b^3*B*d^3 + 2*a*b^2*C*d^3))*Log[Sqrt[-b^2
] - b*Tan[e + f*x]])/(b*(a^2 + b^2)*(c^2 + d^2))) - (2*b^3*(c^2 + d^2)^2*(4
*a^3*b*B*d - 3*a^4*C*d + b^4*(B*c - 3*A*d) + 2*a*b^3*(A*c - c*C + B*d) - a^
2*b^2*(B*c + (5*A + C)*d))*Log[a + b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d
)) + ((b*c - a*d)^3*(b^2*(2*a*A*b*c^3 - a^2*B*c^3 + b^2*B*c^3 - 2*a*b*c^3*C
 + 3*a^2*A*c^2*d - 3*A*b^2*c^2*d + 6*a*b*B*c^2*d - 3*a^2*c^2*C*d + 3*b^2*c^
2*C*d - 6*a*A*b*c*d^2 + 3*a^2*B*c*d^2 - 3*b^2*B*c*d^2 + 6*a*b*c*C*d^2 - a^2
*A*d^3 + A*b^2*d^3 - 2*a*b*B*d^3 + a^2*C*d^3 - b^2*C*d^3) + Sqrt[-b^2]*(-(a
^2*A*b*c^3) + A*b^3*c^3 - 2*a*b^2*B*c^3 + a^2*b*c^3*C - b^3*c^3*C + 6*a*A*b
^2*c^2*d - 3*a^2*b*B*c^2*d + 3*b^3*B*c^2*d - 6*a*b^2*c^2*C*d + 3*a^2*A*b*c*
d^2 - 3*A*b^3*c*d^2 + 6*a*b^2*B*c*d^2 - 3*a^2*b*c*C*d^2 + 3*b^3*c*C*d^2 - 2
*a*A*b^2*d^3 + a^2*b*B*d^3 - b^3*B*d^3 + 2*a*b^2*C*d^3))*Log[Sqrt[-b^2] + b
*Tan[e + f*x]])/(b*(a^2 + b^2)*(c^2 + d^2)) - (2*b*(a^2 + b^2)*d*(b^2*(3*c^
6*C - 6*B*c^5*d + c^4*(10*A - C)*d^2 - 3*B*c^3*d^3 + 9*A*c^2*d^4 - B*c*d^5
+ 3*A*d^6) + a^2*d^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) - 2*a*b*
d^2*(c*(A - C)*d*(5*c^2 + d^2) - B*(2*c^4 - 3*c^2*d^2 - d^4)))*Log[c + d*Ta
n[e + f*x]])/((b*c - a*d)*(c^2 + d^2)))/(b*(-(b*c) + a*d)*(c^2 + d^2)*f)) -
 (d^2*(-2*a*d*(-3*c*(A*b^2 - a*(b*B - a*C))*d + (A*b - a*B - b*C)*d*(b*c -
a*d)) + (2*b*d^2 - 2*c*(-(b*c) + a*d))*(3*A*b^2*d - a*A*(b*c - a*d) - (b*B
- a*C)*(b*c + 2*a*d))) - c*(2*d*(-(b*c) + a*d)*(-3*(A*b^2 - a*(b*B - a*C))*
d^2 - c*(A*b - a*B - b*C)*(b*c - a*d) + d*(3*A*b^2*d - a*A*(b*c - a*d) - (b
*B - a*C)*(b*c + 2*a*d))) - 2*b*c*(-(c*(-3*c*(A*b^2 - a*(b*B - a*C))*d + (A
*b - a*B - b*C)*d*(b*c - a*d))) + d^2*(3*A*b^2*d - a*A*(b*c - a*d) - (b*B -
 a*C)*(b*c + 2*a*d)))))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))
)/(2*(-(b*c) + a*d)*(c^2 + d^2)))/((a^2 + b^2)*(b*c - a*d))
```

<u>fricas</u> [B] time = 20.87, size = 9567, normalized size = 11.11

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] -1/2*(2*(C*a^2*b^5 - B*a*b^6 + A*b^7)*c^9 - 2*(C*a^3*b^4 - B*a^2*b^5 + A*a*
b^6)*c^8*d + 6*(C*a^2*b^5 - B*a*b^6 + A*b^7)*c^7*d^2 + (7*C*a^5*b^2 + 8*C*a
^3*b^4 + 6*B*a^2*b^5 - (6*A - 7*C)*a*b^6)*c^6*d^3 - (10*C*a^6*b + 9*B*a^5*b
^2 + 20*C*a^4*b^3 + 18*B*a^3*b^4 + 4*C*a^2*b^5 + 15*B*a*b^6 - 6*A*b^7)*c^5*
d^4 + (3*C*a^7 + 14*B*a^6*b + (11*A + 7*C)*a^5*b^2 + 28*B*a^4*b^3 + (22*A -
 C)*a^3*b^4 + 20*B*a^2*b^5 + (5*A + C)*a*b^6)*c^4*d^5 - (5*B*a^7 + 2*(9*A -
 C)*a^6*b + 13*B*a^5*b^2 + 4*(9*A - C)*a^4*b^3 + 11*B*a^3*b^4 + 2*(9*A - 2*
C)*a^2*b^5 + 5*B*a*b^6 - 2*A*b^7)*c^3*d^6 + ((7*A - 3*C)*a^7 + 2*B*a^6*b +
(19*A - 6*C)*a^5*b^2 + 4*B*a^4*b^3 + (17*A - 5*C)*a^3*b^4 + 4*B*a^2*b^5 + 3
*A*a*b^6)*c^2*d^7 + (B*a^7 - 6*A*a^6*b + 2*B*a^5*b^2 - 12*A*a^4*b^3 + B*a^3
*b^4 - 6*A*a^2*b^5)*c*d^8 + (A*a^7 + 2*A*a^5*b^2 + A*a^3*b^4)*d^9 - (2*(C*a
```

^3*b^4 - B*a^2*b^5 + A*a*b^6)*c^7*d^2 + (3*C*a^4*b^3 + 2*B*a^3*b^4 - 2*(A - 5*C)*a^2*b^5 + 5*C*b^7)*c^6*d^3 - (6*C*a^5*b^2 + 7*B*a^4*b^3 + 6*C*a^3*b^4 + 20*B*a^2*b^5 - 6*(A - C)*a*b^6 + 7*B*b^7)*c^5*d^4 + (C*a^6*b + 10*B*a^5*b^2 + (9*A - 5*C)*a^4*b^3 + 26*B*a^3*b^4 + (12*A - C)*a^2*b^5 + 10*B*a*b^6 + (9*A - C)*b^7)*c^4*d^5 - (3*B*a^6*b + 2*(7*A - 3*C)*a^5*b^2 + 7*B*a^4*b^3 + 2*(14*A - 9*C)*a^3*b^4 + 11*B*a^2*b^5 + 2*(4*A - 3*C)*a*b^6 + B*b^7)*c^3*d^6 + (5*(A - C)*a^6*b - 2*B*a^5*b^2 + (13*A - 16*C)*a^4*b^3 + 2*B*a^3*b^4 + 5*(A - C)*a^2*b^5 - 2*B*a*b^6 + 3*A*b^7)*c^2*d^7 + (3*B*a^6*b - 2*A*a^5*b^2 + 6*B*a^4*b^3 - 2*(2*A - C)*a^3*b^4 + B*a^2*b^5)*c*d^8 - (A*a^6*b + 2*(A + C)*a^4*b^3 - 2*B*a^3*b^4 + 3*A*a^2*b^5)*d^9 + 2*(((A - C)*a^2*b^5 + 2*B*a*b^6 - (A - C)*b^7)*c^7*d^2 - (4*(A - C)*a^3*b^4 + 5*B*a^2*b^5 + 2*(A - C)*a*b^6 + 3*B*b^7)*c^6*d^3 + 3*(2*(A - C)*a^4*b^3 + 5*(A - C)*a^2*b^5 + 2*B*a*b^6 + (A - C)*b^7)*c^5*d^4 - (4*(A - C)*a^5*b^2 - 10*B*a^4*b^3 + 20*(A - C)*a^3*b^4 - 5*B*a^2*b^5 + 10*(A - C)*a*b^6 - B*b^7)*c^4*d^5 + ((A - C)*a^6*b - 10*B*a^5*b^2 + 5*(A - C)*a^4*b^3 - 20*B*a^3*b^4 + 10*(A - C)*a^2*b^5 - 4*B*a*b^6)*c^3*d^6 + 3*(B*a^6*b + 2*(A - C)*a^5*b^2 + 5*B*a^4*b^3 + 2*B*a^2*b^5)*c^2*d^7 - (3*(A - C)*a^6*b + 2*B*a^5*b^2 + 5*(A - C)*a^4*b^3 + 4*B*a^3*b^4)*c*d^8 - (B*a^6*b - 2*(A - C)*a^5*b^2 - B*a^4*b^3)*d^9)*f*x)*tan(f*x + e)^3 - 2*(((A - C)*a^3*b^4 + 2*B*a^2*b^5 - (A - C)*a*b^6)*c^9 - (4*(A - C)*a^4*b^3 + 5*B*a^3*b^4 + 2*(A - C)*a^2*b^5 + 3*B*a*b^6)*c^8*d + 3*(2*(A - C)*a^5*b^2 + 5*(A - C)*a^3*b^4 + 2*B*a^2*b^5 + (A - C)*a*b^6)*c^7*d^2 - (4*(A - C)*a^6*b - 10*B*a^5*b^2 + 20*(A - C)*a^4*b^3 - 5*B*a^3*b^4 + 10*(A - C)*a^2*b^5 - B*a*b^6)*c^6*d^3 + ((A - C)*a^7 - 10*B*a^6*b + 5*(A - C)*a^5*b^2 - 20*B*a^4*b^3 + 10*(A - C)*a^3*b^4 - 4*B*a^2*b^5)*c^5*d^4 + 3*(B*a^7 + 2*(A - C)*a^6*b + 5*B*a^5*b^2 + 2*B*a^3*b^4)*c^4*d^5 - (3*(A - C)*a^7 + 2*B*a^6*b + 5*(A - C)*a^5*b^2 + 4*B*a^4*b^3)*c^3*d^6 - (B*a^7 - 2*(A - C)*a^6*b - B*a^5*b^2)*c^2*d^7)*f*x - (4*(C*a^3*b^4 - B*a^2*b^5 + A*a*b^6)*c^8*d + 2*(C*a^4*b^3 + 2*B*a^3*b^4 - (2*A - 5*C)*a^2*b^5 + B*a*b^6 - (A - 3*C)*b^7)*c^7*d^2 - (3*C*a^5*b^2 + 8*B*a^4*b^3 - 8*C*a^3*b^4 + 30*B*a^2*b^5 - (14*A - 3*C)*a*b^6 + 8*B*b^7)*c^6*d^3 - (4*C*a^6*b - 5*B*a^5*b^2 - 2*(5*A - 13*C)*a^4*b^3 - 22*B*a^3*b^4 - 2*(4*A - 11*C)*a^2*b^5 - 11*B*a*b^6 - 2*(2*A - 3*C)*b^7)*c^5*d^4 + (C*a^7 + 6*B*a^6*b - (7*A - 13*C)*a^5*b^2 + 18*B*a^4*b^3 - (14*A - 41*C)*a^3*b^4 + 11*(A + C)*a*b^6 + 6*B*b^7)*c^4*d^5 - (3*B*a^7 + 8*A*a^6*b + 19*B*a^5*b^2 + 2*(11*A + 6*C)*a^4*b^3 + 17*B*a^3*b^4 + 2*(16*A + 3*C)*a^2*b^5 + 7*B*a*b^6 + 12*A*b^7)*c^3*d^6 + (5*(A - C)*a^7 + 4*B*a^6*b + (25*A - 14*C)*a^5*b^2 + 10*B*a^4*b^3 + (35*A - 3*C)*a^3*b^4 - 2*B*a^2*b^5 + (25*A - 4*C)*a*b^6 + 2*B*b^7)*c^2*d^7 + (3*B*a^7 - 4*(2*A - C)*a^6*b + 6*B*a^5*b^2 - 4*(5*A - C)*a^4*b^3 + 7*B*a^3*b^4 - 2*(10*A - C)*a^2*b^5 + 2*B*a*b^6 - 6*A*b^7)*c*d^8 - (A*a^7 + 2*B*a^6*b - 2*A*a^5*b^2 + 4*B*a^4*b^3 - (7*A + 2*C)*a^3*b^4 + 4*B*a^2*b^5 - 6*A*a*b^6)*d^9 + 2*(2*((A - C)*a^2*b^5 + 2*B*a*b^6 - (A - C)*b^7)*c^8*d - (7*(A - C)*a^3*b^4 + 8*B*a^2*b^5 + 5*(A - C)*a*b^6 + 6*B*b^7)*c^7*d^2 + (8*(A - C)*a^4*b^3 - 5*B*a^3*b^4 + 28*(A - C)*a^2*b^5 + 9*B*a*b^6 + 6*(A - C)*b^7)*c^6*d^3 - (2*(A - C)*a^5*b^2 - 20*B*a^4*b^3 + 25*(A - C)*a^3*b^4 - 16*B*a^2*b^5 + 17*(A - C)*a*b^6 - 2*B*b^7)*c^5*d^4 - (2*(A - C)*a^6*b + 10*B*a^5*b^2 + 10*(A - C)*a^4*b^3 + 35*B*a^3 *b^4 - 10*(A - C)*a^2*b^5 + 7*B*a*b^6)*c^4*d^5 + ((A - C)*a^7 - 4*B*a^6*b + 17*(A - C)*a^5*b^2 + 10*B*a^4*b^3 + 10*(A - C)*a^3*b^4 + 8*B*a^2*b^5)*c^3*d^6 + (3*B*a^7 + 11*B*a^5*b^2 - 10*(A - C)*a^4*b^3 - 2*B*a^3*b^4)*c^2*d^7 - (3*(A - C)*a^7 + 4*B*a^6*b + (A - C)*a^5*b^2 + 2*B*a^4*b^3)*c*d^8 - (B*a^7 - 2*(A - C)*a^6*b - B*a^5*b^2)*d^9)*f*x)*tan(f*x + e)^2 + ((B*a^3*b^4 - 2*(A - C)*a^2*b^5 - B*a*b^6)*c^9 + (3*C*a^5*b^2 - 4*B*a^4*b^3 + (5*A + C)*a^3*b^4 - 2*B*a^2*b^5 + 3*A*a*b^6)*c^8*d + 3*(B*a^3*b^4 - 2*(A - C)*a^2*b^5 - B*a*b^6)*c^7*d^2 + 3*(3*C*a^5*b^2 - 4*B*a^4*b^3 + (5*A + C)*a^3*b^4 - 2*B*a^2*b^5 + 3*A*a*b^6)*c^6*d^3 + 3*(B*a^3*b^4 - 2*(A - C)*a^2*b^5 - B*a*b^6)*c^5*d^4 + 3*(3*C*a^5*b^2 - 4*B*a^4*b^3 + (5*A + C)*a^3*b^4 - 2*B*a^2*b^5 + 3*A*a*b^6)*c^4*d^5 + (B*a^3*b^4 - 2*(A - C)*a^2*b^5 - B*a*b^6)*c^3*d^6 + (3*C*a^5*b^2 - 4*B*a^4*b^3 + (5*A + C)*a^3*b^4 - 2*B*a^2*b^5 + 3*A*a*b^6)*c^2*d^7 + ((B*a^2*b^5 - 2*(A - C)*a*b^6 - B*b^7)*c^7*d^2 + (3*C*a^4*b^3 - 4*B*a^3*b^4 + (5*A + C)*a^2*b^5 - 2*B*a*b^6 + 3*A*b^7)*c^6*d^3 + 3*(B*a^2*b^5 - 2*(A - C)*a*b^6 - B*b^7)*c^5*d^4 + 3*(3*C*a^4*b^3 - 4*B*a^3*b^4 + (5*A + C)*a^2*b^5 - 2*B*a*b^6 + 3*A*b^7)*c^4*d^5 + 3*(B*a^2*b^5 - 2*(A - C)*a*b^6 - B*b^7)*c^3*d^6 + 3*(3*C*a^4*b^3 - 4*B*a^3*b^4 + (5*A + C)*a^2*b^5 - 2*B*a*b^6 + 3*A*b^7)*c^2*d^7 + (B*a^2*b^5 - 2*(A - C)*a*b^6 - B*b^7)*c*d^8 + (3*C*a^4*b^3 - 4*B*a^3*b^4 + (5*A + C)*a^2*b^5 - 2*B*a*b^6 + 3*A*b^7)*d^9)*tan(f*x + e)^3 + (2*(B*a^2*b^5 - 2*(A - C)*a*b^6 - B*b^7)*c^8*d + (6*C*a^4*b^3 - 7*B*a^3*b^4 + 4*(2*A + C)*a^2*b^5 - 5*B*a*b^6 + 6*A*b^7)*c^7*d^2 + (3*C*a^5*b^2 - 4*B*a^4*b^3 + (5*A + C)*a^3*b^4 + 4*B*a^2*b^5 - 3*(3*A - 4*C)*a*b^6 - 6*B*b^7)*c^6*d^3 + 3*(6*C*a^4*b^3 - 7*B*a^3*b^4 + 4*(2*A + C)*a^2*b^5 - 5*B*a*b^6 + 6*A*b^7)*c^5*d^4 + 3*(3*C*a^5*b^2 - 4*B*a^4*b^3 + (5*A + C)*a^3*b^4 - (A - 4*C)*a*b^6 - 2*B*b^7)*c^4*d^5 + 3*(6*C*a^4*b^3 - 7*B*a^3*b^4 + 4*(2*A + C)*a^2*b^5 - 5*B*a*b^6 + 6*A*b^7)*c^3*d^6 + (9*C*a^5*b^2 - 12*B*a^4*b^3 + 3*(5*A + C)*a^3*b^4 - 4*B*a^2*b^5 + (5*A + 4*C)*a*b^6 - 2*B*b^7)*c^2*d^7 + (6*C*a^4*b^3 - 7*B*a^3*b^4 + 4*(2*A + C)*a^2*b^5 - 5*B*a*b^6 + 6*A*b^7)*c*d^8 + (3*C*a^5*b^2 - 4*B*a^4*b^3 + (5*A + C)*a^3*b^4 - 2*B*a^2*b^5 + 3*A*a*b^6)*d^9)*tan(f*x + e)^2 + ((B*a^2*b^5 - 2*(A - C)*a*b^6 - B*b^7)*c^9 + (3*C*a^4*b^3 - 2*B*a^3*b^4 + (A + 5*C)*a^2*b^5 - 4*B*a*b^6 + 3*A*b^7)*c^8*d + (6*C*a^5*b^2 - 8*B*a^4*b^3 + 2*(5*A + C)*a^3*b^4 - B*a^2*b^5 + 6*C*a*b^6 - 3*B*b^7)*c^7*d^2 + 3*(3*C*a^4*b^3 - 2*B*a^3*b^4 + (A + 5*C)*a^2*b^5 - 4*B*a*b^6 + 3*A*b^7)*c^6*d^3 + 3*(6*C*a^5*b^2 - 8*B*a^4*b^3 + 2*(5*A + C)*a^3*b^4 - 3*B*a^2*b^5 + 2*(2*A + C)*a*b^6 - B*b^7)*c^5*d^4 + 3*(3*C*a^4*b^3 - 2*B*a^3*b^4 + (A + 5*C)*a^2*b^5 - 4*B*a*b^6 + 3*A*b^7)*c^4*d^5 + (18*C*a^5*b^2 - 24*B*a^4*b^3 + 6*(5*A + C)*a^3*b^4 - 11*B*a^2*b^5 + 2*(8*A + C)*a*b^6 - B*b^7)*c^3*d^6 + (3*C*a^4*b^3 - 2*B*a^3*b^4 + (A + 5*C)*a^2*b^5 - 4*B*a*b^6 + 3*A*b^7)*c^2*d^7 + 2*(3*C*a^5*b^2 - 4*B*a^4*b^3 + (5*A + C)*a^3*b^4 - 2*B*a^2*b^5 + 3*A*a*b^6)*c*d^8)*tan(f*x + e))*log((b^2*tan(f*x + e)^2 + 2*a*b*tan(f*x + e) + a^2)/(tan(f*x + e)^2 + 1)) - (3*(C*a^5*b^2 + 2*C*a^3*b^4 + C*a*b^6)*c^8*d - 6*(B*a^5*b^2 + 2*B*a^3*b^4 + B*a*b^6)*c^7*d^2 + (4*B*a^6*b + (10*A - C)*a^5*b^2 + 8*B*a^4*b^3 + 2*(10*A - C)*a^3*b^4 + 4*B*a^2*b^5 + (10*A - C)*a*b^6)*c^6*d^3 - (B*a^7 + 10*(A - C)*a^6*b + 5*B*a^5*b^2 + 20*(A - C)*a^4*b^3 + 7*B*a^3*b^4 + 10*(A - C)*a^2*b^5 + 3*B*a*b^6)*c^5*d^

```
4 + 3*((A - C)*a^7 - 2*B*a^6*b + (5*A - 2*C)*a^5*b^2 - 4*B*a^4*b^3 + (7*A -
 C)*a^3*b^4 - 2*B*a^2*b^5 + 3*A*a*b^6)*c^4*d^5 + (3*B*a^7 - 2*(A - C)*a^6*b
 + 5*B*a^5*b^2 - 4*(A - C)*a^4*b^3 + B*a^3*b^4 - 2*(A - C)*a^2*b^5 - B*a*b^
6)*c^3*d^6 - ((A - C)*a^7 + 2*B*a^6*b - (A + 2*C)*a^5*b^2 + 4*B*a^4*b^3 - (
5*A + C)*a^3*b^4 + 2*B*a^2*b^5 - 3*A*a*b^6)*c^2*d^7 + (3*(C*a^4*b^3 + 2*C*a
^2*b^5 + C*b^7)*c^6*d^3 - 6*(B*a^4*b^3 + 2*B*a^2*b^5 + B*b^7)*c^5*d^4 + (4*
B*a^5*b^2 + (10*A - C)*a^4*b^3 + 8*B*a^3*b^4 + 2*(10*A - C)*a^2*b^5 + 4*B*a
*b^6 + (10*A - C)*b^7)*c^4*d^5 - (B*a^6*b + 10*(A - C)*a^5*b^2 + 5*B*a^4*b^
3 + 20*(A - C)*a^3*b^4 + 7*B*a^2*b^5 + 10*(A - C)*a*b^6 + 3*B*b^7)*c^3*d^6
+ 3*((A - C)*a^6*b - 2*B*a^5*b^2 + (5*A - 2*C)*a^4*b^3 - 4*B*a^3*b^4 + (7*A
 - C)*a^2*b^5 - 2*B*a*b^6 + 3*A*b^7)*c^2*d^7 + (3*B*a^6*b - 2*(A - C)*a^5*b
^2 + 5*B*a^4*b^3 - 4*(A - C)*a^3*b^4 + B*a^2*b^5 - 2*(A - C)*a*b^6 - B*b^7)
*c*d^8 - ((A - C)*a^6*b + 2*B*a^5*b^2 - (A + 2*C)*a^4*b^3 + 4*B*a^3*b^4 - (
5*A + C)*a^2*b^5 + 2*B*a*b^6 - 3*A*b^7)*d^9)*tan(f*x + e)^3 + (6*(C*a^4*b^3
 + 2*C*a^2*b^5 + C*b^7)*c^7*d^2 + 3*(C*a^5*b^2 - 4*B*a^4*b^3 + 2*C*a^3*b^4
- 8*B*a^2*b^5 + C*a*b^6 - 4*B*b^7)*c^6*d^3 + 2*(B*a^5*b^2 + (10*A - C)*a^4*
b^3 + 2*B*a^3*b^4 + 2*(10*A - C)*a^2*b^5 + B*a*b^6 + (10*A - C)*b^7)*c^5*d^
4 + (2*B*a^6*b - (10*A - 19*C)*a^5*b^2 - 2*B*a^4*b^3 - 2*(10*A - 19*C)*a^3*
b^4 - 10*B*a^2*b^5 - (10*A - 19*C)*a*b^6 - 6*B*b^7)*c^4*d^5 - (B*a^7 + 4*(A
 - C)*a^6*b + 17*B*a^5*b^2 - 2*(5*A + 4*C)*a^4*b^3 + 31*B*a^3*b^4 - 4*(8*A
+ C)*a^2*b^5 + 15*B*a*b^6 - 18*A*b^7)*c^3*d^6 + (3*(A - C)*a^7 + (11*A - 2*
C)*a^5*b^2 - 2*B*a^4*b^3 + (13*A + 5*C)*a^3*b^4 - 4*B*a^2*b^5 + (5*A + 4*C)
*a*b^6 - 2*B*b^7)*c^2*d^7 + (3*B*a^7 - 4*(A - C)*a^6*b + B*a^5*b^2 - 2*(A -
 4*C)*a^4*b^3 - 7*B*a^3*b^4 + 4*(2*A + C)*a^2*b^5 - 5*B*a*b^6 + 6*A*b^7)*c*
d^8 - ((A - C)*a^7 + 2*B*a^6*b - (A + 2*C)*a^5*b^2 + 4*B*a^4*b^3 - (5*A + C
)*a^3*b^4 + 2*B*a^2*b^5 - 3*A*a*b^6)*d^9)*tan(f*x + e)^2 + (3*(C*a^4*b^3 +
2*C*a^2*b^5 + C*b^7)*c^8*d + 6*(C*a^5*b^2 - B*a^4*b^3 + 2*C*a^3*b^4 - 2*B*a
^2*b^5 + C*a*b^6 - B*b^7)*c^7*d^2 - (8*B*a^5*b^2 - (10*A - C)*a^4*b^3 + 16*
B*a^3*b^4 - 2*(10*A - C)*a^2*b^5 + 8*B*a*b^6 - (10*A - C)*b^7)*c^6*d^3 + (7
*B*a^6*b + 2*(5*A + 4*C)*a^5*b^2 + 11*B*a^4*b^3 + 4*(5*A + 4*C)*a^3*b^4 + B
*a^2*b^5 + 2*(5*A + 4*C)*a*b^6 - 3*B*b^7)*c^5*d^4 - (2*B*a^7 + 17*(A - C)*a
^6*b + 16*B*a^5*b^2 + (25*A - 34*C)*a^4*b^3 + 26*B*a^3*b^4 - (A + 17*C)*a^2
*b^5 + 12*B*a*b^6 - 9*A*b^7)*c^4*d^5 + (6*(A - C)*a^7 - 9*B*a^6*b + 2*(14*A
 - 5*C)*a^5*b^2 - 19*B*a^4*b^3 + 2*(19*A - C)*a^3*b^4 - 11*B*a^2*b^5 + 2*(8
*A + C)*a*b^6 - B*b^7)*c^3*d^6 + (6*B*a^7 - 5*(A - C)*a^6*b + 8*B*a^5*b^2 -
 (7*A - 10*C)*a^4*b^3 - 2*B*a^3*b^4 + (A + 5*C)*a^2*b^5 - 4*B*a*b^6 + 3*A*b
^7)*c^2*d^7 - 2*((A - C)*a^7 + 2*B*a^6*b - (A + 2*C)*a^5*b^2 + 4*B*a^4*b^3
- (5*A + C)*a^3*b^4 + 2*B*a^2*b^5 - 3*A*a*b^6)*c*d^8)*tan(f*x + e))*log((d^
2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2)/(tan(f*x + e)^2 + 1)) - (2*(C*
a^3*b^4 - B*a^2*b^5 + A*a*b^6)*c^9 - 2*(C*a^4*b^3 - B*a^3*b^4 + (A + 2*C)*a
^2*b^5 - 2*B*a*b^6 + 2*A*b^7)*c^8*d + 2*(3*C*a^5*b^2 + 11*C*a^3*b^4 - 5*B*a
^2*b^5 + (5*A + 3*C)*a*b^6)*c^7*d^2 - (8*C*a^6*b + 8*B*a^5*b^2 + 29*C*a^4*b
^3 + 10*B*a^3*b^4 + 2*(3*A + 17*C)*a^2*b^5 - 4*B*a*b^6 + (12*A + 7*C)*b^7)*
c^6*d^3 + (2*C*a^7 + 12*B*a^6*b + 2*(5*A + 4*C)*a^5*b^2 + 33*B*a^4*b^3 + 4*
(5*A + 7*C)*a^3*b^4 + 12*B*a^2*b^5 + 4*(7*A + C)*a*b^6 + 9*B*b^7)*c^5*d^4 -
```

(4*B*a^7 + (16*A - 9*C)*a^6*b + 16*B*a^5*b^2 + (43*A - 11*C)*a^4*b^3 + 14*B*a^3*b^4 + (44*A + 5*C)*a^2*b^5 - 4*B*a*b^6 + (23*A + C)*b^7)*c^4*d^5 + (6*(A - C)*a^7 - 7*B*a^6*b + 2*(12*A - 7*C)*a^5*b^2 - 11*B*a^4*b^3 + 2*(15*A + 2*C)*a^3*b^4 - 15*B*a^2*b^5 + 2*(13*A - C)*a*b^6 + 3*B*b^7)*c^3*d^6 + (6*B*a^7 + (5*A - C)*a^6*b + 12*B*a^5*b^2 + (5*A - 4*C)*a^4*b^3 + 8*B*a^3*b^4 - (7*A + 5*C)*a^2*b^5 + 4*B*a*b^6 - 9*A*b^7)*c^2*d^7 - (2*(3*A - 2*C)*a^7 + B*a^6*b + 2*(5*A - 4*C)*a^5*b^2 + 2*B*a^4*b^3 + 2*(A - 4*C)*a^3*b^4 + 5*B*a^2*b^5 - 6*A*a*b^6)*c*d^8 - (2*B*a^7 - 3*A*a^6*b + 4*B*a^5*b^2 - 6*A*a^4*b^3 + 2*B*a^3*b^4 - 3*A*a^2*b^5)*d^9 + 2*(((A - C)*a^2*b^5 + 2*B*a*b^6 - (A - C)*b^7)*c^9 - (2*(A - C)*a^3*b^4 + B*a^2*b^5 + 4*(A - C)*a*b^6 + 3*B*b^7)*c^8*d - (2*(A - C)*a^4*b^3 + 10*B*a^3*b^4 - 11*(A - C)*a^2*b^5 - 3*(A - C)*b^7)*c^7*d^2 + (8*(A - C)*a^5*b^2 + 10*B*a^4*b^3 + 10*(A - C)*a^3*b^4 + 17*B*a^2*b^5 - 4*(A - C)*a*b^6 + B*b^7)*c^6*d^3 - (7*(A - C)*a^6*b - 10*B*a^5*b^2 + 35*(A - C)*a^4*b^3 + 10*B*a^3*b^4 + 10*(A - C)*a^2*b^5 + 2*B*a*b^6)*c^5*d^4 + (2*(A - C)*a^7 - 17*B*a^6*b + 16*(A - C)*a^5*b^2 - 25*B*a^4*b^3 + 20*(A - C)*a^3*b^4 - 2*B*a^2*b^5)*c^4*d^5 + (6*B*a^7 + 9*(A - C)*a^6*b + 28*B*a^5*b^2 - 5*(A - C)*a^4*b^3 + 8*B*a^3*b^4)*c^3*d^6 - (6*(A - C)*a^7 + 5*B*a^6*b + 8*(A - C)*a^5*b^2 + 7*B*a^4*b^3)*c^2*d^7 - 2*(B*a^7 - 2*(A - C)*a^6*b - B*a^5*b^2)*c*d^8)*f*x)*tan(f*x + e))/(((a^4*b^5 + 2*a^2*b^7 + b^9)*c^10*d^2 - 4*(a^5*b^4 + 2*a^3*b^6 + a*b^8)*c^9*d^3 + 3*(2*a^6*b^3 + 5*a^4*b^5 + 4*a^2*b^7 + b^9)*c^8*d^4 - 4*(a^7*b^2 + 5*a^5*b^4 + 7*a^3*b^6 + 3*a*b^8)*c^7*d^5 + (a^8*b + 20*a^6*b^3 + 40*a^4*b^5 + 24*a^2*b^7 + 3*b^9)*c^6*d^6 - 12*(a^7*b^2 + 3*a^5*b^4 + 3*a^3*b^6 + a*b^8)*c^5*d^7 + (3*a^8*b + 24*a^6*b^3 + 40*a^4*b^5 + 20*a^2*b^7 + b^9)*c^4*d^8 - 4*(3*a^7*b^2 + 7*a^5*b^4 + 5*a^3*b^6 + a*b^8)*c^3*d^9 + 3*(a^8*b + 4*a^6*b^3 + 5*a^4*b^5 + 2*a^2*b^7)*c^2*d^10 - 4*(a^7*b^2 + 2*a^5*b^4 + a^3*b^6)*c*d^11 + (a^8*b + 2*a^6*b^3 + a^4*b^5)*d^12)*f*tan(f*x + e)^3 + (2*(a^4*b^5 + 2*a^2*b^7 + b^9)*c^11*d - 7*(a^5*b^4 + 2*a^3*b^6 + a*b^8)*c^10*d^2 + 2*(4*a^6*b^3 + 11*a^4*b^5 + 10*a^2*b^7 + 3*b^9)*c^9*d^3 - (2*a^7*b^2 + 25*a^5*b^4 + 44*a^3*b^6 + 21*a*b^8)*c^8*d^4 - 2*(a^8*b - 10*a^6*b^3 - 26*a^4*b^5 - 18*a^2*b^7 - 3*b^9)*c^7*d^5 + (a^9 - 4*a^7*b^2 - 32*a^5*b^4 - 48*a^3*b^6 - 21*a*b^8)*c^6*d^6 - 2*(3*a^8*b - 6*a^6*b^3 - 22*a^4*b^5 - 14*a^2*b^7 - b^9)*c^5*d^7 + (3*a^9 - 16*a^5*b^4 - 20*a^3*b^6 - 7*a*b^8)*c^4*d^8 - 2*(3*a^8*b + 2*a^6*b^3 - 5*a^4*b^5 - 4*a^2*b^7)*c^3*d^9 + (3*a^9 + 4*a^7*b^2 - a^5*b^4 - 2*a^3*b^6)*c^2*d^10 - 2*(a^8*b + 2*a^6*b^3 + a^4*b^5)*c*d^11 + (a^9 + 2*a^7*b^2 + a^5*b^4)*d^12)*f*tan(f*x + e)^2 + ((a^4*b^5 + 2*a^2*b^7 + b^9)*c^12 - 2*(a^5*b^4 + 2*a^3*b^6 + a*b^8)*c^11*d - (2*a^6*b^3 + a^4*b^5 - 4*a^2*b^7 - 3*b^9)*c^10*d^2 + 2*(4*a^7*b^2 + 5*a^5*b^4 - 2*a^3*b^6 - 3*a*b^8)*c^9*d^3 - (7*a^8*b + 20*a^6*b^3 + 16*a^4*b^5 - 3*b^9)*c^8*d^4 + 2*(a^9 + 14*a^7*b^2 + 22*a^5*b^4 + 6*a^3*b^6 - 3*a*b^8)*c^7*d^5 - (21*a^8*b + 48*a^6*b^3 + 32*a^4*b^5 + 4*a^2*b^7 - b^9)*c^6*d^6 + 2*(3*a^9 + 18*a^7*b^2 + 26*a^5*b^4 + 10*a^3*b^6 - a*b^8)*c^5*d^7 - (21*a^8*b + 44*a^6*b^3 + 25*a^4*b^5 + 2*a^2*b^7)*c^4*d^8 + 2*(3*a^9 + 10*a^7*b^2 + 11*a^5*b^4 + 4*a^3*b^6)*c^3*d^9 - 7*(a^8*b + 2*a^6*b^3 + a^4*b^5)*c^2*d^10 + 2*(a^9 + 2*a^7*b^2 + a^5*b^4)*c*d^11)*f*tan(f*x + e) + ((a^5*b^4 + 2*a^3*b^6 + a*b^8)*c^12 - 4*(a^6*b^3 + 2*a^4*b^5 + a^2*b^7)*c^11*d

```
+ 3*(2*a^7*b^2 + 5*a^5*b^4 + 4*a^3*b^6 + a*b^8)*c^10*d^2 - 4*(a^8*b + 5*a^6
*b^3 + 7*a^4*b^5 + 3*a^2*b^7)*c^9*d^3 + (a^9 + 20*a^7*b^2 + 40*a^5*b^4 + 24
*a^3*b^6 + 3*a*b^8)*c^8*d^4 - 12*(a^8*b + 3*a^6*b^3 + 3*a^4*b^5 + a^2*b^7)*
c^7*d^5 + (3*a^9 + 24*a^7*b^2 + 40*a^5*b^4 + 20*a^3*b^6 + a*b^8)*c^6*d^6 -
4*(3*a^8*b + 7*a^6*b^3 + 5*a^4*b^5 + a^2*b^7)*c^5*d^7 + 3*(a^9 + 4*a^7*b^2
+ 5*a^5*b^4 + 2*a^3*b^6)*c^4*d^8 - 4*(a^8*b + 2*a^6*b^3 + a^4*b^5)*c^3*d^9
+ (a^9 + 2*a^7*b^2 + a^5*b^4)*c^2*d^10)*f)
```

giac [B] time = 90.12, size = 3176, normalized size = 3.69

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] 1/2*(2*(A*a^2*c^3 - C*a^2*c^3 + 2*B*a*b*c^3 - A*b^2*c^3 + C*b^2*c^3 + 3*B*a
^2*c^2*d - 6*A*a*b*c^2*d + 6*C*a*b*c^2*d - 3*B*b^2*c^2*d - 3*A*a^2*c*d^2 +
3*C*a^2*c*d^2 - 6*B*a*b*c*d^2 + 3*A*b^2*c*d^2 - 3*C*b^2*c*d^2 - B*a^2*d^3 +
 2*A*a*b*d^3 - 2*C*a*b*d^3 + B*b^2*d^3)*(f*x + e)/(a^4*c^6 + 2*a^2*b^2*c^6
+ b^4*c^6 + 3*a^4*c^4*d^2 + 6*a^2*b^2*c^4*d^2 + 3*b^4*c^4*d^2 + 3*a^4*c^2*d
^4 + 6*a^2*b^2*c^2*d^4 + 3*b^4*c^2*d^4 + a^4*d^6 + 2*a^2*b^2*d^6 + b^4*d^6)
 + (B*a^2*c^3 - 2*A*a*b*c^3 + 2*C*a*b*c^3 - B*b^2*c^3 - 3*A*a^2*c^2*d + 3*C
*a^2*c^2*d - 6*B*a*b*c^2*d + 3*A*b^2*c^2*d - 3*C*b^2*c^2*d - 3*B*a^2*c*d^2
+ 6*A*a*b*c*d^2 - 6*C*a*b*c*d^2 + 3*B*b^2*c*d^2 + A*a^2*d^3 - C*a^2*d^3 + 2
*B*a*b*d^3 - A*b^2*d^3 + C*b^2*d^3)*log(tan(f*x + e)^2 + 1)/(a^4*c^6 + 2*a^
2*b^2*c^6 + b^4*c^6 + 3*a^4*c^4*d^2 + 6*a^2*b^2*c^4*d^2 + 3*b^4*c^4*d^2 + 3
*a^4*c^2*d^4 + 6*a^2*b^2*c^2*d^4 + 3*b^4*c^2*d^4 + a^4*d^6 + 2*a^2*b^2*d^6
+ b^4*d^6) - 2*(B*a^2*b^5*c - 2*A*a*b^6*c + 2*C*a*b^6*c - B*b^7*c + 3*C*a^4
*b^3*d - 4*B*a^3*b^4*d + 5*A*a^2*b^5*d + C*a^2*b^5*d - 2*B*a*b^6*d + 3*A*b^
7*d)*log(abs(b*tan(f*x + e) + a))/(a^4*b^5*c^4 + 2*a^2*b^7*c^4 + b^9*c^4 -
4*a^5*b^4*c^3*d - 8*a^3*b^6*c^3*d - 4*a*b^8*c^3*d + 6*a^6*b^3*c^2*d^2 + 12*
a^4*b^5*c^2*d^2 + 6*a^2*b^7*c^2*d^2 - 4*a^7*b^2*c*d^3 - 8*a^5*b^4*c*d^3 - 4
*a^3*b^6*c*d^3 + a^8*b*d^4 + 2*a^6*b^3*d^4 + a^4*b^5*d^4) + 2*(3*C*b^2*c^6*
d^2 - 6*B*b^2*c^5*d^3 + 4*B*a*b*c^4*d^4 + 10*A*b^2*c^4*d^4 - C*b^2*c^4*d^4
- B*a^2*c^3*d^5 - 10*A*a*b*c^3*d^5 + 10*C*a*b*c^3*d^5 - 3*B*b^2*c^3*d^5 + 3
*A*a^2*c^2*d^6 - 3*C*a^2*c^2*d^6 - 6*B*a*b*c^2*d^6 + 9*A*b^2*c^2*d^6 + 3*B*
a^2*c*d^7 - 2*A*a*b*c*d^7 + 2*C*a*b*c*d^7 - B*b^2*c*d^7 - A*a^2*d^8 + C*a^2
*d^8 - 2*B*a*b*d^8 + 3*A*b^2*d^8)*log(abs(d*tan(f*x + e) + c))/(b^4*c^10*d
- 4*a*b^3*c^9*d^2 + 6*a^2*b^2*c^8*d^3 + 3*b^4*c^8*d^3 - 4*a^3*b*c^7*d^4 - 1
2*a*b^3*c^7*d^4 + a^4*c^6*d^5 + 18*a^2*b^2*c^6*d^5 + 3*b^4*c^6*d^5 - 12*a^3
*b*c^5*d^6 - 12*a*b^3*c^5*d^6 + 3*a^4*c^4*d^7 + 18*a^2*b^2*c^4*d^7 + b^4*c^
4*d^7 - 12*a^3*b*c^3*d^8 - 4*a*b^3*c^3*d^8 + 3*a^4*c^2*d^9 + 6*a^2*b^2*c^2*
d^9 - 4*a^3*b*c*d^10 + a^4*d^11) + 2*(B*a^2*b^5*c*tan(f*x + e) - 2*A*a*b^6*
c*tan(f*x + e) + 2*C*a*b^6*c*tan(f*x + e) - B*b^7*c*tan(f*x + e) + 3*C*a^4*
```

```
b^3*d*tan(f*x + e) - 4*B*a^3*b^4*d*tan(f*x + e) + 5*A*a^2*b^5*d*tan(f*x + e
) + C*a^2*b^5*d*tan(f*x + e) - 2*B*a*b^6*d*tan(f*x + e) + 3*A*b^7*d*tan(f*x
 + e) - C*a^4*b^3*c + 2*B*a^3*b^4*c - 3*A*a^2*b^5*c + C*a^2*b^5*c - A*b^7*c
 + 4*C*a^5*b^2*d - 5*B*a^4*b^3*d + 6*A*a^3*b^4*d + 2*C*a^3*b^4*d - 3*B*a^2*
b^5*d + 4*A*a*b^6*d)/((a^4*b^4*c^4 + 2*a^2*b^6*c^4 + b^8*c^4 - 4*a^5*b^3*c^
3*d - 8*a^3*b^5*c^3*d - 4*a*b^7*c^3*d + 6*a^6*b^2*c^2*d^2 + 12*a^4*b^4*c^2*
d^2 + 6*a^2*b^6*c^2*d^2 - 4*a^7*b*c*d^3 - 8*a^5*b^3*c*d^3 - 4*a^3*b^5*c*d^3
 + a^8*d^4 + 2*a^6*b^2*d^4 + a^4*b^4*d^4)*(b*tan(f*x + e) + a)) - (9*C*b^2*
c^6*d^3*tan(f*x + e)^2 - 18*B*b^2*c^5*d^4*tan(f*x + e)^2 + 12*B*a*b*c^4*d^5
*tan(f*x + e)^2 + 30*A*b^2*c^4*d^5*tan(f*x + e)^2 - 3*C*b^2*c^4*d^5*tan(f*x
 + e)^2 - 3*B*a^2*c^3*d^6*tan(f*x + e)^2 - 30*A*a*b*c^3*d^6*tan(f*x + e)^2
+ 30*C*a*b*c^3*d^6*tan(f*x + e)^2 - 9*B*b^2*c^3*d^6*tan(f*x + e)^2 + 9*A*a^
2*c^2*d^7*tan(f*x + e)^2 - 9*C*a^2*c^2*d^7*tan(f*x + e)^2 - 18*B*a*b*c^2*d^
7*tan(f*x + e)^2 + 27*A*b^2*c^2*d^7*tan(f*x + e)^2 + 9*B*a^2*c*d^8*tan(f*x
+ e)^2 - 6*A*a*b*c*d^8*tan(f*x + e)^2 + 6*C*a*b*c*d^8*tan(f*x + e)^2 - 3*B*
b^2*c*d^8*tan(f*x + e)^2 - 3*A*a^2*d^9*tan(f*x + e)^2 + 3*C*a^2*d^9*tan(f*x
 + e)^2 - 6*B*a*b*d^9*tan(f*x + e)^2 + 9*A*b^2*d^9*tan(f*x + e)^2 + 22*C*b^
2*c^7*d^2*tan(f*x + e) - 4*C*a*b*c^6*d^3*tan(f*x + e) - 42*B*b^2*c^6*d^3*ta
n(f*x + e) + 32*B*a*b*c^5*d^4*tan(f*x + e) + 68*A*b^2*c^5*d^4*tan(f*x + e)
- 2*C*b^2*c^5*d^4*tan(f*x + e) - 8*B*a^2*c^4*d^5*tan(f*x + e) - 72*A*a*b*c^
4*d^5*tan(f*x + e) + 60*C*a*b*c^4*d^5*tan(f*x + e) - 26*B*b^2*c^4*d^5*tan(f
*x + e) + 22*A*a^2*c^3*d^6*tan(f*x + e) - 22*C*a^2*c^3*d^6*tan(f*x + e) - 2
8*B*a*b*c^3*d^6*tan(f*x + e) + 66*A*b^2*c^3*d^6*tan(f*x + e) + 18*B*a^2*c^2
*d^7*tan(f*x + e) - 28*A*a*b*c^2*d^7*tan(f*x + e) + 16*C*a*b*c^2*d^7*tan(f*
x + e) - 8*B*b^2*c^2*d^7*tan(f*x + e) - 2*A*a^2*c*d^8*tan(f*x + e) + 2*C*a^
2*c*d^8*tan(f*x + e) - 12*B*a*b*c*d^8*tan(f*x + e) + 22*A*b^2*c*d^8*tan(f*x
 + e) + 2*B*a^2*d^9*tan(f*x + e) - 4*A*a*b*d^9*tan(f*x + e) + 14*C*b^2*c^8*
d - 6*C*a*b*c^7*d^2 - 25*B*b^2*c^7*d^2 + C*a^2*c^6*d^3 + 22*B*a*b*c^6*d^3 +
 39*A*b^2*c^6*d^3 + 3*C*b^2*c^6*d^3 - 6*B*a^2*c^5*d^4 - 44*A*a*b*c^5*d^4 +
26*C*a*b*c^5*d^4 - 19*B*b^2*c^5*d^4 + 14*A*a^2*c^4*d^5 - 11*C*a^2*c^4*d^5 -
 6*B*a*b*c^4*d^5 + 41*A*b^2*c^4*d^5 + C*b^2*c^4*d^5 + 7*B*a^2*c^3*d^6 - 26*
A*a*b*c^3*d^6 + 8*C*a*b*c^3*d^6 - 6*B*b^2*c^3*d^6 + 3*A*a^2*c^2*d^7 - 4*B*a
*b*c^2*d^7 + 14*A*b^2*c^2*d^7 + B*a^2*c*d^8 - 6*A*a*b*c*d^8 + A*a^2*d^9)/((
b^4*c^10 - 4*a*b^3*c^9*d + 6*a^2*b^2*c^8*d^2 + 3*b^4*c^8*d^2 - 4*a^3*b*c^7*
d^3 - 12*a*b^3*c^7*d^3 + a^4*c^6*d^4 + 18*a^2*b^2*c^6*d^4 + 3*b^4*c^6*d^4 -
 12*a^3*b*c^5*d^5 - 12*a*b^3*c^5*d^5 + 3*a^4*c^4*d^6 + 18*a^2*b^2*c^4*d^6 +
 b^4*c^4*d^6 - 12*a^3*b*c^3*d^7 - 4*a*b^3*c^3*d^7 + 3*a^4*c^2*d^8 + 6*a^2*b
^2*c^2*d^8 - 4*a^3*b*c*d^9 + a^4*d^10)*(d*tan(f*x + e) + c)^2))/f
```

maple [B] time = 0.62, size = 3364, normalized size = 3.91

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e))^3,x
```

```
)

[Out] 2/f*d^4/(a*d-b*c)^3/(c^2+d^2)^2/(c+d*tan(f*x+e))*C*a*c+2/f*d/(a*d-b*c)^3/(c
^2+d^2)^2/(c+d*tan(f*x+e))*C*b*c^4+3/f*d^5/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*t
an(f*x+e))*A*a^2*c^2-3/f*d^5/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*a
^2*c^2+3/f*d/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*b^2*c^6-1/f*d^3/(
a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*b^2*c^4-5/f*b^4/(a*d-b*c)^4/(a^
2+b^2)^2*ln(a+b*tan(f*x+e))*A*a^2*d-3/f/(a^2+b^2)^2/(c^2+d^2)^3*A*arctan(ta
n(f*x+e))*a^2*c*d^2+2/f/(a^2+b^2)^2/(c^2+d^2)^3*A*arctan(tan(f*x+e))*a*b*d^
3+3/f/(a^2+b^2)^2/(c^2+d^2)^3*A*arctan(tan(f*x+e))*b^2*c*d^2+3/f/(a^2+b^2)^
2/(c^2+d^2)^3*B*arctan(tan(f*x+e))*a^2*c^2*d-1/f/(a^2+b^2)^2/(c^2+d^2)^3*ln
(1+tan(f*x+e)^2)*A*a*b*c^3+3/2/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)
*A*b^2*c^2*d-3/2/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a^2*c*d^2+1
/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a*b*d^3+10/f*d^3/(a*d-b*c)^
4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*b^2*c^4+9/f*d^5/(a*d-b*c)^4/(c^2+d^2)^3*
ln(c+d*tan(f*x+e))*A*b^2*c^2-1/f*d^4/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x
+e))*B*a^2*c^3-1/f*b^4/(a*d-b*c)^4/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*C*a^2*d-2
/f*b^5/(a*d-b*c)^4/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*C*a*c-2/f*d^4/(a*d-b*c)^3
/(c^2+d^2)^2/(c+d*tan(f*x+e))*A*a*c+4/f*d^3/(a*d-b*c)^3/(c^2+d^2)^2/(c+d*ta
n(f*x+e))*A*b*c^2+1/f*d^3/(a*d-b*c)^3/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*a*c^2-
3/f*d^2/(a*d-b*c)^3/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*b*c^3-1/f*d^4/(a*d-b*c)^
3/(c^2+d^2)^2/(c+d*tan(f*x+e))*B*b*c-3/2/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan
(f*x+e)^2)*A*a^2*c^2*d+3/f/(a^2+b^2)^2/(c^2+d^2)^3*C*arctan(tan(f*x+e))*a^2
*c*d^2-2/f/(a^2+b^2)^2/(c^2+d^2)^3*C*arctan(tan(f*x+e))*a*b*d^3-3/f/(a^2+b^
2)^2/(c^2+d^2)^3*C*arctan(tan(f*x+e))*b^2*c*d^2+2/f*b^5/(a*d-b*c)^4/(a^2+b^
2)^2*ln(a+b*tan(f*x+e))*A*a*c+4/f*b^3/(a*d-b*c)^4/(a^2+b^2)^2*ln(a+b*tan(f*
x+e))*a^3*B*d-1/f*b^4/(a*d-b*c)^4/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*a^2*c+2/
f*b^5/(a*d-b*c)^4/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*a*d-3/f*b^2/(a*d-b*c)^4/
(a^2+b^2)^2*ln(a+b*tan(f*x+e))*a^4*C*d+3/f*d^6/(a*d-b*c)^4/(c^2+d^2)^3*ln(c
+d*tan(f*x+e))*B*a^2*c-2/f*d^7/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B
*a*b-6/f*d^2/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*b^2*c^5-3/f*d^4/(
a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*b^2*c^3-1/f*d^6/(a*d-b*c)^4/(c^
2+d^2)^3*ln(c+d*tan(f*x+e))*B*b^2*c+2/f/(a^2+b^2)^2/(c^2+d^2)^3*B*arctan(ta
n(f*x+e))*a*b*c^3-3/f/(a^2+b^2)^2/(c^2+d^2)^3*B*arctan(tan(f*x+e))*b^2*c^2*
d-3/2/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*b^2*c^2*d+3/2/f/(a^2+b
^2)^2/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*b^2*c*d^2+3/2/f/(a^2+b^2)^2/(c^2+d^2
)^3*ln(1+tan(f*x+e)^2)*C*a^2*c^2*d+1/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan(f*x
+e)^2)*C*a*b*c^3-3/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*a*b*c*d^2
+6/f/(a^2+b^2)^2/(c^2+d^2)^3*C*arctan(tan(f*x+e))*a*b*c^2*d-3/f/(a^2+b^2)^2
/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a*b*c^2*d-1/2/f/(a^2+b^2)^2/(c^2+d^2)^3*l
n(1+tan(f*x+e)^2)*B*b^2*c^3-1/2/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan(f*x+e)^2
)*C*a^2*d^3-3/f*b^6/(a*d-b*c)^4/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*A*d+1/f*b^6/
(a*d-b*c)^4/(a^2+b^2)^2*ln(a+b*tan(f*x+e))*B*c-1/f*b^3/(a*d-b*c)^3/(a^2+b^2
)/(a+b*tan(f*x+e))*B*a+1/f*b^2/(a*d-b*c)^3/(a^2+b^2)/(a+b*tan(f*x+e))*a^2*C
+2/f*d^5/(a*d-b*c)^3/(c^2+d^2)^2/(c+d*tan(f*x+e))*A*b-1/f*d^5/(a*d-b*c)^3/(
```

```
c^2+d^2)^2/(c+d*tan(f*x+e))*B*a-1/f*d^7/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(
f*x+e))*A*a^2+3/f*d^7/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*b^2+1/2/
f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*C*b^2*d^3+1/f/(a^2+b^2)^2/(c^2
+d^2)^3*A*arctan(tan(f*x+e))*a^2*c^3-1/f/(a^2+b^2)^2/(c^2+d^2)^3*A*arctan(t
an(f*x+e))*b^2*c^3-1/f/(a^2+b^2)^2/(c^2+d^2)^3*B*arctan(tan(f*x+e))*a^2*d^3
+1/f/(a^2+b^2)^2/(c^2+d^2)^3*B*arctan(tan(f*x+e))*b^2*d^3-1/f/(a^2+b^2)^2/(
c^2+d^2)^3*C*arctan(tan(f*x+e))*a^2*c^3+1/f/(a^2+b^2)^2/(c^2+d^2)^3*C*arcta
n(tan(f*x+e))*b^2*c^3+1/f*d^7/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*
a^2+1/2/f*d^2/(a*d-b*c)^2/(c^2+d^2)/(c+d*tan(f*x+e))^2*B*c-1/2/f*d/(a*d-b*c
)^2/(c^2+d^2)/(c+d*tan(f*x+e))^2*c^2*C+1/2/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+t
an(f*x+e)^2)*A*a^2*d^3-1/2/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*A*b
^2*d^3+1/2/f/(a^2+b^2)^2/(c^2+d^2)^3*ln(1+tan(f*x+e)^2)*B*a^2*c^3-2/f*d^6/(
a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*A*a*b*c+4/f*d^3/(a*d-b*c)^4/(c^2+
d^2)^3*ln(c+d*tan(f*x+e))*B*a*b*c^4+2/f*d^6/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*
tan(f*x+e))*C*a*b*c-6/f*d^5/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*B*a*
b*c^2+10/f*d^4/(a*d-b*c)^4/(c^2+d^2)^3*ln(c+d*tan(f*x+e))*C*a*b*c^3-6/f/(a^
2+b^2)^2/(c^2+d^2)^3*B*arctan(tan(f*x+e))*a*b*c*d^2-10/f*d^4/(a*d-b*c)^4/(c
^2+d^2)^3*ln(c+d*tan(f*x+e))*A*a*b*c^3-6/f/(a^2+b^2)^2/(c^2+d^2)^3*A*arctan
(tan(f*x+e))*a*b*c^2*d-1/2/f*d^3/(a*d-b*c)^2/(c^2+d^2)/(c+d*tan(f*x+e))^2*A
+1/f*b^4/(a*d-b*c)^3/(a^2+b^2)/(a+b*tan(f*x+e))*A+3/f/(a^2+b^2)^2/(c^2+d^2)
^3*ln(1+tan(f*x+e)^2)*A*a*b*c*d^2
```

maxima [B] time = 0.79, size = 2537, normalized size = 2.95

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] 1/2*(2*(((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c^3 + 3*(B*a^2 - 2*(A - C)*a*
b - B*b^2)*c^2*d - 3*((A - C)*a^2 + 2*B*a*b - (A - C)*b^2)*c*d^2 - (B*a^2 -
 2*(A - C)*a*b - B*b^2)*d^3)*(f*x + e)/((a^4 + 2*a^2*b^2 + b^4)*c^6 + 3*(a^
4 + 2*a^2*b^2 + b^4)*c^4*d^2 + 3*(a^4 + 2*a^2*b^2 + b^4)*c^2*d^4 + (a^4 + 2
*a^2*b^2 + b^4)*d^6) - 2*((B*a^2*b^4 - 2*(A - C)*a*b^5 - B*b^6)*c + (3*C*a^
4*b^2 - 4*B*a^3*b^3 + (5*A + C)*a^2*b^4 - 2*B*a*b^5 + 3*A*b^6)*d)*log(b*tan
(f*x + e) + a)/((a^4*b^4 + 2*a^2*b^6 + b^8)*c^4 - 4*(a^5*b^3 + 2*a^3*b^5 +
a*b^7)*c^3*d + 6*(a^6*b^2 + 2*a^4*b^4 + a^2*b^6)*c^2*d^2 - 4*(a^7*b + 2*a^5
*b^3 + a^3*b^5)*c*d^3 + (a^8 + 2*a^6*b^2 + a^4*b^4)*d^4) + 2*(3*C*b^2*c^6*d
 - 6*B*b^2*c^5*d^2 + (4*B*a*b + (10*A - C)*b^2)*c^4*d^3 - (B*a^2 + 10*(A -
C)*a*b + 3*B*b^2)*c^3*d^4 + 3*((A - C)*a^2 - 2*B*a*b + 3*A*b^2)*c^2*d^5 + (
3*B*a^2 - 2*(A - C)*a*b - B*b^2)*c*d^6 - ((A - C)*a^2 + 2*B*a*b - 3*A*b^2)*
d^7)*log(d*tan(f*x + e) + c)/(b^4*c^10 - 4*a*b^3*c^9*d - 4*a^3*b*c*d^9 + a^
4*d^10 + 3*(2*a^2*b^2 + b^4)*c^8*d^2 - 4*(a^3*b + 3*a*b^3)*c^7*d^3 + (a^4 +
 18*a^2*b^2 + 3*b^4)*c^6*d^4 - 12*(a^3*b + a*b^3)*c^5*d^5 + (3*a^4 + 18*a^2
```

```
*b^2 + b^4)*c^4*d^6 - 4*(3*a^3*b + a*b^3)*c^3*d^7 + 3*(a^4 + 2*a^2*b^2)*c^2
*d^8) + ((B*a^2 - 2*(A - C)*a*b - B*b^2)*c^3 - 3*((A - C)*a^2 + 2*B*a*b - (
A - C)*b^2)*c^2*d - 3*(B*a^2 - 2*(A - C)*a*b - B*b^2)*c*d^2 + ((A - C)*a^2
+ 2*B*a*b - (A - C)*b^2)*d^3)*log(tan(f*x + e)^2 + 1)/((a^4 + 2*a^2*b^2 + b
^4)*c^6 + 3*(a^4 + 2*a^2*b^2 + b^4)*c^4*d^2 + 3*(a^4 + 2*a^2*b^2 + b^4)*c^2
*d^4 + (a^4 + 2*a^2*b^2 + b^4)*d^6) - (2*(C*a^2*b^2 - B*a*b^3 + A*b^4)*c^6
+ 5*(C*a^3*b + C*a*b^3)*c^5*d - (C*a^4 + 7*B*a^3*b - 3*C*a^2*b^2 + 11*B*a*b
^3 - 4*A*b^4)*c^4*d^2 + (3*B*a^4 + (9*A + C)*a^3*b + 3*B*a^2*b^2 + (9*A + C
)*a*b^3)*c^3*d^3 - ((5*A - 3*C)*a^4 + 3*B*a^3*b + 5*(A - C)*a^2*b^2 + 5*B*a
*b^3 - 2*A*b^4)*c^2*d^4 - (B*a^4 - 5*A*a^3*b + B*a^2*b^2 - 5*A*a*b^3)*c*d^5
 - (A*a^4 + A*a^2*b^2)*d^6 + 2*((3*C*a^2*b^2 - B*a*b^3 + (A + 2*C)*b^4)*c^4
*d^2 - 3*(B*a^2*b^2 + B*b^4)*c^3*d^3 + (B*a^3*b + 2*(2*A + C)*a^2*b^2 - B*a
*b^3 + 6*A*b^4)*c^2*d^4 - (2*(A - C)*a^3*b + B*a^2*b^2 + 2*(A - C)*a*b^3 +
B*b^4)*c*d^5 - (B*a^3*b - (2*A + C)*a^2*b^2 + 2*B*a*b^3 - 3*A*b^4)*d^6)*tan
(f*x + e)^2 + ((9*C*a^2*b^2 - 4*B*a*b^3 + (4*A + 5*C)*b^4)*c^5*d + (3*C*a^3
*b - 7*B*a^2*b^2 + 3*C*a*b^3 - 7*B*b^4)*c^4*d^2 - (3*B*a^3*b - 9*(A + C)*a^
2*b^2 + 11*B*a*b^3 - (17*A + C)*b^4)*c^3*d^3 + (2*B*a^4 + 3*(A + C)*a^3*b -
 B*a^2*b^2 + 3*(A + C)*a*b^3 - 3*B*b^4)*c^2*d^4 - (4*(A - C)*a^4 + 3*B*a^3*
b - (A + 8*C)*a^2*b^2 + 7*B*a*b^3 - 9*A*b^4)*c*d^5 - (2*B*a^4 - 3*A*a^3*b +
 2*B*a^2*b^2 - 3*A*a*b^3)*d^6)*tan(f*x + e))/((a^3*b^3 + a*b^5)*c^9 - 3*(a^
4*b^2 + a^2*b^4)*c^8*d + (3*a^5*b + 5*a^3*b^3 + 2*a*b^5)*c^7*d^2 - (a^6 + 7
*a^4*b^2 + 6*a^2*b^4)*c^6*d^3 + (6*a^5*b + 7*a^3*b^3 + a*b^5)*c^5*d^4 - (2*
a^6 + 5*a^4*b^2 + 3*a^2*b^4)*c^4*d^5 + 3*(a^5*b + a^3*b^3)*c^3*d^6 - (a^6 +
 a^4*b^2)*c^2*d^7 + ((a^2*b^4 + b^6)*c^7*d^2 - 3*(a^3*b^3 + a*b^5)*c^6*d^3
+ (3*a^4*b^2 + 5*a^2*b^4 + 2*b^6)*c^5*d^4 - (a^5*b + 7*a^3*b^3 + 6*a*b^5)*c
^4*d^5 + (6*a^4*b^2 + 7*a^2*b^4 + b^6)*c^3*d^6 - (2*a^5*b + 5*a^3*b^3 + 3*a
*b^5)*c^2*d^7 + 3*(a^4*b^2 + a^2*b^4)*c*d^8 - (a^5*b + a^3*b^3)*d^9)*tan(f*
x + e)^3 + (2*(a^2*b^4 + b^6)*c^8*d - 5*(a^3*b^3 + a*b^5)*c^7*d^2 + (3*a^4*
b^2 + 7*a^2*b^4 + 4*b^6)*c^6*d^3 + (a^5*b - 9*a^3*b^3 - 10*a*b^5)*c^5*d^4 -
 (a^6 - 5*a^4*b^2 - 8*a^2*b^4 - 2*b^6)*c^4*d^5 + (2*a^5*b - 3*a^3*b^3 - 5*a
*b^5)*c^3*d^6 - (2*a^6 - a^4*b^2 - 3*a^2*b^4)*c^2*d^7 + (a^5*b + a^3*b^3)*c
*d^8 - (a^6 + a^4*b^2)*d^9)*tan(f*x + e)^2 + ((a^2*b^4 + b^6)*c^9 - (a^3*b^
3 + a*b^5)*c^8*d - (3*a^4*b^2 + a^2*b^4 - 2*b^6)*c^7*d^2 + (5*a^5*b + 3*a^3
*b^3 - 2*a*b^5)*c^6*d^3 - (2*a^6 + 8*a^4*b^2 + 5*a^2*b^4 - b^6)*c^5*d^4 + (
10*a^5*b + 9*a^3*b^3 - a*b^5)*c^4*d^5 - (4*a^6 + 7*a^4*b^2 + 3*a^2*b^4)*c^3
*d^6 + 5*(a^5*b + a^3*b^3)*c^2*d^7 - 2*(a^6 + a^4*b^2)*c*d^8)*tan(f*x + e))
)/f
```

mupad [B] time = 47.93, size = 128666, normalized size = 149.44

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))^3),x)
```

```
[Out] (((2*A*b^4*c^6 - A*a^4*d^6 - 2*B*a*b^3*c^6 - B*a^4*c*d^5 - A*a^2*b^2*d^6 -
5*A*a^4*c^2*d^4 + 2*C*a^2*b^2*c^6 + 2*A*b^4*c^2*d^4 + 4*A*b^4*c^4*d^2 + 3*B
*a^4*c^3*d^3 + 3*C*a^4*c^2*d^4 - C*a^4*c^4*d^2 + 9*A*a*b^3*c^3*d^3 + 9*A*a^
3*b*c^3*d^3 - 5*B*a*b^3*c^2*d^4 - 11*B*a*b^3*c^4*d^2 - B*a^2*b^2*c*d^5 - 3*
B*a^3*b*c^2*d^4 - 7*B*a^3*b*c^4*d^2 + C*a*b^3*c^3*d^3 + C*a^3*b*c^3*d^3 - 5
*A*a^2*b^2*c^2*d^4 + 3*B*a^2*b^2*c^3*d^3 + 5*C*a^2*b^2*c^2*d^4 + 3*C*a^2*b^
2*c^4*d^2 + 5*A*a*b^3*c*d^5 + 5*A*a^3*b*c*d^5 + 5*C*a*b^3*c^5*d + 5*C*a^3*b
*c^5*d)/(2*(a^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^2*c^4 + a
^2*d^4 + b^2*c^4 + b^2*d^4 + 2*a^2*c^2*d^2 + 2*b^2*c^2*d^2)) + (tan(e + f*x
)*(3*A*a*b^3*d^6 - 2*B*a^4*d^6 + 3*A*a^3*b*d^6 - 4*A*a^4*c*d^5 + 9*A*b^4*c*
d^5 + 4*A*b^4*c^5*d + 4*C*a^4*c*d^5 + 5*C*b^4*c^5*d - 2*B*a^2*b^2*d^6 + 17*
A*b^4*c^3*d^3 + 2*B*a^4*c^2*d^4 - 3*B*b^4*c^2*d^4 - 7*B*b^4*c^4*d^2 + C*b^4
*c^3*d^3 + 3*A*a*b^3*c^2*d^4 + A*a^2*b^2*c*d^5 + 3*A*a^3*b*c^2*d^4 - 11*B*a
*b^3*c^3*d^3 - 3*B*a^3*b*c^3*d^3 + 3*C*a*b^3*c^2*d^4 + 3*C*a*b^3*c^4*d^2 +
8*C*a^2*b^2*c*d^5 + 9*C*a^2*b^2*c^5*d + 3*C*a^3*b*c^2*d^4 + 3*C*a^3*b*c^4*d
^2 + 9*A*a^2*b^2*c^3*d^3 - B*a^2*b^2*c^2*d^4 - 7*B*a^2*b^2*c^4*d^2 + 9*C*a^
2*b^2*c^3*d^3 - 7*B*a*b^3*c*d^5 - 4*B*a*b^3*c^5*d - 3*B*a^3*b*c*d^5))/(2*(a
^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^2*c^4 + a^2*d^4 + b^2*
c^4 + b^2*d^4 + 2*a^2*c^2*d^2 + 2*b^2*c^2*d^2)) + (tan(e + f*x)^2*(3*A*b^4*
d^6 - 2*B*a*b^3*d^6 - B*a^3*b*d^6 - B*b^4*c*d^5 + 2*A*a^2*b^2*d^6 + 6*A*b^4
*c^2*d^4 + A*b^4*c^4*d^2 + C*a^2*b^2*d^6 - 3*B*b^4*c^3*d^3 + 2*C*b^4*c^4*d^
2 - B*a*b^3*c^2*d^4 - B*a*b^3*c^4*d^2 - B*a^2*b^2*c*d^5 + B*a^3*b*c^2*d^4 +
 4*A*a^2*b^2*c^2*d^4 - 3*B*a^2*b^2*c^3*d^3 + 2*C*a^2*b^2*c^2*d^4 + 3*C*a^2*
b^2*c^4*d^2 - 2*A*a*b^3*c*d^5 - 2*A*a^3*b*c*d^5 + 2*C*a*b^3*c*d^5 + 2*C*a^3
*b*c*d^5))/((a^3*d^3 - b^3*c^3 + 3*a*b^2*c^2*d - 3*a^2*b*c*d^2)*(a^2*c^4 +
a^2*d^4 + b^2*c^4 + b^2*d^4 + 2*a^2*c^2*d^2 + 2*b^2*c^2*d^2)))/(tan(e + f*x
)*(b*c^2 + 2*a*c*d) + a*c^2 + tan(e + f*x)^2*(a*d^2 + 2*b*c*d) + b*d^2*tan(
e + f*x)^3) + symsum(log((3*A^3*a^3*b^6*d^10 - A^3*a^5*b^4*d^10 + 4*B^3*a^2
*b^7*d^10 + 6*B^3*a^4*b^5*d^10 + 24*A^3*b^9*c^3*d^7 + 27*A^3*b^9*c^5*d^5 +
C^3*a^5*b^4*d^10 + B^3*b^9*c^2*d^8 + 4*B^3*b^9*c^4*d^6 + 7*B^3*b^9*c^6*d^4
+ 9*A^2*B*b^9*d^10 + 9*A^3*b^9*c*d^9 + 26*A^3*a^2*b^7*c^3*d^7 + 31*A^3*a^2*
b^7*c^5*d^5 + 16*A^3*a^3*b^6*c^2*d^8 - 11*A^3*a^3*b^6*c^4*d^6 - 6*A^3*a^4*b
^5*c^3*d^7 + 3*A^3*a^5*b^4*c^2*d^8 + 5*B^3*a^2*b^7*c^2*d^8 - 14*B^3*a^2*b^7
*c^4*d^6 + 9*B^3*a^2*b^7*c^6*d^4 + 28*B^3*a^3*b^6*c^3*d^7 + 19*B^3*a^3*b^6*
c^5*d^5 + 6*B^3*a^4*b^5*c^2*d^8 - 20*B^3*a^4*b^5*c^4*d^6 + 7*B^3*a^5*b^4*c^
3*d^7 + C^3*a^2*b^7*c^3*d^7 - 4*C^3*a^2*b^7*c^5*d^5 - 9*C^3*a^2*b^7*c^7*d^3
 - 7*C^3*a^3*b^6*c^2*d^8 - 28*C^3*a^3*b^6*c^4*d^6 + 3*C^3*a^3*b^6*c^6*d^4 +
 15*C^3*a^4*b^5*c^3*d^7 - 9*C^3*a^4*b^5*c^7*d^3 - 3*C^3*a^5*b^4*c^2*d^8 - 2
4*C^3*a^5*b^4*c^4*d^6 + 6*C^3*a^6*b^3*c^3*d^7 - 12*A*B^2*a*b^8*d^10 - 6*A*B
^2*b^9*c*d^9 - 9*A^2*C*b^9*c*d^9 + 4*B^3*a*b^8*c*d^9 - 17*A*B^2*a^3*b^6*d^1
0 + 3*A*B^2*a^5*b^4*d^10 + 12*A^2*B*a^2*b^7*d^10 - 7*A^2*B*a^4*b^5*d^10 + 3
*A*C^2*a^3*b^6*d^10 - 3*A*C^2*a^5*b^4*d^10 - 6*A^2*C*a^3*b^6*d^10 + 3*A^2*C
*a^5*b^4*d^10 - 20*A*B^2*b^9*c^3*d^7 - 28*A*B^2*b^9*c^5*d^5 + 6*A*B^2*b^9*c
^7*d^3 - B*C^2*a^4*b^5*d^10 + 3*B*C^2*a^6*b^3*d^10 + 21*A^2*B*b^9*c^2*d^8 +
 13*A^2*B*b^9*c^4*d^6 - 27*A^2*B*b^9*c^6*d^4 - 4*B^2*C*a^3*b^6*d^10 - 9*B^2
```

*C*a^5*b^4*d^10 - 3*A*C^2*b^9*c^3*d^7 - 9*A*C^2*b^9*c^7*d^3 - 21*A^2*C*b^9*
c^3*d^7 - 27*A^2*C*b^9*c^5*d^5 + 9*A^2*C*b^9*c^7*d^3 + B*C^2*b^9*c^4*d^6 +
3*B*C^2*b^9*c^8*d^2 - B^2*C*b^9*c^3*d^7 - 2*B^2*C*b^9*c^5*d^5 - 9*B^2*C*b^9
*c^7*d^3 - 3*A^3*a*b^8*c^2*d^8 - 31*A^3*a*b^8*c^4*d^6 - 8*A^3*a*b^8*c^6*d^4
 + 3*A^3*a^2*b^7*c*d^9 - 10*A^3*a^4*b^5*c*d^9 + 11*B^3*a*b^8*c^3*d^7 + 5*B^
3*a*b^8*c^5*d^5 - 6*B^3*a*b^8*c^7*d^3 + B^3*a^3*b^6*c*d^9 - 5*B^3*a^5*b^4*c
*d^9 - 2*C^3*a*b^8*c^4*d^6 - C^3*a*b^8*c^6*d^4 - 3*C^3*a*b^8*c^8*d^2 - 2*C^
3*a^4*b^5*c*d^9 - 6*C^3*a^6*b^3*c*d^9 - 4*A*B^2*a^2*b^7*c^3*d^7 - 77*A*B^2*
a^2*b^7*c^5*d^5 - 6*A*B^2*a^2*b^7*c^7*d^3 - 60*A*B^2*a^3*b^6*c^2*d^8 + 25*A
*B^2*a^3*b^6*c^4*d^6 + 28*A*B^2*a^3*b^6*c^6*d^4 + 44*A*B^2*a^4*b^5*c^3*d^7
 - 17*A*B^2*a^4*b^5*c^5*d^5 - 21*A*B^2*a^5*b^4*c^2*d^8 + 4*A*B^2*a^5*b^4*c^4
*d^6 + 71*A^2*B*a^2*b^7*c^2*d^8 + 86*A^2*B*a^2*b^7*c^4*d^6 - 13*A^2*B*a^2*b
^7*c^6*d^4 - 116*A^2*B*a^3*b^6*c^3*d^7 - 37*A^2*B*a^3*b^6*c^5*d^5 + 16*A^2*
B*a^4*b^5*c^2*d^8 + 35*A^2*B*a^4*b^5*c^4*d^6 - 9*A^2*B*a^5*b^4*c^3*d^7 - 30
*A*C^2*a^2*b^7*c^3*d^7 - 15*A*C^2*a^2*b^7*c^5*d^5 + 30*A*C^2*a^3*b^6*c^2*d^
8 + 45*A*C^2*a^3*b^6*c^4*d^6 - 6*A*C^2*a^3*b^6*c^6*d^4 - 63*A*C^2*a^4*b^5*c
^3*d^7 - 27*A*C^2*a^4*b^5*c^5*d^5 + 9*A*C^2*a^4*b^5*c^7*d^3 + 9*A*C^2*a^5*b
^4*c^2*d^8 + 48*A*C^2*a^5*b^4*c^4*d^6 - 12*A*C^2*a^6*b^3*c^3*d^7 + 3*A^2*C*
a^2*b^7*c^3*d^7 - 12*A^2*C*a^2*b^7*c^5*d^5 + 9*A^2*C*a^2*b^7*c^7*d^3 - 39*A
^2*C*a^3*b^6*c^2*d^8 - 6*A^2*C*a^3*b^6*c^4*d^6 + 3*A^2*C*a^3*b^6*c^6*d^4 +
54*A^2*C*a^4*b^5*c^3*d^7 + 27*A^2*C*a^4*b^5*c^5*d^5 - 9*A^2*C*a^5*b^4*c^2*d
^8 - 24*A^2*C*a^5*b^4*c^4*d^6 + 6*A^2*C*a^6*b^3*c^3*d^7 + 11*B*C^2*a^2*b^7*
c^2*d^8 + 47*B*C^2*a^2*b^7*c^4*d^6 + 17*B*C^2*a^2*b^7*c^6*d^4 - 3*B*C^2*a^2
*b^7*c^8*d^2 + 16*B*C^2*a^3*b^6*c^3*d^7 - 25*B*C^2*a^3*b^6*c^5*d^5 + 12*B*C
^2*a^3*b^6*c^7*d^3 - 17*B*C^2*a^4*b^5*c^2*d^8 + 47*B*C^2*a^4*b^5*c^4*d^6 +
27*B*C^2*a^4*b^5*c^6*d^4 + 39*B*C^2*a^5*b^4*c^3*d^7 - 12*B*C^2*a^5*b^4*c^5*
d^5 - 18*B*C^2*a^6*b^3*c^2*d^8 + 3*B*C^2*a^6*b^3*c^4*d^6 - 35*B^2*C*a^2*b^7
*c^3*d^7 + 26*B^2*C*a^2*b^7*c^5*d^5 + 3*B^2*C*a^2*b^7*c^7*d^3 + 9*B^2*C*a^3
*b^6*c^2*d^8 - 16*B^2*C*a^3*b^6*c^4*d^6 - 37*B^2*C*a^3*b^6*c^6*d^4 - 68*B^2
*C*a^4*b^5*c^3*d^7 - 4*B^2*C*a^4*b^5*c^5*d^5 + 9*B^2*C*a^5*b^4*c^2*d^8 + 14
*B^2*C*a^5*b^4*c^4*d^6 - 6*B^2*C*a^6*b^3*c^3*d^7 + 6*A*B*C*a^2*b^7*d^10 + 1
7*A*B*C*a^4*b^5*d^10 - 3*A*B*C*a^6*b^3*d^10 + 6*A*B*C*b^9*c^2*d^8 + 13*A*B*
C*b^9*c^4*d^6 + 36*A*B*C*b^9*c^6*d^4 - 3*A*B*C*b^9*c^8*d^2 - 24*A^2*B*a*b^8
*c*d^9 - 19*A*B^2*a*b^8*c^2*d^8 + 37*A*B^2*a*b^8*c^4*d^6 + 32*A*B^2*a*b^8*c
^6*d^4 + 11*A*B^2*a^2*b^7*c*d^9 + 25*A*B^2*a^4*b^5*c*d^9 - 81*A^2*B*a*b^8*c
^3*d^7 - 15*A^2*B*a*b^8*c^5*d^5 + 6*A^2*B*a*b^8*c^7*d^3 - 23*A^2*B*a^3*b^6*
c*d^9 + 11*A^2*B*a^5*b^4*c*d^9 - 3*A*C^2*a*b^8*c^2*d^8 - 27*A*C^2*a*b^8*c^4
*d^6 - 6*A*C^2*a*b^8*c^6*d^4 + 6*A*C^2*a*b^8*c^8*d^2 - 15*A*C^2*a^2*b^7*c*d
^9 - 15*A*C^2*a^4*b^5*c*d^9 + 12*A*C^2*a^6*b^3*c*d^9 + 6*A^2*C*a*b^8*c^2*d^
8 + 60*A^2*C*a*b^8*c^4*d^6 + 15*A^2*C*a*b^8*c^6*d^4 - 3*A^2*C*a*b^8*c^8*d^2
 + 12*A^2*C*a^2*b^7*c*d^9 + 27*A^2*C*a^4*b^5*c*d^9 - 6*A^2*C*a^6*b^3*c*d^9
+ 3*B*C^2*a*b^8*c^3*d^7 + 9*B*C^2*a*b^8*c^5*d^5 + 18*B*C^2*a*b^8*c^7*d^3 +
13*B*C^2*a^3*b^6*c*d^9 + 23*B*C^2*a^5*b^4*c*d^9 - 8*B^2*C*a*b^8*c^2*d^8 - 2
8*B^2*C*a*b^8*c^4*d^6 - 29*B^2*C*a*b^8*c^6*d^4 + 3*B^2*C*a*b^8*c^8*d^2 - 14
*B^2*C*a^2*b^7*c*d^9 - 16*B^2*C*a^4*b^5*c*d^9 + 6*B^2*C*a^6*b^3*c*d^9 - 28*

A*B*C*a^2*b^7*c^2*d^8 - 79*A*B*C*a^2*b^7*c^4*d^6 + 14*A*B*C*a^2*b^7*c^6*d^4
 + 3*A*B*C*a^2*b^7*c^8*d^2 + 100*A*B*C*a^3*b^6*c^3*d^7 + 62*A*B*C*a^3*b^6*c
^5*d^5 - 12*A*B*C*a^3*b^6*c^7*d^3 + 28*A*B*C*a^4*b^5*c^2*d^8 - 55*A*B*C*a^4
*b^5*c^4*d^6 - 18*A*B*C*a^4*b^5*c^6*d^4 - 30*A*B*C*a^5*b^4*c^3*d^7 + 12*A*B
*C*a^5*b^4*c^5*d^5 + 18*A*B*C*a^6*b^3*c^2*d^8 - 3*A*B*C*a^6*b^3*c^4*d^6 + 2
4*A*B*C*a*b^8*c*d^9 + 78*A*B*C*a*b^8*c^3*d^7 + 6*A*B*C*a*b^8*c^5*d^5 - 24*A
*B*C*a*b^8*c^7*d^3 + 10*A*B*C*a^3*b^6*c*d^9 - 34*A*B*C*a^5*b^4*c*d^9)/(a^10
*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 + 2*a^8*b^
2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10*c^8*d^6
+ b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^2 - 6*a*b
^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d^3 - 12*a
^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c*d^13 - 2
4*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c^9*d^5 +
15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2*b^8*
c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*c^5*d^9 -
 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 + 15*a^4*b
^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*b^6*c^8*d
^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d^11 - 202
*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*b^5*c
^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4*c^6*d^8
+ 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2 - 68*a^7*
b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*b^3*c^9*d
^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^10 + 98*a
^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^9*c^13*d
- 6*a^9*b*c*d^13) - root(640*a^15*b*c^7*d^13*f^4 + 640*a*b^15*c^13*d^7*f^4
+ 480*a^15*b*c^9*d^11*f^4 + 480*a^15*b*c^5*d^15*f^4 + 480*a*b^15*c^15*d^5*f
^4 + 480*a*b^15*c^11*d^9*f^4 + 192*a^15*b*c^11*d^9*f^4 + 192*a^15*b*c^3*d^1
7*f^4 + 192*a^11*b^5*c*d^19*f^4 + 192*a^5*b^11*c^19*d*f^4 + 192*a*b^15*c^17
*d^3*f^4 + 192*a*b^15*c^9*d^11*f^4 + 128*a^13*b^3*c*d^19*f^4 + 128*a^9*b^7*
c*d^19*f^4 + 128*a^7*b^9*c^19*d*f^4 + 128*a^3*b^13*c^19*d*f^4 + 32*a^15*b*c
^13*d^7*f^4 + 32*a^9*b^7*c^19*d*f^4 + 32*a^7*b^9*c*d^19*f^4 + 32*a*b^15*c^7
*d^13*f^4 + 32*a^15*b*c*d^19*f^4 + 32*a*b^15*c^19*d*f^4 - 47088*a^8*b^8*c^1
0*d^10*f^4 + 42432*a^9*b^7*c^9*d^11*f^4 + 42432*a^7*b^9*c^11*d^9*f^4 + 3932
8*a^9*b^7*c^11*d^9*f^4 + 39328*a^7*b^9*c^9*d^11*f^4 - 36912*a^8*b^8*c^12*d^
8*f^4 - 36912*a^8*b^8*c^8*d^12*f^4 - 34256*a^10*b^6*c^10*d^10*f^4 - 34256*a
^6*b^10*c^10*d^10*f^4 - 31152*a^10*b^6*c^8*d^12*f^4 - 31152*a^6*b^10*c^12*d
^8*f^4 + 28128*a^9*b^7*c^7*d^13*f^4 + 28128*a^7*b^9*c^13*d^7*f^4 + 24160*a^
11*b^5*c^9*d^11*f^4 + 24160*a^5*b^11*c^11*d^9*f^4 - 23088*a^10*b^6*c^12*d^8
*f^4 - 23088*a^6*b^10*c^8*d^12*f^4 + 22272*a^9*b^7*c^13*d^7*f^4 + 22272*a^7
*b^9*c^7*d^13*f^4 + 19072*a^11*b^5*c^11*d^9*f^4 + 19072*a^5*b^11*c^9*d^11*f
^4 + 18624*a^11*b^5*c^7*d^13*f^4 + 18624*a^5*b^11*c^13*d^7*f^4 - 17328*a^8*
b^8*c^14*d^6*f^4 - 17328*a^8*b^8*c^6*d^14*f^4 - 17232*a^10*b^6*c^6*d^14*f^4
 - 17232*a^6*b^10*c^14*d^6*f^4 - 13520*a^12*b^4*c^8*d^12*f^4 - 13520*a^4*b^
12*c^12*d^8*f^4 - 12464*a^12*b^4*c^10*d^10*f^4 - 12464*a^4*b^12*c^10*d^10*f
^4 + 10880*a^9*b^7*c^5*d^15*f^4 + 10880*a^7*b^9*c^15*d^5*f^4 - 9072*a^10*b^

6*c^14*d^6*f^4 - 9072*a^6*b^10*c^6*d^14*f^4 + 8928*a^11*b^5*c^13*d^7*f^4 + 8928*a^5*b^11*c^7*d^13*f^4 - 8880*a^12*b^4*c^6*d^14*f^4 - 8880*a^4*b^12*c^14*d^6*f^4 + 8480*a^11*b^5*c^5*d^15*f^4 + 8480*a^5*b^11*c^15*d^5*f^4 + 7200*a^9*b^7*c^15*d^5*f^4 + 7200*a^7*b^9*c^5*d^15*f^4 - 6912*a^12*b^4*c^12*d^8*f^4 - 6912*a^4*b^12*c^8*d^12*f^4 + 6400*a^13*b^3*c^9*d^11*f^4 + 6400*a^3*b^13*c^11*d^9*f^4 + 5920*a^13*b^3*c^7*d^13*f^4 + 5920*a^3*b^13*c^13*d^7*f^4 - 5392*a^10*b^6*c^4*d^16*f^4 - 5392*a^6*b^10*c^16*d^4*f^4 - 4428*a^8*b^8*c^16*d^4*f^4 - 4428*a^8*b^8*c^4*d^16*f^4 + 4128*a^13*b^3*c^11*d^9*f^4 + 4128*a^3*b^13*c^9*d^11*f^4 - 3328*a^12*b^4*c^4*d^16*f^4 - 3328*a^4*b^12*c^16*d^4*f^4 + 3264*a^13*b^3*c^5*d^15*f^4 + 3264*a^3*b^13*c^15*d^5*f^4 - 2480*a^14*b^2*c^8*d^12*f^4 - 2480*a^2*b^14*c^12*d^8*f^4 + 2240*a^11*b^5*c^15*d^5*f^4 + 2240*a^5*b^11*c^5*d^15*f^4 - 2128*a^12*b^4*c^14*d^6*f^4 - 2128*a^4*b^12*c^6*d^14*f^4 + 2112*a^9*b^7*c^3*d^17*f^4 + 2112*a^7*b^9*c^17*d^3*f^4 + 2048*a^11*b^5*c^3*d^17*f^4 + 2048*a^5*b^11*c^17*d^3*f^4 - 2000*a^14*b^2*c^6*d^14*f^4 - 2000*a^2*b^14*c^14*d^6*f^4 - 1792*a^10*b^6*c^16*d^4*f^4 - 1792*a^6*b^10*c^4*d^16*f^4 - 1776*a^14*b^2*c^10*d^10*f^4 - 1776*a^2*b^14*c^10*d^10*f^4 + 1472*a^13*b^3*c^13*d^7*f^4 + 1472*a^3*b^13*c^7*d^13*f^4 + 1088*a^9*b^7*c^17*d^3*f^4 + 1088*a^7*b^9*c^3*d^17*f^4 + 992*a^13*b^3*c^3*d^17*f^4 + 992*a^3*b^13*c^17*d^3*f^4 - 912*a^14*b^2*c^4*d^16*f^4 - 912*a^2*b^14*c^16*d^4*f^4 - 768*a^10*b^6*c^2*d^18*f^4 - 768*a^6*b^10*c^18*d^2*f^4 - 688*a^14*b^2*c^12*d^8*f^4 - 688*a^2*b^14*c^8*d^12*f^4 - 592*a^12*b^4*c^2*d^18*f^4 - 592*a^4*b^12*c^18*d^2*f^4 - 472*a^8*b^8*c^18*d^2*f^4 - 472*a^8*b^8*c^2*d^18*f^4 - 280*a^12*b^4*c^16*d^4*f^4 - 280*a^4*b^12*c^4*d^16*f^4 + 224*a^13*b^3*c^15*d^5*f^4 + 224*a^11*b^5*c^17*d^3*f^4 + 224*a^5*b^11*c^3*d^17*f^4 + 224*a^3*b^13*c^5*d^15*f^4 - 208*a^14*b^2*c^2*d^18*f^4 - 208*a^2*b^14*c^18*d^2*f^4 - 112*a^14*b^2*c^14*d^6*f^4 - 112*a^10*b^6*c^18*d^2*f^4 - 112*a^6*b^10*c^2*d^18*f^4 - 112*a^2*b^14*c^6*d^14*f^4 - 80*b^16*c^14*d^6*f^4 - 60*b^16*c^16*d^4*f^4 - 60*b^16*c^12*d^8*f^4 - 24*b^16*c^18*d^2*f^4 - 24*b^16*c^10*d^10*f^4 - 4*b^16*c^8*d^12*f^4 - 80*a^16*c^6*d^14*f^4 - 60*a^16*c^8*d^12*f^4 - 60*a^16*c^4*d^16*f^4 - 24*a^16*c^10*d^10*f^4 - 24*a^16*c^2*d^18*f^4 - 4*a^16*c^12*d^8*f^4 - 24*a^12*b^4*d^20*f^4 - 16*a^14*b^2*d^20*f^4 - 16*a^10*b^6*d^20*f^4 - 4*a^8*b^8*d^20*f^4 - 24*a^4*b^12*c^20*f^4 - 16*a^6*b^10*c^20*f^4 - 16*a^2*b^14*c^20*f^4 - 4*a^8*b^8*c^20*f^4 - 4*b^16*c^20*f^4 - 4*a^16*d^20*f^4 + 56*A*C*a*b^11*c^13*d*f^2 - 48*A*C*a^11*b*c*d^13*f^2 + 48*A*C*a*b^11*c*d^13*f^2 + 5904*B*C*a^6*b^6*c^7*d^7*f^2 - 5016*B*C*a^5*b^7*c^8*d^6*f^2 - 4608*B*C*a^7*b^5*c^6*d^8*f^2 - 4512*B*C*a^5*b^7*c^6*d^8*f^2 - 4384*B*C*a^7*b^5*c^8*d^6*f^2 + 3056*B*C*a^8*b^4*c^7*d^7*f^2 + 2256*B*C*a^4*b^8*c^7*d^7*f^2 - 1824*B*C*a^3*b^9*c^8*d^6*f^2 + 1632*B*C*a^9*b^3*c^4*d^10*f^2 - 1400*B*C*a^8*b^4*c^3*d^11*f^2 - 1320*B*C*a^4*b^8*c^11*d^3*f^2 - 1248*B*C*a^3*b^9*c^6*d^8*f^2 + 1152*B*C*a^3*b^9*c^10*d^4*f^2 - 1072*B*C*a^9*b^3*c^6*d^8*f^2 + 1068*B*C*a^6*b^6*c^9*d^5*f^2 - 1004*B*C*a^4*b^8*c^5*d^9*f^2 - 968*B*C*a^6*b^6*c^3*d^11*f^2 - 864*B*C*a^8*b^4*c^5*d^9*f^2 - 828*B*C*a^4*b^8*c^9*d^5*f^2 - 792*B*C*a^4*b^8*c^3*d^11*f^2 - 792*B*C*a^2*b^10*c^11*d^3*f^2 - 776*B*C*a^9*b^3*c^8*d^6*f^2 + 688*B*C*a^7*b^5*c^4*d^10*f^2 - 672*B*C*a^10*b^2*c^3*d^11*f^2 - 592*B*C*a^2*b^10*c^9*d^5*f^2 + 544*B*C*a^10*b^2*c^7*d^7*f^2 - 492*B*C

*a^2*b^10*c^5*d^9*f^2 + 480*B*C*a^5*b^7*c^10*d^4*f^2 - 392*B*C*a^10*b^2*c^5
*d^9*f^2 + 332*B*C*a^8*b^4*c^9*d^5*f^2 - 328*B*C*a^6*b^6*c^11*d^3*f^2 + 320
*B*C*a^9*b^3*c^2*d^12*f^2 + 272*B*C*a^3*b^9*c^12*d^2*f^2 - 248*B*C*a^5*b^7*
c^4*d^10*f^2 - 248*B*C*a^2*b^10*c^3*d^11*f^2 - 208*B*C*a^7*b^5*c^10*d^4*f^2
 - 192*B*C*a^5*b^7*c^2*d^12*f^2 + 144*B*C*a^2*b^10*c^7*d^7*f^2 - 96*B*C*a^3
*b^9*c^4*d^10*f^2 + 88*B*C*a^5*b^7*c^12*d^2*f^2 - 72*B*C*a^8*b^4*c^11*d^3*f
^2 + 48*B*C*a^9*b^3*c^10*d^4*f^2 - 48*B*C*a^7*b^5*c^12*d^2*f^2 - 48*B*C*a^7
*b^5*c^2*d^12*f^2 - 48*B*C*a^3*b^9*c^2*d^12*f^2 - 12*B*C*a^10*b^2*c^9*d^5*f
^2 + 4*B*C*a^6*b^6*c^5*d^9*f^2 + 5824*A*C*a^7*b^5*c^5*d^9*f^2 - 4378*A*C*a^
8*b^4*c^6*d^8*f^2 + 4296*A*C*a^5*b^7*c^5*d^9*f^2 - 3912*A*C*a^6*b^6*c^6*d^8
*f^2 - 3672*A*C*a^5*b^7*c^9*d^5*f^2 + 3594*A*C*a^4*b^8*c^8*d^6*f^2 + 3236*A
*C*a^6*b^6*c^8*d^6*f^2 + 2816*A*C*a^9*b^3*c^5*d^9*f^2 + 2624*A*C*a^3*b^9*c^
5*d^9*f^2 + 2432*A*C*a^7*b^5*c^7*d^7*f^2 - 2366*A*C*a^8*b^4*c^4*d^10*f^2 +
2298*A*C*a^4*b^8*c^10*d^4*f^2 + 1872*A*C*a^3*b^9*c^7*d^7*f^2 + 1848*A*C*a^6
*b^6*c^10*d^4*f^2 - 1644*A*C*a^6*b^6*c^4*d^10*f^2 - 1488*A*C*a^7*b^5*c^9*d^
5*f^2 - 1408*A*C*a^3*b^9*c^9*d^5*f^2 - 1308*A*C*a^4*b^8*c^6*d^8*f^2 + 1248*
A*C*a^5*b^7*c^7*d^7*f^2 - 1012*A*C*a^10*b^2*c^6*d^8*f^2 + 1008*A*C*a^7*b^5*
c^3*d^11*f^2 + 992*A*C*a^5*b^7*c^3*d^11*f^2 + 928*A*C*a^3*b^9*c^3*d^11*f^2
+ 848*A*C*a^9*b^3*c^7*d^7*f^2 + 636*A*C*a^2*b^10*c^8*d^6*f^2 - 628*A*C*a^10
*b^2*c^4*d^10*f^2 - 600*A*C*a^2*b^10*c^6*d^8*f^2 - 576*A*C*a^5*b^7*c^11*d^3
*f^2 + 572*A*C*a^2*b^10*c^10*d^4*f^2 + 464*A*C*a^8*b^4*c^8*d^6*f^2 + 304*A*
C*a^6*b^6*c^2*d^12*f^2 - 304*A*C*a^4*b^8*c^4*d^10*f^2 + 296*A*C*a^4*b^8*c^2
*d^12*f^2 + 260*A*C*a^8*b^4*c^10*d^4*f^2 - 232*A*C*a^9*b^3*c^9*d^5*f^2 - 23
2*A*C*a^2*b^10*c^12*d^2*f^2 + 228*A*C*a^10*b^2*c^2*d^12*f^2 - 188*A*C*a^2*b
^10*c^4*d^10*f^2 + 144*A*C*a^3*b^9*c^11*d^3*f^2 + 116*A*C*a^6*b^6*c^12*d^2*
f^2 + 112*A*C*a^9*b^3*c^3*d^11*f^2 - 112*A*C*a^7*b^5*c^11*d^3*f^2 + 92*A*C*
a^10*b^2*c^8*d^6*f^2 + 74*A*C*a^4*b^8*c^12*d^2*f^2 + 62*A*C*a^8*b^4*c^2*d^1
2*f^2 + 40*A*C*a^2*b^10*c^2*d^12*f^2 - 7008*A*B*a^6*b^6*c^7*d^7*f^2 - 4032*
A*B*a^4*b^8*c^7*d^7*f^2 + 3952*A*B*a^7*b^5*c^8*d^6*f^2 + 3648*A*B*a^5*b^7*c
^8*d^6*f^2 - 3392*A*B*a^8*b^4*c^7*d^7*f^2 + 3264*A*B*a^7*b^5*c^6*d^8*f^2 -
2992*A*B*a^5*b^7*c^4*d^10*f^2 - 2368*A*B*a^7*b^5*c^4*d^10*f^2 - 2304*A*B*a^
3*b^9*c^4*d^10*f^2 - 1968*A*B*a^6*b^6*c^9*d^5*f^2 - 1872*A*B*a^9*b^3*c^4*d^
10*f^2 - 1728*A*B*a^2*b^10*c^7*d^7*f^2 + 1712*A*B*a^8*b^4*c^3*d^11*f^2 + 15
36*A*B*a^5*b^7*c^6*d^8*f^2 - 1536*A*B*a^3*b^9*c^10*d^4*f^2 - 1392*A*B*a^5*b
^7*c^2*d^12*f^2 + 1328*A*B*a^6*b^6*c^3*d^11*f^2 - 1104*A*B*a^3*b^9*c^2*d^12
*f^2 - 1056*A*B*a^3*b^9*c^6*d^8*f^2 + 976*A*B*a^9*b^3*c^6*d^8*f^2 + 960*A*B
*a^4*b^8*c^11*d^3*f^2 + 936*A*B*a^8*b^4*c^5*d^9*f^2 - 912*A*B*a^5*b^7*c^10*
d^4*f^2 + 848*A*B*a^9*b^3*c^8*d^6*f^2 - 816*A*B*a^7*b^5*c^2*d^12*f^2 + 816*
A*B*a^4*b^8*c^3*d^11*f^2 + 768*A*B*a^10*b^2*c^3*d^11*f^2 + 672*A*B*a^3*b^9*
c^8*d^6*f^2 - 632*A*B*a^8*b^4*c^9*d^5*f^2 - 608*A*B*a^2*b^10*c^9*d^5*f^2 -
552*A*B*a^4*b^8*c^9*d^5*f^2 - 544*A*B*a^10*b^2*c^7*d^7*f^2 - 480*A*B*a^2*b^
10*c^5*d^9*f^2 + 464*A*B*a^10*b^2*c^5*d^9*f^2 - 464*A*B*a^9*b^3*c^2*d^12*f^
2 + 432*A*B*a^2*b^10*c^11*d^3*f^2 - 368*A*B*a^3*b^9*c^12*d^2*f^2 - 256*A*B*
a^6*b^6*c^5*d^9*f^2 - 208*A*B*a^5*b^7*c^12*d^2*f^2 + 176*A*B*a^4*b^8*c^5*d^
9*f^2 + 112*A*B*a^7*b^5*c^10*d^4*f^2 + 112*A*B*a^6*b^6*c^11*d^3*f^2 - 16*A*

B*a^2*b^10*c^3*d^11*f^2 - 576*B*C*a*b^11*c^8*d^6*f^2 + 400*B*C*a^11*b*c^4*d
^10*f^2 - 288*B*C*a*b^11*c^6*d^8*f^2 - 176*B*C*a^11*b*c^6*d^8*f^2 + 128*B*C
*a*b^11*c^10*d^4*f^2 - 108*B*C*a^4*b^8*c*d^13*f^2 - 104*B*C*a*b^11*c^4*d^10
*f^2 - 92*B*C*a^4*b^8*c^13*d*f^2 - 60*B*C*a^8*b^4*c*d^13*f^2 - 60*B*C*a^6*b
^6*c*d^13*f^2 + 48*B*C*a^11*b*c^2*d^12*f^2 - 40*B*C*a^2*b^10*c*d^13*f^2 - 2
8*B*C*a^2*b^10*c^13*d*f^2 - 24*B*C*a*b^11*c^12*d^2*f^2 + 20*B*C*a^10*b^2*c*
d^13*f^2 - 16*B*C*a*b^11*c^2*d^12*f^2 + 12*B*C*a^6*b^6*c^13*d*f^2 + 912*A*C
*a*b^11*c^7*d^7*f^2 + 808*A*C*a*b^11*c^5*d^9*f^2 + 432*A*C*a^11*b*c^5*d^9*f
^2 + 336*A*C*a*b^11*c^3*d^11*f^2 + 224*A*C*a*b^11*c^11*d^3*f^2 - 112*A*C*a^
11*b*c^3*d^11*f^2 + 112*A*C*a^3*b^9*c*d^13*f^2 - 88*A*C*a^9*b^3*c*d^13*f^2
+ 80*A*C*a^3*b^9*c^13*d*f^2 + 56*A*C*a^5*b^7*c*d^13*f^2 + 48*A*C*a*b^11*c^9
*d^5*f^2 - 40*A*C*a^5*b^7*c^13*d*f^2 - 16*A*C*a^11*b*c^7*d^7*f^2 + 16*A*C*a
^7*b^5*c*d^13*f^2 - 496*A*B*a*b^11*c^4*d^10*f^2 - 400*A*B*a^11*b*c^4*d^10*f
^2 + 288*A*B*a*b^11*c^8*d^6*f^2 - 288*A*B*a*b^11*c^6*d^8*f^2 - 272*A*B*a*b^
11*c^2*d^12*f^2 + 240*A*B*a^6*b^6*c*d^13*f^2 - 224*A*B*a*b^11*c^10*d^4*f^2
+ 192*A*B*a^8*b^4*c*d^13*f^2 + 192*A*B*a^4*b^8*c*d^13*f^2 + 176*A*B*a^11*b*
c^6*d^8*f^2 + 104*A*B*a^4*b^8*c^13*d*f^2 - 48*A*B*a^11*b*c^2*d^12*f^2 + 16*
A*B*a^10*b^2*c*d^13*f^2 + 16*A*B*a^2*b^10*c^13*d*f^2 + 16*A*B*a^2*b^10*c*d^
13*f^2 - 112*B*C*b^12*c^11*d^3*f^2 + 4*B*C*b^12*c^5*d^9*f^2 + 150*A*C*b^12*
c^10*d^4*f^2 - 80*B*C*a^12*c^3*d^11*f^2 + 66*A*C*b^12*c^8*d^6*f^2 - 30*A*C*
b^12*c^12*d^2*f^2 + 24*B*C*a^12*c^5*d^9*f^2 - 12*A*C*b^12*c^4*d^10*f^2 - 57
6*A*B*b^12*c^7*d^7*f^2 - 432*A*B*b^12*c^9*d^5*f^2 - 400*A*B*b^12*c^5*d^9*f^
2 - 144*A*B*b^12*c^3*d^11*f^2 - 96*B*C*a^7*b^5*d^14*f^2 - 72*B*C*a^5*b^7*d^
14*f^2 - 66*A*C*a^12*c^4*d^10*f^2 + 54*A*C*a^12*c^2*d^12*f^2 - 32*A*B*b^12*
c^11*d^3*f^2 - 24*B*C*a^9*b^3*d^14*f^2 - 16*B*C*a^3*b^9*d^14*f^2 + 2*A*C*a^
12*c^6*d^8*f^2 + 116*A*C*a^6*b^6*d^14*f^2 + 100*A*C*a^4*b^8*d^14*f^2 + 80*A
*B*a^12*c^3*d^11*f^2 + 24*A*C*a^2*b^10*d^14*f^2 - 24*A*B*a^12*c^5*d^9*f^2 +
22*A*C*a^8*b^4*d^14*f^2 + 16*B*C*a^3*b^9*c^14*f^2 + 8*A*C*a^10*b^2*d^14*f^
2 - 192*A*B*a^5*b^7*d^14*f^2 - 176*A*B*a^3*b^9*d^14*f^2 - 48*A*B*a^7*b^5*d^
14*f^2 - 28*A*C*a^2*b^10*c^14*f^2 + 2*A*C*a^4*b^8*c^14*f^2 - 16*A*B*a^3*b^9
*c^14*f^2 + 2508*C^2*a^6*b^6*c^6*d^8*f^2 + 2376*C^2*a^5*b^7*c^9*d^5*f^2 + 2
357*C^2*a^8*b^4*c^6*d^8*f^2 - 2048*C^2*a^7*b^5*c^5*d^9*f^2 + 1304*C^2*a^3*b
^9*c^9*d^5*f^2 + 1303*C^2*a^8*b^4*c^4*d^10*f^2 + 1212*C^2*a^6*b^6*c^4*d^10*
f^2 - 1203*C^2*a^4*b^8*c^8*d^6*f^2 - 1192*C^2*a^9*b^3*c^5*d^9*f^2 + 1062*C^
2*a^4*b^8*c^6*d^8*f^2 + 984*C^2*a^7*b^5*c^9*d^5*f^2 - 952*C^2*a^6*b^6*c^8*d
^6*f^2 + 768*C^2*a^5*b^7*c^7*d^7*f^2 - 681*C^2*a^4*b^8*c^10*d^4*f^2 - 672*C
^2*a^5*b^7*c^5*d^9*f^2 - 480*C^2*a^6*b^6*c^10*d^4*f^2 + 458*C^2*a^10*b^2*c^
6*d^8*f^2 - 448*C^2*a^7*b^5*c^7*d^7*f^2 + 422*C^2*a^4*b^8*c^4*d^10*f^2 + 37
2*C^2*a^2*b^10*c^6*d^8*f^2 + 360*C^2*a^5*b^7*c^11*d^3*f^2 + 312*C^2*a^3*b^9
*c^7*d^7*f^2 + 278*C^2*a^10*b^2*c^4*d^10*f^2 - 232*C^2*a^9*b^3*c^7*d^7*f^2
+ 194*C^2*a^2*b^10*c^12*d^2*f^2 + 176*C^2*a^9*b^3*c^9*d^5*f^2 + 152*C^2*a^5
*b^7*c^3*d^11*f^2 + 124*C^2*a^2*b^10*c^4*d^10*f^2 - 120*C^2*a^7*b^5*c^3*d^1
1*f^2 - 114*C^2*a^10*b^2*c^2*d^12*f^2 - 102*C^2*a^2*b^10*c^8*d^6*f^2 + 101*
C^2*a^4*b^8*c^12*d^2*f^2 + 100*C^2*a^6*b^6*c^2*d^12*f^2 - 88*C^2*a^3*b^9*c^
5*d^9*f^2 + 77*C^2*a^8*b^4*c^2*d^12*f^2 + 72*C^2*a^3*b^9*c^11*d^3*f^2 - 64*

C^2*a^10*b^2*c^8*d^6*f^2 + 64*C^2*a^3*b^9*c^3*d^11*f^2 - 58*C^2*a^2*b^10*c^10*d^4*f^2 + 56*C^2*a^7*b^5*c^11*d^3*f^2 + 56*C^2*a^6*b^6*c^12*d^2*f^2 + 40*C^2*a^9*b^3*c^3*d^11*f^2 + 36*C^2*a^8*b^4*c^12*d^2*f^2 + 32*C^2*a^4*b^8*c^2*d^12*f^2 + 26*C^2*a^8*b^4*c^10*d^4*f^2 + 16*C^2*a^2*b^10*c^2*d^12*f^2 + 2*C^2*a^8*b^4*c^8*d^6*f^2 + 2277*B^2*a^4*b^8*c^8*d^6*f^2 + 2144*B^2*a^7*b^5*c^5*d^9*f^2 - 2112*B^2*a^5*b^7*c^9*d^5*f^2 + 2028*B^2*a^6*b^6*c^8*d^6*f^2 - 1671*B^2*a^8*b^4*c^6*d^8*f^2 + 1275*B^2*a^4*b^8*c^10*d^4*f^2 + 1176*B^2*a^5*b^7*c^5*d^9*f^2 + 1096*B^2*a^9*b^3*c^5*d^9*f^2 - 1044*B^2*a^6*b^6*c^6*d^8*f^2 + 984*B^2*a^6*b^6*c^10*d^4*f^2 - 968*B^2*a^3*b^9*c^9*d^5*f^2 - 888*B^2*a^7*b^5*c^9*d^5*f^2 + 672*B^2*a^7*b^5*c^7*d^7*f^2 + 664*B^2*a^3*b^9*c^5*d^9*f^2 - 649*B^2*a^8*b^4*c^4*d^10*f^2 + 618*B^2*a^2*b^10*c^8*d^6*f^2 + 514*B^2*a^4*b^8*c^4*d^10*f^2 + 460*B^2*a^6*b^6*c^2*d^12*f^2 + 422*B^2*a^8*b^4*c^8*d^6*f^2 + 406*B^2*a^2*b^10*c^10*d^4*f^2 - 382*B^2*a^10*b^2*c^6*d^8*f^2 + 368*B^2*a^4*b^8*c^2*d^12*f^2 - 312*B^2*a^5*b^7*c^11*d^3*f^2 + 312*B^2*a^3*b^9*c^7*d^7*f^2 + 248*B^2*a^9*b^3*c^7*d^7*f^2 + 245*B^2*a^8*b^4*c^2*d^12*f^2 - 192*B^2*a^5*b^7*c^7*d^7*f^2 - 184*B^2*a^9*b^3*c^3*d^11*f^2 + 182*B^2*a^10*b^2*c^2*d^12*f^2 + 176*B^2*a^3*b^9*c^3*d^11*f^2 + 174*B^2*a^4*b^8*c^6*d^8*f^2 - 170*B^2*a^10*b^2*c^4*d^10*f^2 - 152*B^2*a^9*b^3*c^9*d^5*f^2 + 152*B^2*a^2*b^10*c^4*d^10*f^2 + 142*B^2*a^8*b^4*c^10*d^4*f^2 - 90*B^2*a^2*b^10*c^12*d^2*f^2 + 88*B^2*a^2*b^10*c^2*d^12*f^2 + 84*B^2*a^10*b^2*c^8*d^6*f^2 + 84*B^2*a^2*b^10*c^6*d^8*f^2 + 60*B^2*a^6*b^6*c^12*d^2*f^2 - 56*B^2*a^7*b^5*c^11*d^3*f^2 + 53*B^2*a^4*b^8*c^12*d^2*f^2 + 24*B^2*a^7*b^5*c^3*d^11*f^2 + 24*B^2*a^6*b^6*c^4*d^10*f^2 + 24*B^2*a^3*b^9*c^11*d^3*f^2 - 8*B^2*a^5*b^7*c^3*d^11*f^2 + 4566*A^2*a^4*b^8*c^6*d^8*f^2 + 4284*A^2*a^6*b^6*c^6*d^8*f^2 - 3776*A^2*a^7*b^5*c^5*d^9*f^2 - 3624*A^2*a^5*b^7*c^5*d^9*f^2 + 3122*A^2*a^4*b^8*c^4*d^10*f^2 + 3108*A^2*a^2*b^10*c^6*d^8*f^2 + 2741*A^2*a^8*b^4*c^6*d^8*f^2 + 2592*A^2*a^6*b^6*c^4*d^10*f^2 - 2536*A^2*a^3*b^9*c^5*d^9*f^2 + 2224*A^2*a^2*b^10*c^4*d^10*f^2 - 2184*A^2*a^3*b^9*c^7*d^7*f^2 - 2016*A^2*a^5*b^7*c^7*d^7*f^2 - 1984*A^2*a^7*b^5*c^7*d^7*f^2 + 1626*A^2*a^2*b^10*c^8*d^6*f^2 - 1624*A^2*a^9*b^3*c^5*d^9*f^2 + 1603*A^2*a^8*b^4*c^4*d^10*f^2 + 1296*A^2*a^5*b^7*c^9*d^5*f^2 - 1144*A^2*a^5*b^7*c^3*d^11*f^2 - 992*A^2*a^3*b^9*c^3*d^11*f^2 + 968*A^2*a^4*b^8*c^2*d^12*f^2 - 888*A^2*a^7*b^5*c^3*d^11*f^2 + 849*A^2*a^4*b^8*c^8*d^6*f^2 + 808*A^2*a^2*b^10*c^2*d^12*f^2 - 616*A^2*a^9*b^3*c^7*d^7*f^2 + 554*A^2*a^10*b^2*c^6*d^8*f^2 + 504*A^2*a^7*b^5*c^9*d^5*f^2 - 504*A^2*a^6*b^6*c^10*d^4*f^2 + 460*A^2*a^6*b^6*c^2*d^12*f^2 + 350*A^2*a^10*b^2*c^4*d^10*f^2 + 350*A^2*a^2*b^10*c^10*d^4*f^2 - 321*A^2*a^4*b^8*c^10*d^4*f^2 + 216*A^2*a^5*b^7*c^11*d^3*f^2 - 216*A^2*a^3*b^9*c^11*d^3*f^2 + 182*A^2*a^2*b^10*c^12*d^2*f^2 - 152*A^2*a^9*b^3*c^3*d^11*f^2 - 124*A^2*a^6*b^6*c^8*d^6*f^2 - 114*A^2*a^10*b^2*c^2*d^12*f^2 + 104*A^2*a^3*b^9*c^9*d^5*f^2 + 77*A^2*a^8*b^4*c^2*d^12*f^2 + 74*A^2*a^8*b^4*c^8*d^6*f^2 - 70*A^2*a^8*b^4*c^10*d^4*f^2 + 56*A^2*a^9*b^3*c^9*d^5*f^2 + 56*A^2*a^7*b^5*c^11*d^3*f^2 + 41*A^2*a^4*b^8*c^12*d^2*f^2 - 28*A^2*a^10*b^2*c^8*d^6*f^2 - 28*A^2*a^6*b^6*c^12*d^2*f^2 + 12*B*C*b^12*c^13*d*f^2 + 24*B*C*a^12*c*d^13*f^2 - 24*A*B*b^12*c^13*d*f^2 - 24*A*B*b^12*c*d^13*f^2 - 16*B*C*a^11*b*d^14*f^2 - 24*A*B*a^12*c*d^13*f^2 - 16*B*C*a*b^11*c^14*f^2 - 48*A*B*a*b^11*d^14*f^2 + 16*A*B*a^11*b

*d^14*f^2 + 16*A*B*a*b^11*c^14*f^2 - 216*C^2*a^11*b*c^5*d^9*f^2 + 216*C^2*a
*b^11*c^9*d^5*f^2 + 56*C^2*a^11*b*c^3*d^11*f^2 + 56*C^2*a^9*b^3*c*d^13*f^2
+ 56*C^2*a^5*b^7*c*d^13*f^2 + 40*C^2*a^7*b^5*c*d^13*f^2 - 40*C^2*a*b^11*c^1
1*d^3*f^2 + 32*C^2*a^5*b^7*c^13*d*f^2 - 24*C^2*a*b^11*c^7*d^7*f^2 - 16*C^2*
a^3*b^9*c^13*d*f^2 + 16*C^2*a^3*b^9*c*d^13*f^2 + 8*C^2*a^11*b*c^7*d^7*f^2 -
8*C^2*a*b^11*c^5*d^9*f^2 + 264*B^2*a*b^11*c^7*d^7*f^2 + 224*B^2*a*b^11*c^5
*d^9*f^2 + 168*B^2*a^11*b*c^5*d^9*f^2 - 112*B^2*a^9*b^3*c*d^13*f^2 - 104*B^
2*a^11*b*c^3*d^11*f^2 - 104*B^2*a^7*b^5*c*d^13*f^2 + 96*B^2*a*b^11*c^3*d^11
*f^2 + 88*B^2*a*b^11*c^11*d^3*f^2 - 72*B^2*a*b^11*c^9*d^5*f^2 - 64*B^2*a^5*
b^7*c*d^13*f^2 + 32*B^2*a^3*b^9*c^13*d*f^2 - 24*B^2*a^11*b*c^7*d^7*f^2 - 24
*B^2*a^5*b^7*c^13*d*f^2 + 16*B^2*a^3*b^9*c*d^13*f^2 - 888*A^2*a*b^11*c^7*d^
7*f^2 - 800*A^2*a*b^11*c^5*d^9*f^2 - 336*A^2*a*b^11*c^3*d^11*f^2 - 264*A^2*
a*b^11*c^9*d^5*f^2 - 216*A^2*a^11*b*c^5*d^9*f^2 - 184*A^2*a*b^11*c^11*d^3*f
^2 - 128*A^2*a^3*b^9*c*d^13*f^2 - 112*A^2*a^5*b^7*c*d^13*f^2 - 64*A^2*a^3*b
^9*c^13*d*f^2 + 56*A^2*a^11*b*c^3*d^11*f^2 - 56*A^2*a^7*b^5*c*d^13*f^2 + 32
*A^2*a^9*b^3*c*d^13*f^2 + 8*A^2*a^11*b*c^7*d^7*f^2 + 8*A^2*a^5*b^7*c^13*d*f
^2 + 24*C^2*a^11*b*c*d^13*f^2 - 16*C^2*a*b^11*c^13*d*f^2 - 40*B^2*a^11*b*c*
d^13*f^2 + 24*B^2*a*b^11*c^13*d*f^2 + 16*B^2*a*b^11*c*d^13*f^2 - 48*A^2*a*b
^11*c*d^13*f^2 - 40*A^2*a*b^11*c^13*d*f^2 + 24*A^2*a^11*b*c*d^13*f^2 - 6*A*
C*a^12*d^14*f^2 + 2*A*C*b^12*c^14*f^2 + 33*C^2*b^12*c^12*d^2*f^2 - 27*C^2*b
^12*c^10*d^4*f^2 + 3*C^2*b^12*c^8*d^6*f^2 + 117*B^2*b^12*c^10*d^4*f^2 + 111
*B^2*b^12*c^8*d^6*f^2 + 72*B^2*b^12*c^6*d^8*f^2 + 33*C^2*a^12*c^4*d^10*f^2
- 27*C^2*a^12*c^2*d^12*f^2 + 24*B^2*b^12*c^4*d^10*f^2 + 4*B^2*b^12*c^2*d^12
*f^2 - 3*B^2*b^12*c^12*d^2*f^2 - C^2*a^12*c^6*d^8*f^2 + 720*A^2*b^12*c^6*d^
8*f^2 + 552*A^2*b^12*c^4*d^10*f^2 + 471*A^2*b^12*c^8*d^6*f^2 + 216*A^2*b^12
*c^2*d^12*f^2 + 93*A^2*b^12*c^10*d^4*f^2 + 33*B^2*a^12*c^2*d^12*f^2 + 33*A^
2*b^12*c^12*d^2*f^2 + 31*C^2*a^8*b^4*d^14*f^2 - 27*B^2*a^12*c^4*d^10*f^2 +
20*C^2*a^6*b^6*d^14*f^2 + 4*C^2*a^4*b^8*d^14*f^2 + 3*B^2*a^12*c^6*d^8*f^2 +
2*C^2*a^10*b^2*d^14*f^2 + 80*B^2*a^6*b^6*d^14*f^2 + 64*B^2*a^4*b^8*d^14*f^
2 + 33*A^2*a^12*c^4*d^10*f^2 + 31*B^2*a^8*b^4*d^14*f^2 - 27*A^2*a^12*c^2*d^
12*f^2 + 16*B^2*a^2*b^10*d^14*f^2 + 14*C^2*a^2*b^10*c^14*f^2 + 14*B^2*a^10*
b^2*d^14*f^2 - C^2*a^4*b^8*c^14*f^2 - A^2*a^12*c^6*d^8*f^2 + 120*A^2*a^2*b^
10*d^14*f^2 + 112*A^2*a^4*b^8*d^14*f^2 - 17*A^2*a^8*b^4*d^14*f^2 - 10*B^2*a
^2*b^10*c^14*f^2 - 10*A^2*a^10*b^2*d^14*f^2 + 8*A^2*a^6*b^6*d^14*f^2 + 3*B^
2*a^4*b^8*c^14*f^2 + 14*A^2*a^2*b^10*c^14*f^2 - A^2*a^4*b^8*c^14*f^2 + 3*C^
2*a^12*d^14*f^2 - C^2*b^12*c^14*f^2 + 36*A^2*b^12*d^14*f^2 + 3*B^2*b^12*c^1
4*f^2 - B^2*a^12*d^14*f^2 + 3*A^2*a^12*d^14*f^2 - A^2*b^12*c^14*f^2 - 44*A*
B*C*a*b^9*c^10*d*f + 3816*A*B*C*a^5*b^5*c^4*d^7*f + 2920*A*B*C*a^2*b^8*c^5*
d^6*f - 2736*A*B*C*a^3*b^7*c^6*d^5*f - 2672*A*B*C*a^4*b^6*c^3*d^8*f + 1996*
A*B*C*a^4*b^6*c^7*d^4*f - 1412*A*B*C*a^6*b^4*c^5*d^6*f + 1120*A*B*C*a^3*b^7
*c^2*d^9*f + 1080*A*B*C*a^2*b^8*c^7*d^4*f + 1040*A*B*C*a^5*b^5*c^2*d^9*f +
684*A*B*C*a^4*b^6*c^5*d^6*f + 592*A*B*C*a^3*b^7*c^4*d^7*f - 560*A*B*C*a^7*b
^3*c^2*d^9*f - 448*A*B*C*a^2*b^8*c^3*d^8*f - 400*A*B*C*a^5*b^5*c^8*d^3*f -
398*A*B*C*a^2*b^8*c^9*d^2*f - 312*A*B*C*a^6*b^4*c^3*d^8*f + 166*A*B*C*a^8*b
^2*c^3*d^8*f + 136*A*B*C*a^5*b^5*c^6*d^5*f + 128*A*B*C*a^7*b^3*c^6*d^5*f -

100*A*B*C*a^6*b^4*c^7*d^4*f + 64*A*B*C*a^7*b^3*c^4*d^7*f - 64*A*B*C*a^4*b^6
*c^9*d^2*f - 32*A*B*C*a^3*b^7*c^8*d^3*f - 16*A*B*C*a^8*b^2*c^5*d^6*f - 1312
*A*B*C*a*b^9*c^4*d^7*f + 996*A*B*C*a*b^9*c^8*d^3*f + 728*A*B*C*a^6*b^4*c*d^
10*f - 624*A*B*C*a*b^9*c^6*d^5*f - 584*A*B*C*a^2*b^8*c*d^10*f - 512*A*B*C*a
^4*b^6*c*d^10*f - 320*A*B*C*a*b^9*c^2*d^9*f - 98*A*B*C*a^8*b^2*c*d^10*f + 3
6*A*B*C*a^9*b*c^2*d^9*f + 32*A*B*C*a^3*b^7*c^10*d*f - 16*A*B*C*a^9*b*c^4*d^
7*f + 46*B*C^2*a*b^9*c^10*d*f - 16*B^2*C*a*b^9*c*d^10*f - 2*B^2*C*a^9*b*c*d
^10*f + 312*A^2*C*a*b^9*c*d^10*f - 48*A*C^2*a*b^9*c*d^10*f - 6*A^2*C*a^9*b*
c*d^10*f + 6*A*C^2*a^9*b*c*d^10*f + 208*A*B^2*a*b^9*c*d^10*f - 2*A^2*B*a*b^
9*c^10*d*f + 2*A*B^2*a^9*b*c*d^10*f - 480*A*B*C*b^10*c^7*d^4*f + 78*A*B*C*b
^10*c^9*d^2*f - 64*A*B*C*b^10*c^5*d^6*f + 2*A*B*C*a^10*c^3*d^8*f - 224*A*B*
C*a^5*b^5*d^11*f + 80*A*B*C*a^7*b^3*d^11*f - 32*A*B*C*a^3*b^7*d^11*f + 2*A*
B*C*a^2*b^8*c^11*f - 1692*B*C^2*a^5*b^5*c^4*d^7*f - 1500*B^2*C*a^5*b^5*c^5*
d^6*f - 1464*B^2*C*a^3*b^7*c^5*d^6*f + 1426*B*C^2*a^6*b^4*c^5*d^6*f - 1158*
B^2*C*a^6*b^4*c^4*d^7*f + 1152*B*C^2*a^3*b^7*c^6*d^5*f + 1026*B^2*C*a^4*b^6
*c^6*d^5*f - 974*B*C^2*a^4*b^6*c^7*d^4*f + 960*B^2*C*a^5*b^5*c^3*d^8*f - 88
4*B*C^2*a^2*b^8*c^5*d^6*f - 764*B^2*C*a^5*b^5*c^7*d^4*f + 752*B^2*C*a^2*b^8
*c^4*d^7*f - 752*B*C^2*a^3*b^7*c^4*d^7*f + 738*B^2*C*a^4*b^6*c^4*d^7*f - 68
8*B^2*C*a^6*b^4*c^2*d^9*f - 675*B^2*C*a^2*b^8*c^8*d^3*f + 560*B*C^2*a^5*b^5
*c^8*d^3*f + 496*B*C^2*a^7*b^3*c^2*d^9*f + 496*B*C^2*a^4*b^6*c^3*d^8*f - 46
8*B*C^2*a^2*b^8*c^7*d^4*f + 456*B^2*C*a^7*b^3*c^3*d^8*f - 452*B^2*C*a^4*b^6
*c^8*d^3*f - 416*B*C^2*a^3*b^7*c^2*d^9*f + 378*B*C^2*a^4*b^6*c^5*d^6*f + 37
6*B*C^2*a^3*b^7*c^8*d^3*f - 360*B^2*C*a^2*b^8*c^6*d^5*f + 355*B*C^2*a^2*b^8
*c^9*d^2*f + 346*B^2*C*a^6*b^4*c^6*d^5*f - 320*B^2*C*a^4*b^6*c^2*d^9*f + 26
8*B^2*C*a^2*b^8*c^2*d^9*f + 216*B^2*C*a^3*b^7*c^7*d^4*f - 203*B*C^2*a^8*b^2
*c^3*d^8*f - 184*B*C^2*a^7*b^3*c^6*d^5*f + 170*B*C^2*a^6*b^4*c^7*d^4*f + 16
0*B^2*C*a^7*b^3*c^5*d^6*f - 160*B*C^2*a^5*b^5*c^2*d^9*f - 140*B^2*C*a^8*b^2
*c^4*d^7*f - 136*B*C^2*a^2*b^8*c^3*d^8*f + 112*B^2*C*a^3*b^7*c^9*d^2*f + 91
*B^2*C*a^8*b^2*c^2*d^9*f + 88*B*C^2*a^7*b^3*c^4*d^7*f + 72*B^2*C*a^6*b^4*c^
8*d^3*f - 64*B^2*C*a^3*b^7*c^3*d^8*f - 60*B*C^2*a^6*b^4*c^3*d^8*f + 56*B*C^
2*a^4*b^6*c^9*d^2*f + 52*B*C^2*a^5*b^5*c^6*d^5*f - 48*B^2*C*a^7*b^3*c^7*d^4
*f + 48*B^2*C*a^5*b^5*c^9*d^2*f + 44*B*C^2*a^8*b^2*c^5*d^6*f - 36*B*C^2*a^6
*b^4*c^9*d^2*f + 12*B^2*C*a^8*b^2*c^6*d^5*f - 2958*A^2*C*a^4*b^6*c^4*d^7*f
- 1932*A^2*C*a^2*b^8*c^4*d^7*f + 1848*A^2*C*a^3*b^7*c^5*d^6*f + 1728*A^2*C*
a^3*b^7*c^3*d^8*f + 1524*A^2*C*a^5*b^5*c^5*d^6*f + 1374*A*C^2*a^4*b^6*c^4*d
^7*f - 1272*A*C^2*a^3*b^7*c^5*d^6*f - 1236*A*C^2*a^5*b^5*c^5*d^6*f + 1116*A
*C^2*a^2*b^8*c^4*d^7*f - 1110*A^2*C*a^4*b^6*c^6*d^5*f + 1038*A*C^2*a^4*b^6*
c^6*d^5*f - 768*A^2*C*a^2*b^8*c^2*d^9*f - 696*A^2*C*a^3*b^7*c^7*d^4*f - 666
*A*C^2*a^6*b^4*c^4*d^7*f + 564*A^2*C*a^2*b^8*c^6*d^5*f - 564*A*C^2*a^5*b^5*
c^7*d^4*f - 555*A*C^2*a^2*b^8*c^8*d^3*f + 519*A^2*C*a^2*b^8*c^8*d^3*f - 480
*A*C^2*a^3*b^7*c^3*d^8*f + 456*A*C^2*a^5*b^5*c^3*d^8*f - 420*A*C^2*a^6*b^4*
c^2*d^9*f + 408*A*C^2*a^3*b^7*c^7*d^4*f + 408*A*C^2*a^2*b^8*c^2*d^9*f + 348
*A^2*C*a^6*b^4*c^2*d^9*f - 348*A*C^2*a^2*b^8*c^6*d^5*f + 342*A*C^2*a^6*b^4*
c^6*d^5*f - 336*A*C^2*a^4*b^6*c^8*d^3*f + 324*A^2*C*a^5*b^5*c^7*d^4*f - 312
*A^2*C*a^4*b^6*c^2*d^9*f + 264*A^2*C*a^4*b^6*c^8*d^3*f + 240*A*C^2*a^7*b^3*

c^5*d^6*f + 195*A*C^2*a^8*b^2*c^2*d^9*f - 174*A^2*C*a^6*b^4*c^6*d^5*f + 144
*A*C^2*a^3*b^7*c^9*d^2*f - 123*A^2*C*a^8*b^2*c^2*d^9*f + 120*A*C^2*a^7*b^3*
c^3*d^8*f + 108*A*C^2*a^6*b^4*c^8*d^3*f - 102*A^2*C*a^6*b^4*c^4*d^7*f - 96*
A^2*C*a^8*b^2*c^4*d^7*f + 72*A^2*C*a^7*b^3*c^3*d^8*f + 72*A*C^2*a^5*b^5*c^9
*d^2*f + 48*A^2*C*a^7*b^3*c^5*d^6*f - 48*A^2*C*a^3*b^7*c^9*d^2*f - 48*A*C^2
*a^4*b^6*c^2*d^9*f - 24*A^2*C*a^5*b^5*c^3*d^8*f - 12*A*C^2*a^8*b^2*c^4*d^7*
f + 2736*A^2*B*a^3*b^7*c^6*d^5*f + 2464*A^2*B*a^4*b^6*c^3*d^8*f - 2298*A*B^
2*a^4*b^6*c^4*d^7*f - 2252*A^2*B*a^2*b^8*c^5*d^6*f - 1692*A^2*B*a^5*b^5*c^4
*d^7*f - 1592*A*B^2*a^2*b^8*c^4*d^7*f - 1338*A*B^2*a^4*b^6*c^6*d^5*f + 1320
*A*B^2*a^3*b^7*c^5*d^6*f + 1212*A*B^2*a^5*b^5*c^5*d^6*f - 1056*A*B^2*a^5*b^
5*c^3*d^8*f + 1024*A^2*B*a^3*b^7*c^4*d^7*f - 1022*A^2*B*a^4*b^6*c^7*d^4*f -
880*A^2*B*a^5*b^5*c^2*d^9*f - 846*A^2*B*a^4*b^6*c^5*d^6*f - 840*A*B^2*a^3*
b^7*c^7*d^4*f + 760*A*B^2*a^6*b^4*c^2*d^9*f - 704*A^2*B*a^3*b^7*c^2*d^9*f +
688*A*B^2*a^3*b^7*c^3*d^8*f + 660*A^2*B*a^6*b^4*c^3*d^8*f - 612*A^2*B*a^2*
b^8*c^7*d^4*f + 462*A*B^2*a^6*b^4*c^4*d^7*f + 459*A*B^2*a^2*b^8*c^8*d^3*f -
412*A*B^2*a^2*b^8*c^2*d^9*f - 408*A*B^2*a^7*b^3*c^3*d^8*f + 388*A^2*B*a^5*
b^5*c^6*d^5*f + 296*A^2*B*a^2*b^8*c^3*d^8*f + 288*A*B^2*a^2*b^8*c^6*d^5*f +
284*A*B^2*a^5*b^5*c^7*d^4*f + 236*A*B^2*a^4*b^6*c^8*d^3*f - 226*A*B^2*a^6*
b^4*c^6*d^5*f + 212*A*B^2*a^4*b^6*c^2*d^9*f + 202*A^2*B*a^6*b^4*c^5*d^6*f -
152*A^2*B*a^7*b^3*c^4*d^7*f + 88*A^2*B*a^3*b^7*c^8*d^3*f + 79*A^2*B*a^2*b^
8*c^9*d^2*f - 70*A^2*B*a^6*b^4*c^7*d^4*f + 68*A*B^2*a^8*b^2*c^4*d^7*f + 64*
A^2*B*a^7*b^3*c^2*d^9*f - 64*A*B^2*a^3*b^7*c^9*d^2*f + 56*A^2*B*a^7*b^3*c^6
*d^5*f + 56*A^2*B*a^5*b^5*c^8*d^3*f + 37*A^2*B*a^8*b^2*c^3*d^8*f - 28*A^2*B
*a^8*b^2*c^5*d^6*f - 28*A^2*B*a^4*b^6*c^9*d^2*f + 17*A*B^2*a^8*b^2*c^2*d^9*
f - 16*A*B^2*a^7*b^3*c^5*d^6*f + 24*A*B*C*b^10*c*d^10*f - 6*A*B*C*a^10*c*d^
10*f + 48*A*B*C*a*b^9*d^11*f + 4*A*B*C*a^9*b*d^11*f + 432*B^2*C*a*b^9*c^7*d
^4*f - 376*B*C^2*a^6*b^4*c*d^10*f - 354*B*C^2*a*b^9*c^8*d^3*f + 352*B^2*C*a
^5*b^5*c*d^10*f + 320*B^2*C*a*b^9*c^5*d^6*f + 256*B^2*C*a^3*b^7*c*d^10*f -
232*B^2*C*a^7*b^3*c*d^10*f - 210*B^2*C*a*b^9*c^9*d^2*f - 152*B*C^2*a^4*b^6*
c*d^10*f + 85*B*C^2*a^8*b^2*c*d^10*f + 72*B^2*C*a*b^9*c^3*d^8*f - 48*B*C^2*
a*b^9*c^6*d^5*f - 40*B*C^2*a^3*b^7*c^10*d*f + 40*B*C^2*a^2*b^8*c*d^10*f + 3
7*B^2*C*a^2*b^8*c^10*d*f + 22*B^2*C*a^9*b*c^3*d^8*f - 18*B*C^2*a^9*b*c^2*d^
9*f + 16*B*C^2*a*b^9*c^2*d^9*f - 12*B^2*C*a^4*b^6*c^10*d*f + 8*B*C^2*a^9*b*
c^4*d^7*f + 8*B*C^2*a*b^9*c^4*d^7*f - 984*A^2*C*a*b^9*c^7*d^4*f + 672*A^2*C
*a*b^9*c^3*d^8*f + 552*A*C^2*a*b^9*c^7*d^4*f - 504*A^2*C*a^5*b^5*c*d^10*f -
408*A^2*C*a*b^9*c^5*d^6*f + 408*A*C^2*a*b^9*c^5*d^6*f + 336*A*C^2*a^5*b^5*
c*d^10*f - 216*A*C^2*a^7*b^3*c*d^10*f + 192*A*C^2*a^3*b^7*c*d^10*f - 162*A*
C^2*a*b^9*c^9*d^2*f + 120*A^2*C*a^7*b^3*c*d^10*f + 96*A^2*C*a^3*b^7*c*d^10*
f + 90*A^2*C*a*b^9*c^9*d^2*f + 66*A^2*C*a^9*b*c^3*d^8*f - 66*A*C^2*a^9*b*c^
3*d^8*f + 57*A*C^2*a^2*b^8*c^10*d*f - 48*A*C^2*a*b^9*c^3*d^8*f - 9*A^2*C*a^
2*b^8*c^10*d*f + 1736*A^2*B*a*b^9*c^4*d^7*f + 1248*A^2*B*a*b^9*c^6*d^5*f -
1008*A*B^2*a*b^9*c^7*d^4*f + 772*A^2*B*a^4*b^6*c*d^10*f - 688*A*B^2*a^5*b^5
*c*d^10*f - 608*A*B^2*a*b^9*c^5*d^6*f + 436*A^2*B*a^2*b^8*c*d^10*f - 426*A^
2*B*a*b^9*c^8*d^3*f + 312*A*B^2*a*b^9*c^3*d^8*f + 304*A^2*B*a*b^9*c^2*d^9*f
- 244*A^2*B*a^6*b^4*c*d^10*f - 160*A*B^2*a^3*b^7*c*d^10*f + 114*A*B^2*a*b^

9*c^9*d^2*f + 88*A*B^2*a^7*b^3*c*d^10*f - 22*A*B^2*a^9*b*c^3*d^8*f - 18*A^2
*B*a^9*b*c^2*d^9*f + 13*A^2*B*a^8*b^2*c*d^10*f - 13*A*B^2*a^2*b^8*c^10*d*f
+ 8*A^2*B*a^9*b*c^4*d^7*f + 8*A^2*B*a^3*b^7*c^10*d*f + 111*B^2*C*b^10*c^8*d
^3*f - 39*B*C^2*b^10*c^9*d^2*f + 24*B*C^2*b^10*c^7*d^4*f - 4*B^2*C*b^10*c^2
*d^9*f - 4*B*C^2*b^10*c^5*d^6*f + 432*A^2*C*b^10*c^6*d^5*f + 192*A^2*C*b^10
*c^4*d^7*f - 111*A^2*C*b^10*c^8*d^3*f + 111*A*C^2*b^10*c^8*d^3*f - 72*A*C^2
*b^10*c^6*d^5*f + 12*A*C^2*b^10*c^4*d^7*f - 3*B^2*C*a^10*c^2*d^9*f - B*C^2*
a^10*c^3*d^8*f + 456*A^2*B*b^10*c^7*d^4*f - 288*A^2*B*b^10*c^3*d^8*f + 252*
A*B^2*b^10*c^6*d^5*f + 192*A*B^2*b^10*c^4*d^7*f - 183*A*B^2*b^10*c^8*d^3*f
- 148*A^2*B*b^10*c^5*d^6*f + 112*B^2*C*a^6*b^4*d^11*f + 76*A*B^2*b^10*c^2*d
^9*f - 64*B*C^2*a^7*b^3*d^11*f + 16*B^2*C*a^4*b^6*d^11*f - 16*B^2*C*a^2*b^8
*d^11*f + 16*B*C^2*a^5*b^5*d^11*f + 16*B*C^2*a^3*b^7*d^11*f - 9*A^2*C*a^10*
c^2*d^9*f + 9*A*C^2*a^10*c^2*d^9*f - 3*A^2*B*b^10*c^9*d^2*f - B^2*C*a^8*b^2
*d^11*f + 96*A^2*C*a^4*b^6*d^11*f - 84*A^2*C*a^6*b^4*d^11*f + 72*A*C^2*a^6*
b^4*d^11*f - 24*A*C^2*a^4*b^6*d^11*f - 24*A*C^2*a^2*b^8*d^11*f - 21*A*C^2*a
^8*b^2*d^11*f + 12*A^2*C*a^2*b^8*d^11*f + 9*A^2*C*a^8*b^2*d^11*f + 3*A*B^2*
a^10*c^2*d^9*f - A^2*B*a^10*c^3*d^8*f - B*C^2*a^2*b^8*c^11*f + 176*A*B^2*a^
4*b^6*d^11*f + 136*A^2*B*a^5*b^5*d^11*f - 128*A^2*B*a^3*b^7*d^11*f + 112*A*
B^2*a^2*b^8*d^11*f - 64*A*B^2*a^6*b^4*d^11*f - 16*A^2*B*a^7*b^3*d^11*f - A^
2*B*a^2*b^8*c^11*f - 2*C^3*a^9*b*c*d^10*f - 2*B^3*a*b^9*c^10*d*f - 264*A^3*
a*b^9*c*d^10*f + 2*A^3*a^9*b*c*d^10*f - 9*B^2*C*b^10*c^10*d*f + 9*A^2*C*b^1
0*c^10*d*f - 9*A*C^2*b^10*c^10*d*f + 3*B*C^2*a^10*c*d^10*f - 132*A^2*B*b^10
*c*d^10*f - 3*A*B^2*b^10*c^10*d*f - 2*B*C^2*a^9*b*d^11*f + 3*A^2*B*a^10*c*d
^10*f - 2*B^2*C*a*b^9*c^11*f - 120*A^2*B*a*b^9*d^11*f - 6*A^2*C*a*b^9*c^11*
f + 6*A*C^2*a*b^9*c^11*f - 2*A^2*B*a^9*b*d^11*f + 2*A*B^2*a*b^9*c^11*f + 52
0*C^3*a^3*b^7*c^5*d^6*f + 460*C^3*a^5*b^5*c^5*d^6*f - 418*C^3*a^4*b^6*c^6*d
^5*f + 406*C^3*a^6*b^4*c^4*d^7*f + 268*C^3*a^5*b^5*c^7*d^4*f - 266*C^3*a^6*
b^4*c^6*d^5*f + 233*C^3*a^2*b^8*c^8*d^3*f - 176*C^3*a^7*b^3*c^5*d^6*f + 164
*C^3*a^6*b^4*c^2*d^9*f + 140*C^3*a^2*b^8*c^6*d^5*f + 136*C^3*a^4*b^6*c^2*d^
9*f - 128*C^3*a^3*b^7*c^9*d^2*f + 128*C^3*a^3*b^7*c^3*d^8*f - 108*C^3*a^6*b
^4*c^8*d^3*f - 104*C^3*a^7*b^3*c^3*d^8*f - 104*C^3*a^5*b^5*c^3*d^8*f + 100*
C^3*a^4*b^6*c^8*d^3*f - 89*C^3*a^8*b^2*c^2*d^9*f - 72*C^3*a^5*b^5*c^9*d^2*f
 + 40*C^3*a^8*b^2*c^4*d^7*f - 40*C^3*a^3*b^7*c^7*d^4*f - 28*C^3*a^2*b^8*c^4
*d^7*f - 16*C^3*a^2*b^8*c^2*d^9*f - 2*C^3*a^4*b^6*c^4*d^7*f + 828*B^3*a^5*b
^5*c^4*d^7*f + 408*B^3*a^2*b^8*c^5*d^6*f + 390*B^3*a^4*b^6*c^7*d^4*f - 372*
B^3*a^4*b^6*c^3*d^8*f - 336*B^3*a^3*b^7*c^6*d^5*f - 314*B^3*a^6*b^4*c^5*d^6
*f + 288*B^3*a^3*b^7*c^4*d^7*f + 216*B^3*a^2*b^8*c^7*d^4*f - 176*B^3*a^7*b^
3*c^2*d^9*f + 128*B^3*a^3*b^7*c^2*d^9*f + 108*B^3*a^5*b^5*c^6*d^5*f + 88*B^
3*a^7*b^3*c^4*d^7*f + 72*B^3*a^5*b^5*c^2*d^9*f - 68*B^3*a^2*b^8*c^3*d^8*f -
 65*B^3*a^2*b^8*c^9*d^2*f - 56*B^3*a^5*b^5*c^8*d^3*f + 40*B^3*a^7*b^3*c^6*d
^5*f + 37*B^3*a^8*b^2*c^3*d^8*f + 30*B^3*a^4*b^6*c^5*d^6*f - 28*B^3*a^8*b^2
*c^5*d^6*f + 24*B^3*a^3*b^7*c^8*d^3*f - 4*B^3*a^4*b^6*c^9*d^2*f - 2*B^3*a^6
*b^4*c^7*d^4*f + 1586*A^3*a^4*b^6*c^4*d^7*f - 1376*A^3*a^3*b^7*c^3*d^8*f -
1096*A^3*a^3*b^7*c^5*d^6*f + 844*A^3*a^2*b^8*c^4*d^7*f - 748*A^3*a^5*b^5*c^
5*d^6*f + 490*A^3*a^4*b^6*c^6*d^5*f + 376*A^3*a^2*b^8*c^2*d^9*f + 362*A^3*a

^6*b^4*c^4*d^7*f - 356*A^3*a^2*b^8*c^6*d^5*f - 328*A^3*a^5*b^5*c^3*d^8*f +
328*A^3*a^3*b^7*c^7*d^4*f + 224*A^3*a^4*b^6*c^2*d^9*f - 197*A^3*a^2*b^8*c^8
*d^3*f - 112*A^3*a^7*b^3*c^5*d^6*f + 98*A^3*a^6*b^4*c^6*d^5*f - 92*A^3*a^6*
b^4*c^2*d^9*f - 88*A^3*a^7*b^3*c^3*d^8*f + 68*A^3*a^8*b^2*c^4*d^7*f + 32*A^
3*a^3*b^7*c^9*d^2*f - 28*A^3*a^5*b^5*c^7*d^4*f - 28*A^3*a^4*b^6*c^8*d^3*f +
17*A^3*a^8*b^2*c^2*d^9*f + 104*C^3*a^7*b^3*c*d^10*f + 54*C^3*a*b^9*c^9*d^2
*f - 40*C^3*a*b^9*c^7*d^4*f - 35*C^3*a^2*b^8*c^10*d*f + 22*C^3*a^9*b*c^3*d^
8*f + 16*C^3*a^5*b^5*c*d^10*f - 16*C^3*a^3*b^7*c*d^10*f + 8*C^3*a*b^9*c^5*d
^6*f - 2*A*B*C*b^10*c^11*f + 198*B^3*a*b^9*c^8*d^3*f + 192*B^3*a^6*b^4*c*d^
10*f - 128*B^3*a*b^9*c^4*d^7*f - 80*B^3*a^2*b^8*c*d^10*f - 56*B^3*a*b^9*c^2
*d^9*f - 24*B^3*a*b^9*c^6*d^5*f - 18*B^3*a^9*b*c^2*d^9*f - 16*B^3*a^4*b^6*c
*d^10*f + 13*B^3*a^8*b^2*c*d^10*f + 8*B^3*a^9*b*c^4*d^7*f + 8*B^3*a^3*b^7*c
^10*d*f - 624*A^3*a*b^9*c^3*d^8*f + 472*A^3*a*b^9*c^7*d^4*f - 272*A^3*a^3*b
^7*c*d^10*f + 152*A^3*a^5*b^5*c*d^10*f - 22*A^3*a^9*b*c^3*d^8*f + 18*A^3*a*
b^9*c^9*d^2*f - 13*A^3*a^2*b^8*c^10*d*f - 8*A^3*a^7*b^3*c*d^10*f - 8*A^3*a*
b^9*c^5*d^6*f + A*B^2*a^8*b^2*d^11*f - C^3*b^10*c^8*d^3*f - 60*B^3*b^10*c^7
*d^4*f - 32*B^3*b^10*c^5*d^6*f + 21*B^3*b^10*c^9*d^2*f - 12*B^3*b^10*c^3*d^
8*f - 3*C^3*a^10*c^2*d^9*f - 360*A^3*b^10*c^6*d^5*f - 204*A^3*b^10*c^4*d^7*
f + 11*C^3*a^8*b^2*d^11*f - 8*C^3*a^6*b^4*d^11*f - 4*C^3*a^4*b^6*d^11*f - B
^3*a^10*c^3*d^8*f - 64*B^3*a^5*b^5*d^11*f - 32*B^3*a^3*b^7*d^11*f + 3*A^3*a
^10*c^2*d^9*f - 68*A^3*a^4*b^6*d^11*f + 20*A^3*a^6*b^4*d^11*f + 12*A^3*a^2*
b^8*d^11*f - B^3*a^2*b^8*c^11*f + 3*C^3*b^10*c^10*d*f + 3*B^3*a^10*c*d^10*f
- 3*A^3*b^10*c^10*d*f - 2*C^3*a*b^9*c^11*f - 2*B^3*a^9*b*d^11*f + 2*A^3*a*
b^9*c^11*f - 36*A^2*C*b^10*d^11*f + 3*A^2*C*a^10*d^11*f - 3*A*C^2*a^10*d^11
*f - A*B^2*a^10*d^11*f + 36*A^3*b^10*d^11*f - A^3*a^10*d^11*f + A^3*b^10*c^
8*d^3*f + A^3*a^8*b^2*d^11*f + B^2*C*a^10*d^11*f + B*C^2*b^10*c^11*f + A^2*
B*b^10*c^11*f + C^3*a^10*d^11*f + B^3*b^10*c^11*f - 6*A*B^2*C*a*b^7*c^7*d +
4*A*B^2*C*a*b^7*c*d^7 + 168*A^2*B*C*a^3*b^5*c^2*d^6 + 144*A*B*C^2*a^4*b^4*
c^3*d^5 - 129*A^2*B*C*a^4*b^4*c^3*d^5 - 96*A*B*C^2*a^3*b^5*c^2*d^6 + 84*A*B
*C^2*a^2*b^6*c^3*d^5 + 72*A^2*B*C*a^3*b^5*c^4*d^4 - 72*A^2*B*C*a^2*b^6*c^3*
d^5 + 64*A*B^2*C*a^4*b^4*c^4*d^4 - 60*A*B*C^2*a^3*b^5*c^4*d^4 + 57*A^2*B*C*
a^2*b^6*c^5*d^3 - 56*A*B^2*C*a^3*b^5*c^5*d^3 - 39*A*B^2*C*a^4*b^4*c^2*d^6 -
38*A*B^2*C*a^5*b^3*c^3*d^5 + 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^4*b
^4*c^5*d^3 - 30*A*B*C^2*a^2*b^6*c^5*d^3 + 27*A*B^2*C*a^2*b^6*c^6*d^2 - 24*A
*B^2*C*a^2*b^6*c^2*d^6 - 24*A*B*C^2*a^5*b^3*c^4*d^4 + 24*A*B*C^2*a^3*b^5*c^
6*d^2 + 18*A^2*B*C*a^5*b^3*c^2*d^6 - 18*A^2*B*C*a^4*b^4*c^5*d^3 - 15*A*B^2*
C*a^2*b^6*c^4*d^4 + 12*A^2*B*C*a^5*b^3*c^4*d^4 - 12*A^2*B*C*a^3*b^5*c^6*d^2
+ 9*A*B^2*C*a^6*b^2*c^2*d^6 + 6*A*B*C^2*a^6*b^2*c^3*d^5 - 3*A^2*B*C*a^6*b^
2*c^3*d^5 + 60*A^2*B*C*a*b^7*c^2*d^6 - 51*A^2*B*C*a^4*b^4*c*d^7 + 48*A*B*C^
2*a*b^7*c^6*d^2 - 42*A^2*B*C*a^2*b^6*c*d^7 - 42*A^2*B*C*a*b^7*c^6*d^2 + 36*
A*B*C^2*a^4*b^4*c*d^7 + 36*A*B*C^2*a^2*b^6*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4
- 30*A^2*B*C*a*b^7*c^4*d^4 + 24*A*B^2*C*a*b^7*c^3*d^5 - 24*A*B*C^2*a*b^7*c
^2*d^6 + 18*A*B^2*C*a^5*b^3*c*d^7 - 18*A*B*C^2*a^6*b^2*c*d^7 + 12*A*B^2*C*a
^3*b^5*c*d^7 + 9*A^2*B*C*a^6*b^2*c*d^7 + 6*A*B^2*C*a*b^7*c^5*d^3 - 6*A*B*C^
2*a^2*b^6*c^7*d + 3*A^2*B*C*a^2*b^6*c^7*d - 18*B^3*C*a*b^7*c^6*d^2 - 18*B*C

^3*a*b^7*c^6*d^2 - 14*B^3*C*a*b^7*c^4*d^4 - 14*B*C^3*a*b^7*c^4*d^4 - 10*B^3
*C*a^2*b^6*c*d^7 - 10*B*C^3*a^2*b^6*c*d^7 + 9*B^3*C*a^6*b^2*c*d^7 + 9*B*C^3
*a^6*b^2*c*d^7 - 7*B^3*C*a^4*b^4*c*d^7 - 7*B*C^3*a^4*b^4*c*d^7 + 6*B^2*C^2*
a*b^7*c^7*d - 4*B^3*C*a*b^7*c^2*d^6 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a*b^7
*c^2*d^6 + 3*B^3*C*a^2*b^6*c^7*d + 3*B*C^3*a^2*b^6*c^7*d + 144*A^3*C*a*b^7*
c^3*d^5 + 62*A^3*C*a*b^7*c^5*d^3 + 48*A*C^3*a*b^7*c^3*d^5 - 36*A^2*C^2*a*b^
7*c*d^7 + 26*A*C^3*a*b^7*c^5*d^3 + 20*A^3*C*a^3*b^5*c*d^7 + 18*A^2*C^2*a*b^
7*c^7*d - 18*A*C^3*a^5*b^3*c*d^7 - 6*A^3*C*a^5*b^3*c*d^7 - 4*A*C^3*a^3*b^5*
c*d^7 - 32*A^3*B*a*b^7*c^2*d^6 - 32*A*B^3*a*b^7*c^2*d^6 + 22*A^3*B*a^4*b^4*
c*d^7 + 22*A*B^3*a^4*b^4*c*d^7 + 16*A^3*B*a^2*b^6*c*d^7 + 16*A*B^3*a^2*b^6*
c*d^7 + 12*A^3*B*a*b^7*c^6*d^2 + 12*A*B^3*a*b^7*c^6*d^2 + 8*A^3*B*a*b^7*c^4
*d^4 - 8*A^2*B^2*a*b^7*c*d^7 + 8*A*B^3*a*b^7*c^4*d^4 + 57*A^2*B*C*b^8*c^5*d
^3 + 36*A^2*B*C*b^8*c^3*d^5 - 30*A*B*C^2*b^8*c^5*d^3 - 18*A*B*C^2*b^8*c^3*d
^5 - 9*A*B^2*C*b^8*c^4*d^4 - 3*A*B^2*C*b^8*c^6*d^2 - 2*A*B^2*C*b^8*c^2*d^6
+ 36*A^2*B*C*a^3*b^5*d^8 + 24*A*B*C^2*a^5*b^3*d^8 - 18*A^2*B*C*a^5*b^3*d^8
- 12*A*B*C^2*a^3*b^5*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 3*A*B^2*C*a^4*b^4*d^8 -
2*A*B^2*C*a^2*b^6*d^8 + 34*B^2*C^2*a^5*b^3*c^3*d^5 + 28*B^2*C^2*a^3*b^5*c^5
*d^3 + 24*B^2*C^2*a^4*b^4*c^2*d^6 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2
*a^3*b^5*c^3*d^5 + 12*B^2*C^2*a^2*b^6*c^2*d^6 - 9*B^2*C^2*a^6*b^2*c^2*d^6 +
9*B^2*C^2*a^4*b^4*c^6*d^2 + 9*B^2*C^2*a^2*b^6*c^4*d^4 - 3*B^2*C^2*a^2*b^6*
c^6*d^2 + 159*A^2*C^2*a^2*b^6*c^4*d^4 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^
2*C^2*a^5*b^3*c^3*d^5 + 78*A^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4
*d^4 - 27*A^2*C^2*a^6*b^2*c^2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2
*a^4*b^4*c^2*d^6 + 9*A^2*C^2*a^4*b^4*c^6*d^2 + 66*A^2*B^2*a^2*b^6*c^2*d^6 +
60*A^2*B^2*a^2*b^6*c^4*d^4 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^4*b
^4*c^2*d^6 + 28*A^2*B^2*a^3*b^5*c^5*d^3 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^
2*B^2*a^2*b^6*c^6*d^2 + 4*A^2*B^2*a^5*b^3*c^3*d^5 + 36*A^3*C*a*b^7*c*d^7 -
18*A*C^3*a*b^7*c^7*d + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a*b^7*c^7*d + 12*A^2*
B*C*b^8*c*d^7 + 6*A*B*C^2*b^8*c^7*d - 6*A*B*C^2*b^8*c*d^7 - 3*A^2*B*C*b^8*c
^7*d + 24*A^2*B*C*a*b^7*d^8 - 12*A*B*C^2*a*b^7*d^8 - 53*B^3*C*a^4*b^4*c^3*d
^5 - 53*B*C^3*a^4*b^4*c^3*d^5 - 32*B^3*C*a^2*b^6*c^3*d^5 - 32*B*C^3*a^2*b^6
*c^3*d^5 - 18*B^3*C*a^4*b^4*c^5*d^3 - 18*B*C^3*a^4*b^4*c^5*d^3 + 16*B^3*C*a
^3*b^5*c^4*d^4 + 16*B*C^3*a^3*b^5*c^4*d^4 + 12*B^3*C*a^5*b^3*c^4*d^4 - 12*B
^3*C*a^3*b^5*c^6*d^2 + 12*B^2*C^2*a*b^7*c^3*d^5 + 12*B*C^3*a^5*b^3*c^4*d^4
- 12*B*C^3*a^3*b^5*c^6*d^2 + 8*B^3*C*a^3*b^5*c^2*d^6 + 8*B*C^3*a^3*b^5*c^2*
d^6 - 6*B^3*C*a^5*b^3*c^2*d^6 - 6*B^2*C^2*a^5*b^3*c*d^7 + 6*B^2*C^2*a*b^7*c
^5*d^3 - 6*B*C^3*a^5*b^3*c^2*d^6 - 3*B^3*C*a^6*b^2*c^3*d^5 - 3*B*C^3*a^6*b^
2*c^3*d^5 - 175*A^3*C*a^2*b^6*c^4*d^4 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2
*C^2*a*b^7*c^3*d^5 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^5*b^3*c^3*d^5 -
73*A*C^3*a^2*b^6*c^4*d^4 - 66*A^2*C^2*a*b^7*c^5*d^3 + 44*A*C^3*a^3*b^5*c^3
*d^5 + 36*A*C^3*a^4*b^4*c^4*d^4 - 30*A^3*C*a^5*b^3*c^3*d^5 + 30*A^3*C*a^4*b
^4*c^4*d^4 + 27*A*C^3*a^6*b^2*c^2*d^6 + 21*A*C^3*a^4*b^4*c^2*d^6 + 18*A^2*C
^2*a^5*b^3*c*d^7 - 18*A*C^3*a^4*b^4*c^6*d^2 - 16*A*C^3*a^2*b^6*c^2*d^6 - 15
*A^3*C*a^4*b^4*c^2*d^6 + 15*A^3*C*a^2*b^6*c^6*d^2 - 12*A^2*C^2*a^3*b^5*c*d^
7 + 9*A^3*C*a^6*b^2*c^2*d^6 + 9*A*C^3*a^2*b^6*c^6*d^2 - 80*A^3*B*a^3*b^5*c^

2*d^6 - 80*A*B^3*a^3*b^5*c^2*d^6 + 38*A^3*B*a^4*b^4*c^3*d^5 + 38*A*B^3*a^4*
b^4*c^3*d^5 - 36*A^2*B^2*a*b^7*c^3*d^5 - 28*A^3*B*a^3*b^5*c^4*d^4 - 28*A^3*
B*a^2*b^6*c^5*d^3 - 28*A*B^3*a^3*b^5*c^4*d^4 - 28*A*B^3*a^2*b^6*c^5*d^3 + 2
0*A^3*B*a^2*b^6*c^3*d^5 + 20*A*B^3*a^2*b^6*c^3*d^5 - 12*A^3*B*a^5*b^3*c^2*d
^6 - 12*A^2*B^2*a^5*b^3*c*d^7 - 12*A^2*B^2*a^3*b^5*c*d^7 - 12*A^2*B^2*a*b^7
*c^5*d^3 - 12*A*B^3*a^5*b^3*c^2*d^6 + 6*B^2*C^2*b^8*c^6*d^2 + 3*B^2*C^2*b^8
*c^4*d^4 + 36*A^2*C^2*b^8*c^4*d^4 + 27*A^2*C^2*b^8*c^2*d^6 - 18*A^2*C^2*b^8
*c^6*d^2 + 33*A^2*B^2*b^8*c^4*d^4 + 28*A^2*B^2*b^8*c^2*d^6 + 9*B^2*C^2*a^4*
b^4*d^8 + 6*A^2*B^2*b^8*c^6*d^2 + 4*B^2*C^2*a^2*b^6*d^8 + 3*B^2*C^2*a^6*b^2
*d^8 - 30*A^2*C^2*a^4*b^4*d^8 + 9*A^2*C^2*a^6*b^2*d^8 + 16*A^2*B^2*a^2*b^6*
d^8 + 3*A^2*B^2*a^4*b^4*d^8 + 6*C^4*a^5*b^3*c*d^7 + 4*C^4*a^3*b^5*c*d^7 - 2
*C^4*a*b^7*c^5*d^3 - 12*B^4*a^5*b^3*c*d^7 + 12*B^4*a*b^7*c^3*d^5 + 8*B^4*a*
b^7*c^5*d^3 - 4*B^4*a^3*b^5*c*d^7 - 48*A^4*a*b^7*c^3*d^5 - 20*A^4*a*b^7*c^5
*d^3 - 8*A^4*a^3*b^5*c*d^7 - 63*A^3*C*b^8*c^4*d^4 - 54*A^3*C*b^8*c^2*d^6 +
9*A^3*C*b^8*c^6*d^2 + 9*A*C^3*b^8*c^6*d^2 - 3*A*C^3*b^8*c^4*d^4 - 28*A^3*B*
b^8*c^5*d^3 - 28*A*B^3*b^8*c^5*d^3 - 18*A^3*B*b^8*c^3*d^5 - 18*A*B^3*b^8*c^
3*d^5 - 10*B^3*C*a^5*b^3*d^8 - 10*B*C^3*a^5*b^3*d^8 - 4*B^3*C*a^3*b^5*d^8 -
 4*B*C^3*a^3*b^5*d^8 + 23*A^3*C*a^4*b^4*d^8 - 18*A^3*C*a^2*b^6*d^8 + 11*A*C
^3*a^4*b^4*d^8 - 9*A*C^3*a^6*b^2*d^8 + 6*A*C^3*a^2*b^6*d^8 - 3*A^3*C*a^6*b^
2*d^8 - 20*A^3*B*a^3*b^5*d^8 - 20*A*B^3*a^3*b^5*d^8 + 4*A^3*B*a^5*b^3*d^8 +
 4*A*B^3*a^5*b^3*d^8 + B^3*C*a^2*b^6*c^5*d^3 + B*C^3*a^2*b^6*c^5*d^3 + 6*C^
4*a*b^7*c^7*d + 4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 3*B^3*C*b^8*c^7*d
 - 3*B*C^3*b^8*c^7*d - 6*A^3*B*b^8*c*d^7 - 6*A*B^3*b^8*c*d^7 - 12*A^3*B*a*b^
7*d^8 - 12*A*B^3*a*b^7*d^8 + 30*C^4*a^5*b^3*c^3*d^5 + 19*C^4*a^2*b^6*c^4*d^
4 - 9*C^4*a^6*b^2*c^2*d^6 + 9*C^4*a^4*b^4*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 +
 4*C^4*a^2*b^6*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^4*b^4*c^2*d^6 + 3*
C^4*a^2*b^6*c^6*d^2 + 28*B^4*a^3*b^5*c^5*d^3 + 27*B^4*a^4*b^4*c^2*d^6 - 17*
B^4*a^4*b^4*c^4*d^4 - 10*B^4*a^2*b^6*c^4*d^4 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^
4*a^2*b^6*c^2*d^6 - 6*B^4*a^2*b^6*c^6*d^2 + 4*B^4*a^5*b^3*c^3*d^5 + 70*A^4*
a^2*b^6*c^4*d^4 + 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*
a^4*b^4*c^2*d^6 + B^2*C^2*b^8*c^2*d^6 - 18*A^3*C*b^8*d^8 + B^3*C*b^8*c^5*d^
3 + B*C^3*b^8*c^5*d^3 + 6*B^4*b^8*c^6*d^2 + 3*B^4*b^8*c^4*d^4 + 30*A^4*b^8*
c^4*d^4 + 27*A^4*b^8*c^2*d^6 + 3*C^4*a^6*b^2*d^8 + 8*B^4*a^4*b^4*d^8 + 4*B^
4*a^2*b^6*d^8 + 12*A^4*a^2*b^6*d^8 - 5*A^4*a^4*b^4*d^8 + 9*A^2*C^2*b^8*d^8
 + 9*A^2*B^2*b^8*d^8 + 9*A^4*b^8*d^8 + B^4*b^8*c^2*d^6 + C^4*a^4*b^4*d^8, f,
 k)*(root(640*a^15*b*c^7*d^13*f^4 + 640*a*b^15*c^13*d^7*f^4 + 480*a^15*b*c^
9*d^11*f^4 + 480*a^15*b*c^5*d^15*f^4 + 480*a*b^15*c^15*d^5*f^4 + 480*a*b^15
*c^11*d^9*f^4 + 192*a^15*b*c^11*d^9*f^4 + 192*a^15*b*c^3*d^17*f^4 + 192*a^1
1*b^5*c*d^19*f^4 + 192*a^5*b^11*c^19*d*f^4 + 192*a*b^15*c^17*d^3*f^4 + 192*
a*b^15*c^9*d^11*f^4 + 128*a^13*b^3*c*d^19*f^4 + 128*a^9*b^7*c*d^19*f^4 + 12
8*a^7*b^9*c^19*d*f^4 + 128*a^3*b^13*c^19*d*f^4 + 32*a^15*b*c^13*d^7*f^4 + 3
2*a^9*b^7*c^19*d*f^4 + 32*a^7*b^9*c*d^19*f^4 + 32*a*b^15*c^7*d^13*f^4 + 32*
a^15*b*c*d^19*f^4 + 32*a*b^15*c^19*d*f^4 - 47088*a^8*b^8*c^10*d^10*f^4 + 42
432*a^9*b^7*c^9*d^11*f^4 + 42432*a^7*b^9*c^11*d^9*f^4 + 39328*a^9*b^7*c^11*
d^9*f^4 + 39328*a^7*b^9*c^9*d^11*f^4 - 36912*a^8*b^8*c^12*d^8*f^4 - 36912*a

^8*b^8*c^8*d^12*f^4 - 34256*a^10*b^6*c^10*d^10*f^4 - 34256*a^6*b^10*c^10*d^10*f^4 - 31152*a^10*b^6*c^8*d^12*f^4 - 31152*a^6*b^10*c^12*d^8*f^4 + 28128*a^9*b^7*c^7*d^13*f^4 + 28128*a^7*b^9*c^13*d^7*f^4 + 24160*a^11*b^5*c^9*d^11*f^4 + 24160*a^5*b^11*c^11*d^9*f^4 - 23088*a^10*b^6*c^12*d^8*f^4 - 23088*a^6*b^10*c^8*d^12*f^4 + 22272*a^9*b^7*c^13*d^7*f^4 + 22272*a^7*b^9*c^7*d^13*f^4 + 19072*a^11*b^5*c^11*d^9*f^4 + 19072*a^5*b^11*c^9*d^11*f^4 + 18624*a^11*b^5*c^7*d^13*f^4 + 18624*a^5*b^11*c^13*d^7*f^4 - 17328*a^8*b^8*c^14*d^6*f^4 - 17328*a^8*b^8*c^6*d^14*f^4 - 17232*a^10*b^6*c^6*d^14*f^4 - 17232*a^6*b^10*c^14*d^6*f^4 - 13520*a^12*b^4*c^8*d^12*f^4 - 13520*a^4*b^12*c^12*d^8*f^4 - 12464*a^12*b^4*c^10*d^10*f^4 - 12464*a^4*b^12*c^10*d^10*f^4 + 10880*a^9*b^7*c^5*d^15*f^4 + 10880*a^7*b^9*c^15*d^5*f^4 - 9072*a^10*b^6*c^14*d^6*f^4 - 9072*a^6*b^10*c^6*d^14*f^4 + 8928*a^11*b^5*c^13*d^7*f^4 + 8928*a^5*b^11*c^7*d^13*f^4 - 8880*a^12*b^4*c^6*d^14*f^4 - 8880*a^4*b^12*c^14*d^6*f^4 + 8480*a^11*b^5*c^5*d^15*f^4 + 8480*a^5*b^11*c^15*d^5*f^4 + 7200*a^9*b^7*c^15*d^5*f^4 + 7200*a^7*b^9*c^5*d^15*f^4 - 6912*a^12*b^4*c^12*d^8*f^4 - 6912*a^4*b^12*c^8*d^12*f^4 + 6400*a^13*b^3*c^9*d^11*f^4 + 6400*a^3*b^13*c^11*d^9*f^4 + 5920*a^13*b^3*c^7*d^13*f^4 + 5920*a^3*b^13*c^13*d^7*f^4 - 5392*a^10*b^6*c^4*d^16*f^4 - 5392*a^6*b^10*c^16*d^4*f^4 - 4428*a^8*b^8*c^16*d^4*f^4 - 4428*a^8*b^8*c^4*d^16*f^4 + 4128*a^13*b^3*c^11*d^9*f^4 + 4128*a^3*b^13*c^9*d^11*f^4 - 3328*a^12*b^4*c^4*d^16*f^4 - 3328*a^4*b^12*c^16*d^4*f^4 + 3264*a^13*b^3*c^5*d^15*f^4 + 3264*a^3*b^13*c^15*d^5*f^4 - 2480*a^14*b^2*c^8*d^12*f^4 - 2480*a^2*b^14*c^12*d^8*f^4 + 2240*a^11*b^5*c^15*d^5*f^4 + 2240*a^5*b^11*c^5*d^15*f^4 - 2128*a^12*b^4*c^14*d^6*f^4 - 2128*a^4*b^12*c^6*d^14*f^4 + 2112*a^9*b^7*c^3*d^17*f^4 + 2112*a^7*b^9*c^17*d^3*f^4 + 2048*a^11*b^5*c^3*d^17*f^4 + 2048*a^5*b^11*c^17*d^3*f^4 - 2000*a^14*b^2*c^6*d^14*f^4 - 2000*a^2*b^14*c^14*d^6*f^4 - 1792*a^10*b^6*c^16*d^4*f^4 - 1792*a^6*b^10*c^4*d^16*f^4 - 1776*a^14*b^2*c^10*d^10*f^4 - 1776*a^2*b^14*c^10*d^10*f^4 + 1472*a^13*b^3*c^13*d^7*f^4 + 1472*a^3*b^13*c^7*d^13*f^4 + 1088*a^9*b^7*c^17*d^3*f^4 + 1088*a^7*b^9*c^3*d^17*f^4 + 992*a^13*b^3*c^3*d^17*f^4 + 992*a^3*b^13*c^17*d^3*f^4 - 912*a^14*b^2*c^4*d^16*f^4 - 912*a^2*b^14*c^16*d^4*f^4 - 768*a^10*b^6*c^2*d^18*f^4 - 768*a^6*b^10*c^18*d^2*f^4 - 688*a^14*b^2*c^12*d^8*f^4 - 688*a^2*b^14*c^8*d^12*f^4 - 592*a^12*b^4*c^2*d^18*f^4 - 592*a^4*b^12*c^18*d^2*f^4 - 472*a^8*b^8*c^18*d^2*f^4 - 472*a^8*b^8*c^2*d^18*f^4 - 280*a^12*b^4*c^16*d^4*f^4 - 280*a^4*b^12*c^4*d^16*f^4 + 224*a^13*b^3*c^15*d^5*f^4 + 224*a^11*b^5*c^17*d^3*f^4 + 224*a^5*b^11*c^3*d^17*f^4 + 224*a^3*b^13*c^5*d^15*f^4 - 208*a^14*b^2*c^2*d^18*f^4 - 208*a^2*b^14*c^18*d^2*f^4 - 112*a^14*b^2*c^14*d^6*f^4 - 112*a^10*b^6*c^18*d^2*f^4 - 112*a^6*b^10*c^2*d^18*f^4 - 112*a^2*b^14*c^6*d^14*f^4 - 80*b^16*c^14*d^6*f^4 - 60*b^16*c^16*d^4*f^4 - 60*b^16*c^12*d^8*f^4 - 24*b^16*c^18*d^2*f^4 - 24*b^16*c^10*d^10*f^4 - 4*b^16*c^8*d^12*f^4 - 80*a^16*c^6*d^14*f^4 - 60*a^16*c^8*d^12*f^4 - 60*a^16*c^4*d^16*f^4 - 24*a^16*c^10*d^10*f^4 - 24*a^16*c^2*d^18*f^4 - 4*a^16*c^12*d^8*f^4 - 24*a^12*b^4*d^20*f^4 - 16*a^14*b^2*d^20*f^4 - 16*a^10*b^6*d^20*f^4 - 4*a^8*b^8*d^20*f^4 - 24*a^4*b^12*c^20*f^4 - 16*a^6*b^10*c^20*f^4 - 16*a^2*b^14*c^20*f^4 - 4*a^8*b^8*c^20*f^4 - 4*b^16*c^20*f^4 - 4*a^16*d^20*f^4 + 56*A*C*a*b^11*c^13*d*f^2 - 48*A*C*a^11*b*c*d^13*f^2 + 48*A*C*a*b^11*c*d^13*f^2 + 5904*B

*C*a^6*b^6*c^7*d^7*f^2 - 5016*B*C*a^5*b^7*c^8*d^6*f^2 - 4608*B*C*a^7*b^5*c^6*d^8*f^2 - 4512*B*C*a^5*b^7*c^6*d^8*f^2 - 4384*B*C*a^7*b^5*c^8*d^6*f^2 + 3056*B*C*a^8*b^4*c^7*d^7*f^2 + 2256*B*C*a^4*b^8*c^7*d^7*f^2 - 1824*B*C*a^3*b^9*c^8*d^6*f^2 + 1632*B*C*a^9*b^3*c^4*d^10*f^2 - 1400*B*C*a^8*b^4*c^3*d^11*f^2 - 1320*B*C*a^4*b^8*c^11*d^3*f^2 - 1248*B*C*a^3*b^9*c^6*d^8*f^2 + 1152*B*C*a^3*b^9*c^10*d^4*f^2 - 1072*B*C*a^9*b^3*c^6*d^8*f^2 + 1068*B*C*a^6*b^6*c^9*d^5*f^2 - 1004*B*C*a^4*b^8*c^5*d^9*f^2 - 968*B*C*a^6*b^6*c^3*d^11*f^2 - 864*B*C*a^8*b^4*c^5*d^9*f^2 - 828*B*C*a^4*b^8*c^9*d^5*f^2 - 792*B*C*a^4*b^8*c^3*d^11*f^2 - 792*B*C*a^2*b^10*c^11*d^3*f^2 - 776*B*C*a^9*b^3*c^8*d^6*f^2 + 688*B*C*a^7*b^5*c^4*d^10*f^2 - 672*B*C*a^10*b^2*c^3*d^11*f^2 - 592*B*C*a^2*b^10*c^9*d^5*f^2 + 544*B*C*a^10*b^2*c^7*d^7*f^2 - 492*B*C*a^2*b^10*c^5*d^9*f^2 + 480*B*C*a^5*b^7*c^10*d^4*f^2 - 392*B*C*a^10*b^2*c^5*d^9*f^2 + 332*B*C*a^8*b^4*c^9*d^5*f^2 - 328*B*C*a^6*b^6*c^11*d^3*f^2 + 320*B*C*a^9*b^3*c^2*d^12*f^2 + 272*B*C*a^3*b^9*c^12*d^2*f^2 - 248*B*C*a^5*b^7*c^4*d^10*f^2 - 248*B*C*a^2*b^10*c^3*d^11*f^2 - 208*B*C*a^7*b^5*c^10*d^4*f^2 - 192*B*C*a^5*b^7*c^2*d^12*f^2 + 144*B*C*a^2*b^10*c^7*d^7*f^2 - 96*B*C*a^3*b^9*c^4*d^10*f^2 + 88*B*C*a^5*b^7*c^12*d^2*f^2 - 72*B*C*a^8*b^4*c^11*d^3*f^2 + 48*B*C*a^9*b^3*c^10*d^4*f^2 - 48*B*C*a^7*b^5*c^12*d^2*f^2 - 48*B*C*a^7*b^5*c^2*d^12*f^2 - 48*B*C*a^3*b^9*c^2*d^12*f^2 - 12*B*C*a^10*b^2*c^9*d^5*f^2 + 4*B*C*a^6*b^6*c^5*d^9*f^2 + 5824*A*C*a^7*b^5*c^5*d^9*f^2 - 4378*A*C*a^8*b^4*c^6*d^8*f^2 + 4296*A*C*a^5*b^7*c^5*d^9*f^2 - 3912*A*C*a^6*b^6*c^6*d^8*f^2 - 3672*A*C*a^5*b^7*c^9*d^5*f^2 + 3594*A*C*a^4*b^8*c^8*d^6*f^2 + 3236*A*C*a^6*b^6*c^8*d^6*f^2 + 2816*A*C*a^9*b^3*c^5*d^9*f^2 + 2624*A*C*a^3*b^9*c^5*d^9*f^2 + 2432*A*C*a^7*b^5*c^7*d^7*f^2 - 2366*A*C*a^8*b^4*c^4*d^10*f^2 + 2298*A*C*a^4*b^8*c^10*d^4*f^2 + 1872*A*C*a^3*b^9*c^7*d^7*f^2 + 1848*A*C*a^6*b^6*c^10*d^4*f^2 - 1644*A*C*a^6*b^6*c^4*d^10*f^2 - 1488*A*C*a^7*b^5*c^9*d^5*f^2 - 1408*A*C*a^3*b^9*c^9*d^5*f^2 - 1308*A*C*a^4*b^8*c^6*d^8*f^2 + 1248*A*C*a^5*b^7*c^7*d^7*f^2 - 1012*A*C*a^10*b^2*c^6*d^8*f^2 + 1008*A*C*a^7*b^5*c^3*d^11*f^2 + 992*A*C*a^5*b^7*c^3*d^11*f^2 + 928*A*C*a^3*b^9*c^3*d^11*f^2 + 848*A*C*a^9*b^3*c^7*d^7*f^2 + 636*A*C*a^2*b^10*c^8*d^6*f^2 - 628*A*C*a^10*b^2*c^4*d^10*f^2 - 600*A*C*a^2*b^10*c^6*d^8*f^2 - 576*A*C*a^5*b^7*c^11*d^3*f^2 + 572*A*C*a^2*b^10*c^10*d^4*f^2 + 464*A*C*a^8*b^4*c^8*d^6*f^2 + 304*A*C*a^6*b^6*c^2*d^12*f^2 - 304*A*C*a^4*b^8*c^4*d^10*f^2 + 296*A*C*a^4*b^8*c^2*d^12*f^2 + 260*A*C*a^8*b^4*c^10*d^4*f^2 - 232*A*C*a^9*b^3*c^9*d^5*f^2 - 232*A*C*a^2*b^10*c^12*d^2*f^2 + 228*A*C*a^10*b^2*c^2*d^12*f^2 - 188*A*C*a^2*b^10*c^4*d^10*f^2 + 144*A*C*a^3*b^9*c^11*d^3*f^2 + 116*A*C*a^6*b^6*c^12*d^2*f^2 + 112*A*C*a^9*b^3*c^3*d^11*f^2 - 112*A*C*a^7*b^5*c^11*d^3*f^2 + 92*A*C*a^10*b^2*c^8*d^6*f^2 + 74*A*C*a^4*b^8*c^12*d^2*f^2 + 62*A*C*a^8*b^4*c^2*d^12*f^2 + 40*A*C*a^2*b^10*c^2*d^12*f^2 - 7008*A*B*a^6*b^6*c^7*d^7*f^2 - 4032*A*B*a^4*b^8*c^7*d^7*f^2 + 3952*A*B*a^7*b^5*c^8*d^6*f^2 + 3648*A*B*a^5*b^7*c^8*d^6*f^2 - 3392*A*B*a^8*b^4*c^7*d^7*f^2 + 3264*A*B*a^7*b^5*c^6*d^8*f^2 - 2992*A*B*a^5*b^7*c^4*d^10*f^2 - 2368*A*B*a^7*b^5*c^4*d^10*f^2 - 2304*A*B*a^3*b^9*c^4*d^10*f^2 - 1968*A*B*a^6*b^6*c^9*d^5*f^2 - 1872*A*B*a^9*b^3*c^4*d^10*f^2 - 1728*A*B*a^2*b^10*c^7*d^7*f^2 + 1712*A*B*a^8*b^4*c^3*d^11*f^2 + 1536*A*B*a^5*b^7*c^6*d^8*f^2 - 1536*A*B*a^3*b^9*c^10*d^4*f^2 - 1392*A*B*a^5*b^7*c^2*d^12*f^2

+ 1328*A*B*a^6*b^6*c^3*d^11*f^2 - 1104*A*B*a^3*b^9*c^2*d^12*f^2 - 1056*A*B
*a^3*b^9*c^6*d^8*f^2 + 976*A*B*a^9*b^3*c^6*d^8*f^2 + 960*A*B*a^4*b^8*c^11*d
^3*f^2 + 936*A*B*a^8*b^4*c^5*d^9*f^2 - 912*A*B*a^5*b^7*c^10*d^4*f^2 + 848*A
*B*a^9*b^3*c^8*d^6*f^2 - 816*A*B*a^7*b^5*c^2*d^12*f^2 + 816*A*B*a^4*b^8*c^3
*d^11*f^2 + 768*A*B*a^10*b^2*c^3*d^11*f^2 + 672*A*B*a^3*b^9*c^8*d^6*f^2 - 6
32*A*B*a^8*b^4*c^9*d^5*f^2 - 608*A*B*a^2*b^10*c^9*d^5*f^2 - 552*A*B*a^4*b^8
*c^9*d^5*f^2 - 544*A*B*a^10*b^2*c^7*d^7*f^2 - 480*A*B*a^2*b^10*c^5*d^9*f^2
+ 464*A*B*a^10*b^2*c^5*d^9*f^2 - 464*A*B*a^9*b^3*c^2*d^12*f^2 + 432*A*B*a^2
*b^10*c^11*d^3*f^2 - 368*A*B*a^3*b^9*c^12*d^2*f^2 - 256*A*B*a^6*b^6*c^5*d^9
*f^2 - 208*A*B*a^5*b^7*c^12*d^2*f^2 + 176*A*B*a^4*b^8*c^5*d^9*f^2 + 112*A*B
*a^7*b^5*c^10*d^4*f^2 + 112*A*B*a^6*b^6*c^11*d^3*f^2 - 16*A*B*a^2*b^10*c^3*
d^11*f^2 - 576*B*C*a*b^11*c^8*d^6*f^2 + 400*B*C*a^11*b*c^4*d^10*f^2 - 288*B
*C*a*b^11*c^6*d^8*f^2 - 176*B*C*a^11*b*c^6*d^8*f^2 + 128*B*C*a*b^11*c^10*d^
4*f^2 - 108*B*C*a^4*b^8*c*d^13*f^2 - 104*B*C*a*b^11*c^4*d^10*f^2 - 92*B*C*a
^4*b^8*c^13*d*f^2 - 60*B*C*a^8*b^4*c*d^13*f^2 - 60*B*C*a^6*b^6*c*d^13*f^2 +
48*B*C*a^11*b*c^2*d^12*f^2 - 40*B*C*a^2*b^10*c*d^13*f^2 - 28*B*C*a^2*b^10*
c^13*d*f^2 - 24*B*C*a*b^11*c^12*d^2*f^2 + 20*B*C*a^10*b^2*c*d^13*f^2 - 16*B
*C*a*b^11*c^2*d^12*f^2 + 12*B*C*a^6*b^6*c^13*d*f^2 + 912*A*C*a*b^11*c^7*d^7
*f^2 + 808*A*C*a*b^11*c^5*d^9*f^2 + 432*A*C*a^11*b*c^5*d^9*f^2 + 336*A*C*a*
b^11*c^3*d^11*f^2 + 224*A*C*a*b^11*c^11*d^3*f^2 - 112*A*C*a^11*b*c^3*d^11*f
^2 + 112*A*C*a^3*b^9*c*d^13*f^2 - 88*A*C*a^9*b^3*c*d^13*f^2 + 80*A*C*a^3*b^
9*c^13*d*f^2 + 56*A*C*a^5*b^7*c*d^13*f^2 + 48*A*C*a*b^11*c^9*d^5*f^2 - 40*A
*C*a^5*b^7*c^13*d*f^2 - 16*A*C*a^11*b*c^7*d^7*f^2 + 16*A*C*a^7*b^5*c*d^13*f
^2 - 496*A*B*a*b^11*c^4*d^10*f^2 - 400*A*B*a^11*b*c^4*d^10*f^2 + 288*A*B*a*
b^11*c^8*d^6*f^2 - 288*A*B*a*b^11*c^6*d^8*f^2 - 272*A*B*a*b^11*c^2*d^12*f^2
+ 240*A*B*a^6*b^6*c*d^13*f^2 - 224*A*B*a*b^11*c^10*d^4*f^2 + 192*A*B*a^8*b
^4*c*d^13*f^2 + 192*A*B*a^4*b^8*c*d^13*f^2 + 176*A*B*a^11*b*c^6*d^8*f^2 + 1
04*A*B*a^4*b^8*c^13*d*f^2 - 48*A*B*a^11*b*c^2*d^12*f^2 + 16*A*B*a^10*b^2*c*
d^13*f^2 + 16*A*B*a^2*b^10*c^13*d*f^2 + 16*A*B*a^2*b^10*c*d^13*f^2 - 112*B*
C*b^12*c^11*d^3*f^2 + 4*B*C*b^12*c^5*d^9*f^2 + 150*A*C*b^12*c^10*d^4*f^2 -
80*B*C*a^12*c^3*d^11*f^2 + 66*A*C*b^12*c^8*d^6*f^2 - 30*A*C*b^12*c^12*d^2*f
^2 + 24*B*C*a^12*c^5*d^9*f^2 - 12*A*C*b^12*c^4*d^10*f^2 - 576*A*B*b^12*c^7*
d^7*f^2 - 432*A*B*b^12*c^9*d^5*f^2 - 400*A*B*b^12*c^5*d^9*f^2 - 144*A*B*b^1
2*c^3*d^11*f^2 - 96*B*C*a^7*b^5*d^14*f^2 - 72*B*C*a^5*b^7*d^14*f^2 - 66*A*C
*a^12*c^4*d^10*f^2 + 54*A*C*a^12*c^2*d^12*f^2 - 32*A*B*b^12*c^11*d^3*f^2 -
24*B*C*a^9*b^3*d^14*f^2 - 16*B*C*a^3*b^9*d^14*f^2 + 2*A*C*a^12*c^6*d^8*f^2
+ 116*A*C*a^6*b^6*d^14*f^2 + 100*A*C*a^4*b^8*d^14*f^2 + 80*A*B*a^12*c^3*d^1
1*f^2 + 24*A*C*a^2*b^10*d^14*f^2 - 24*A*B*a^12*c^5*d^9*f^2 + 22*A*C*a^8*b^4
*d^14*f^2 + 16*B*C*a^3*b^9*c^14*f^2 + 8*A*C*a^10*b^2*d^14*f^2 - 192*A*B*a^5
*b^7*d^14*f^2 - 176*A*B*a^3*b^9*d^14*f^2 - 48*A*B*a^7*b^5*d^14*f^2 - 28*A*C
*a^2*b^10*c^14*f^2 + 2*A*C*a^4*b^8*c^14*f^2 - 16*A*B*a^3*b^9*c^14*f^2 + 250
8*C^2*a^6*b^6*c^6*d^8*f^2 + 2376*C^2*a^5*b^7*c^9*d^5*f^2 + 2357*C^2*a^8*b^4
*c^6*d^8*f^2 - 2048*C^2*a^7*b^5*c^5*d^9*f^2 + 1304*C^2*a^3*b^9*c^9*d^5*f^2
+ 1303*C^2*a^8*b^4*c^4*d^10*f^2 + 1212*C^2*a^6*b^6*c^4*d^10*f^2 - 1203*C^2*
a^4*b^8*c^8*d^6*f^2 - 1192*C^2*a^9*b^3*c^5*d^9*f^2 + 1062*C^2*a^4*b^8*c^6*d

^8*f^2 + 984*C^2*a^7*b^5*c^9*d^5*f^2 - 952*C^2*a^6*b^6*c^8*d^6*f^2 + 768*C^
2*a^5*b^7*c^7*d^7*f^2 - 681*C^2*a^4*b^8*c^10*d^4*f^2 - 672*C^2*a^5*b^7*c^5*
d^9*f^2 - 480*C^2*a^6*b^6*c^10*d^4*f^2 + 458*C^2*a^10*b^2*c^6*d^8*f^2 - 448
*C^2*a^7*b^5*c^7*d^7*f^2 + 422*C^2*a^4*b^8*c^4*d^10*f^2 + 372*C^2*a^2*b^10*
c^6*d^8*f^2 + 360*C^2*a^5*b^7*c^11*d^3*f^2 + 312*C^2*a^3*b^9*c^7*d^7*f^2 +
278*C^2*a^10*b^2*c^4*d^10*f^2 - 232*C^2*a^9*b^3*c^7*d^7*f^2 + 194*C^2*a^2*b
^10*c^12*d^2*f^2 + 176*C^2*a^9*b^3*c^9*d^5*f^2 + 152*C^2*a^5*b^7*c^3*d^11*f
^2 + 124*C^2*a^2*b^10*c^4*d^10*f^2 - 120*C^2*a^7*b^5*c^3*d^11*f^2 - 114*C^2
*a^10*b^2*c^2*d^12*f^2 - 102*C^2*a^2*b^10*c^8*d^6*f^2 + 101*C^2*a^4*b^8*c^1
2*d^2*f^2 + 100*C^2*a^6*b^6*c^2*d^12*f^2 - 88*C^2*a^3*b^9*c^5*d^9*f^2 + 77*
C^2*a^8*b^4*c^2*d^12*f^2 + 72*C^2*a^3*b^9*c^11*d^3*f^2 - 64*C^2*a^10*b^2*c^
8*d^6*f^2 + 64*C^2*a^3*b^9*c^3*d^11*f^2 - 58*C^2*a^2*b^10*c^10*d^4*f^2 + 56
*C^2*a^7*b^5*c^11*d^3*f^2 + 56*C^2*a^6*b^6*c^12*d^2*f^2 + 40*C^2*a^9*b^3*c^
3*d^11*f^2 + 36*C^2*a^8*b^4*c^12*d^2*f^2 + 32*C^2*a^4*b^8*c^2*d^12*f^2 + 26
*C^2*a^8*b^4*c^10*d^4*f^2 + 16*C^2*a^2*b^10*c^2*d^12*f^2 + 2*C^2*a^8*b^4*c^
8*d^6*f^2 + 2277*B^2*a^4*b^8*c^8*d^6*f^2 + 2144*B^2*a^7*b^5*c^5*d^9*f^2 - 2
112*B^2*a^5*b^7*c^9*d^5*f^2 + 2028*B^2*a^6*b^6*c^8*d^6*f^2 - 1671*B^2*a^8*b
^4*c^6*d^8*f^2 + 1275*B^2*a^4*b^8*c^10*d^4*f^2 + 1176*B^2*a^5*b^7*c^5*d^9*f
^2 + 1096*B^2*a^9*b^3*c^5*d^9*f^2 - 1044*B^2*a^6*b^6*c^6*d^8*f^2 + 984*B^2*
a^6*b^6*c^10*d^4*f^2 - 968*B^2*a^3*b^9*c^9*d^5*f^2 - 888*B^2*a^7*b^5*c^9*d^
5*f^2 + 672*B^2*a^7*b^5*c^7*d^7*f^2 + 664*B^2*a^3*b^9*c^5*d^9*f^2 - 649*B^2
*a^8*b^4*c^4*d^10*f^2 + 618*B^2*a^2*b^10*c^8*d^6*f^2 + 514*B^2*a^4*b^8*c^4*
d^10*f^2 + 460*B^2*a^6*b^6*c^2*d^12*f^2 + 422*B^2*a^8*b^4*c^8*d^6*f^2 + 406
*B^2*a^2*b^10*c^10*d^4*f^2 - 382*B^2*a^10*b^2*c^6*d^8*f^2 + 368*B^2*a^4*b^8
*c^2*d^12*f^2 - 312*B^2*a^5*b^7*c^11*d^3*f^2 + 312*B^2*a^3*b^9*c^7*d^7*f^2
+ 248*B^2*a^9*b^3*c^7*d^7*f^2 + 245*B^2*a^8*b^4*c^2*d^12*f^2 - 192*B^2*a^5*
b^7*c^7*d^7*f^2 - 184*B^2*a^9*b^3*c^3*d^11*f^2 + 182*B^2*a^10*b^2*c^2*d^12*
f^2 + 176*B^2*a^3*b^9*c^3*d^11*f^2 + 174*B^2*a^4*b^8*c^6*d^8*f^2 - 170*B^2*
a^10*b^2*c^4*d^10*f^2 - 152*B^2*a^9*b^3*c^9*d^5*f^2 + 152*B^2*a^2*b^10*c^4*
d^10*f^2 + 142*B^2*a^8*b^4*c^10*d^4*f^2 - 90*B^2*a^2*b^10*c^12*d^2*f^2 + 88
*B^2*a^2*b^10*c^2*d^12*f^2 + 84*B^2*a^10*b^2*c^8*d^6*f^2 + 84*B^2*a^2*b^10*
c^6*d^8*f^2 + 60*B^2*a^6*b^6*c^12*d^2*f^2 - 56*B^2*a^7*b^5*c^11*d^3*f^2 + 5
3*B^2*a^4*b^8*c^12*d^2*f^2 + 24*B^2*a^7*b^5*c^3*d^11*f^2 + 24*B^2*a^6*b^6*c
^4*d^10*f^2 + 24*B^2*a^3*b^9*c^11*d^3*f^2 - 8*B^2*a^5*b^7*c^3*d^11*f^2 + 45
66*A^2*a^4*b^8*c^6*d^8*f^2 + 4284*A^2*a^6*b^6*c^6*d^8*f^2 - 3776*A^2*a^7*b^
5*c^5*d^9*f^2 - 3624*A^2*a^5*b^7*c^5*d^9*f^2 + 3122*A^2*a^4*b^8*c^4*d^10*f^
2 + 3108*A^2*a^2*b^10*c^6*d^8*f^2 + 2741*A^2*a^8*b^4*c^6*d^8*f^2 + 2592*A^2
*a^6*b^6*c^4*d^10*f^2 - 2536*A^2*a^3*b^9*c^5*d^9*f^2 + 2224*A^2*a^2*b^10*c^
4*d^10*f^2 - 2184*A^2*a^3*b^9*c^7*d^7*f^2 - 2016*A^2*a^5*b^7*c^7*d^7*f^2 -
1984*A^2*a^7*b^5*c^7*d^7*f^2 + 1626*A^2*a^2*b^10*c^8*d^6*f^2 - 1624*A^2*a^9
*b^3*c^5*d^9*f^2 + 1603*A^2*a^8*b^4*c^4*d^10*f^2 + 1296*A^2*a^5*b^7*c^9*d^5
*f^2 - 1144*A^2*a^5*b^7*c^3*d^11*f^2 - 992*A^2*a^3*b^9*c^3*d^11*f^2 + 968*A
^2*a^4*b^8*c^2*d^12*f^2 - 888*A^2*a^7*b^5*c^3*d^11*f^2 + 849*A^2*a^4*b^8*c^
8*d^6*f^2 + 808*A^2*a^2*b^10*c^2*d^12*f^2 - 616*A^2*a^9*b^3*c^7*d^7*f^2 + 5
54*A^2*a^10*b^2*c^6*d^8*f^2 + 504*A^2*a^7*b^5*c^9*d^5*f^2 - 504*A^2*a^6*b^6

*c^10*d^4*f^2 + 460*A^2*a^6*b^6*c^2*d^12*f^2 + 350*A^2*a^10*b^2*c^4*d^10*f^
2 + 350*A^2*a^2*b^10*c^10*d^4*f^2 - 321*A^2*a^4*b^8*c^10*d^4*f^2 + 216*A^2*
a^5*b^7*c^11*d^3*f^2 - 216*A^2*a^3*b^9*c^11*d^3*f^2 + 182*A^2*a^2*b^10*c^12
*d^2*f^2 - 152*A^2*a^9*b^3*c^3*d^11*f^2 - 124*A^2*a^6*b^6*c^8*d^6*f^2 - 114
*A^2*a^10*b^2*c^2*d^12*f^2 + 104*A^2*a^3*b^9*c^9*d^5*f^2 + 77*A^2*a^8*b^4*c
^2*d^12*f^2 + 74*A^2*a^8*b^4*c^8*d^6*f^2 - 70*A^2*a^8*b^4*c^10*d^4*f^2 + 56
*A^2*a^9*b^3*c^9*d^5*f^2 + 56*A^2*a^7*b^5*c^11*d^3*f^2 + 41*A^2*a^4*b^8*c^1
2*d^2*f^2 - 28*A^2*a^10*b^2*c^8*d^6*f^2 - 28*A^2*a^6*b^6*c^12*d^2*f^2 + 12*
B*C*b^12*c^13*d*f^2 + 24*B*C*a^12*c*d^13*f^2 - 24*A*B*b^12*c^13*d*f^2 - 24*
A*B*b^12*c*d^13*f^2 - 16*B*C*a^11*b*d^14*f^2 - 24*A*B*a^12*c*d^13*f^2 - 16*
B*C*a*b^11*c^14*f^2 - 48*A*B*a*b^11*d^14*f^2 + 16*A*B*a^11*b*d^14*f^2 + 16*
A*B*a*b^11*c^14*f^2 - 216*C^2*a^11*b*c^5*d^9*f^2 + 216*C^2*a*b^11*c^9*d^5*f
^2 + 56*C^2*a^11*b*c^3*d^11*f^2 + 56*C^2*a^9*b^3*c*d^13*f^2 + 56*C^2*a^5*b^
7*c*d^13*f^2 + 40*C^2*a^7*b^5*c*d^13*f^2 - 40*C^2*a*b^11*c^11*d^3*f^2 + 32*
C^2*a^5*b^7*c^13*d*f^2 - 24*C^2*a*b^11*c^7*d^7*f^2 - 16*C^2*a^3*b^9*c^13*d*
f^2 + 16*C^2*a^3*b^9*c*d^13*f^2 + 8*C^2*a^11*b*c^7*d^7*f^2 - 8*C^2*a*b^11*c
^5*d^9*f^2 + 264*B^2*a*b^11*c^7*d^7*f^2 + 224*B^2*a*b^11*c^5*d^9*f^2 + 168*
B^2*a^11*b*c^5*d^9*f^2 - 112*B^2*a^9*b^3*c*d^13*f^2 - 104*B^2*a^11*b*c^3*d^
11*f^2 - 104*B^2*a^7*b^5*c*d^13*f^2 + 96*B^2*a*b^11*c^3*d^11*f^2 + 88*B^2*a
*b^11*c^11*d^3*f^2 - 72*B^2*a*b^11*c^9*d^5*f^2 - 64*B^2*a^5*b^7*c*d^13*f^2
+ 32*B^2*a^3*b^9*c^13*d*f^2 - 24*B^2*a^11*b*c^7*d^7*f^2 - 24*B^2*a^5*b^7*c^
13*d*f^2 + 16*B^2*a^3*b^9*c*d^13*f^2 - 888*A^2*a*b^11*c^7*d^7*f^2 - 800*A^2
*a*b^11*c^5*d^9*f^2 - 336*A^2*a*b^11*c^3*d^11*f^2 - 264*A^2*a*b^11*c^9*d^5*
f^2 - 216*A^2*a^11*b*c^5*d^9*f^2 - 184*A^2*a*b^11*c^11*d^3*f^2 - 128*A^2*a^
3*b^9*c*d^13*f^2 - 112*A^2*a^5*b^7*c*d^13*f^2 - 64*A^2*a^3*b^9*c^13*d*f^2 +
56*A^2*a^11*b*c^3*d^11*f^2 - 56*A^2*a^7*b^5*c*d^13*f^2 + 32*A^2*a^9*b^3*c*
d^13*f^2 + 8*A^2*a^11*b*c^7*d^7*f^2 + 8*A^2*a^5*b^7*c^13*d*f^2 + 24*C^2*a^1
1*b*c*d^13*f^2 - 16*C^2*a*b^11*c^13*d*f^2 - 40*B^2*a^11*b*c*d^13*f^2 + 24*B
^2*a*b^11*c^13*d*f^2 + 16*B^2*a*b^11*c*d^13*f^2 - 48*A^2*a*b^11*c*d^13*f^2
- 40*A^2*a*b^11*c^13*d*f^2 + 24*A^2*a^11*b*c*d^13*f^2 - 6*A*C*a^12*d^14*f^2
+ 2*A*C*b^12*c^14*f^2 + 33*C^2*b^12*c^12*d^2*f^2 - 27*C^2*b^12*c^10*d^4*f^
2 + 3*C^2*b^12*c^8*d^6*f^2 + 117*B^2*b^12*c^10*d^4*f^2 + 111*B^2*b^12*c^8*d
^6*f^2 + 72*B^2*b^12*c^6*d^8*f^2 + 33*C^2*a^12*c^4*d^10*f^2 - 27*C^2*a^12*c
^2*d^12*f^2 + 24*B^2*b^12*c^4*d^10*f^2 + 4*B^2*b^12*c^2*d^12*f^2 - 3*B^2*b^
12*c^12*d^2*f^2 - C^2*a^12*c^6*d^8*f^2 + 720*A^2*b^12*c^6*d^8*f^2 + 552*A^2
*b^12*c^4*d^10*f^2 + 471*A^2*b^12*c^8*d^6*f^2 + 216*A^2*b^12*c^2*d^12*f^2 +
93*A^2*b^12*c^10*d^4*f^2 + 33*B^2*a^12*c^2*d^12*f^2 + 33*A^2*b^12*c^12*d^2
*f^2 + 31*C^2*a^8*b^4*d^14*f^2 - 27*B^2*a^12*c^4*d^10*f^2 + 20*C^2*a^6*b^6*
d^14*f^2 + 4*C^2*a^4*b^8*d^14*f^2 + 3*B^2*a^12*c^6*d^8*f^2 + 2*C^2*a^10*b^2
*d^14*f^2 + 80*B^2*a^6*b^6*d^14*f^2 + 64*B^2*a^4*b^8*d^14*f^2 + 33*A^2*a^12
*c^4*d^10*f^2 + 31*B^2*a^8*b^4*d^14*f^2 - 27*A^2*a^12*c^2*d^12*f^2 + 16*B^2
*a^2*b^10*d^14*f^2 + 14*C^2*a^2*b^10*c^14*f^2 + 14*B^2*a^10*b^2*d^14*f^2 -
C^2*a^4*b^8*c^14*f^2 - A^2*a^12*c^6*d^8*f^2 + 120*A^2*a^2*b^10*d^14*f^2 + 1
12*A^2*a^4*b^8*d^14*f^2 - 17*A^2*a^8*b^4*d^14*f^2 - 10*B^2*a^2*b^10*c^14*f^
2 - 10*A^2*a^10*b^2*d^14*f^2 + 8*A^2*a^6*b^6*d^14*f^2 + 3*B^2*a^4*b^8*c^14*

f^2 + 14*A^2*a^2*b^10*c^14*f^2 - A^2*a^4*b^8*c^14*f^2 + 3*C^2*a^12*d^14*f^2 - C^2*b^12*c^14*f^2 + 36*A^2*b^12*d^14*f^2 + 3*B^2*b^12*c^14*f^2 - B^2*a^12*d^14*f^2 + 3*A^2*a^12*d^14*f^2 - A^2*b^12*c^14*f^2 - 44*A*B*C*a*b^9*c^10*d*f + 3816*A*B*C*a^5*b^5*c^4*d^7*f + 2920*A*B*C*a^2*b^8*c^5*d^6*f - 2736*A*B*C*a^3*b^7*c^6*d^5*f - 2672*A*B*C*a^4*b^6*c^3*d^8*f + 1996*A*B*C*a^4*b^6*c^7*d^4*f - 1412*A*B*C*a^6*b^4*c^5*d^6*f + 1120*A*B*C*a^3*b^7*c^2*d^9*f + 1080*A*B*C*a^2*b^8*c^7*d^4*f + 1040*A*B*C*a^5*b^5*c^2*d^9*f + 684*A*B*C*a^4*b^6*c^5*d^6*f + 592*A*B*C*a^3*b^7*c^4*d^7*f - 560*A*B*C*a^7*b^3*c^2*d^9*f - 448*A*B*C*a^2*b^8*c^3*d^8*f - 400*A*B*C*a^5*b^5*c^8*d^3*f - 398*A*B*C*a^2*b^8*c^9*d^2*f - 312*A*B*C*a^6*b^4*c^3*d^8*f + 166*A*B*C*a^8*b^2*c^3*d^8*f + 136*A*B*C*a^5*b^5*c^6*d^5*f + 128*A*B*C*a^7*b^3*c^6*d^5*f - 100*A*B*C*a^6*b^4*c^7*d^4*f + 64*A*B*C*a^7*b^3*c^4*d^7*f - 64*A*B*C*a^4*b^6*c^9*d^2*f - 32*A*B*C*a^3*b^7*c^8*d^3*f - 16*A*B*C*a^8*b^2*c^5*d^6*f - 1312*A*B*C*a*b^9*c^4*d^7*f + 996*A*B*C*a*b^9*c^8*d^3*f + 728*A*B*C*a^6*b^4*c*d^10*f - 624*A*B*C*a*b^9*c^6*d^5*f - 584*A*B*C*a^2*b^8*c*d^10*f - 512*A*B*C*a^4*b^6*c*d^10*f - 320*A*B*C*a*b^9*c^2*d^9*f - 98*A*B*C*a^8*b^2*c*d^10*f + 36*A*B*C*a^9*b*c^2*d^9*f + 32*A*B*C*a^3*b^7*c^10*d*f - 16*A*B*C*a^9*b*c^4*d^7*f + 46*B*C^2*a*b^9*c^10*d*f - 16*B^2*C*a*b^9*c*d^10*f - 2*B^2*C*a^9*b*c*d^10*f + 312*A^2*C*a*b^9*c*d^10*f - 48*A*C^2*a*b^9*c*d^10*f - 6*A^2*C*a^9*b*c*d^10*f + 6*A*C^2*a^9*b*c*d^10*f + 208*A*B^2*a*b^9*c*d^10*f - 2*A^2*B*a*b^9*c^10*d*f + 2*A*B^2*a^9*b*c*d^10*f - 480*A*B*C*b^10*c^7*d^4*f + 78*A*B*C*b^10*c^9*d^2*f - 64*A*B*C*b^10*c^5*d^6*f + 2*A*B*C*a^10*c^3*d^8*f - 224*A*B*C*a^5*b^5*d^11*f + 80*A*B*C*a^7*b^3*d^11*f - 32*A*B*C*a^3*b^7*d^11*f + 2*A*B*C*a^2*b^8*c^11*f - 1692*B*C^2*a^5*b^5*c^4*d^7*f - 1500*B^2*C*a^5*b^5*c^5*d^6*f - 1464*B^2*C*a^3*b^7*c^5*d^6*f + 1426*B*C^2*a^6*b^4*c^5*d^6*f - 1158*B^2*C*a^6*b^4*c^4*d^7*f + 1152*B*C^2*a^3*b^7*c^6*d^5*f + 1026*B^2*C*a^4*b^6*c^6*d^5*f - 974*B*C^2*a^4*b^6*c^7*d^4*f + 960*B^2*C*a^5*b^5*c^3*d^8*f - 884*B*C^2*a^2*b^8*c^5*d^6*f - 764*B^2*C*a^5*b^5*c^7*d^4*f + 752*B^2*C*a^2*b^8*c^4*d^7*f - 752*B*C^2*a^3*b^7*c^4*d^7*f + 738*B^2*C*a^4*b^6*c^4*d^7*f - 688*B^2*C*a^6*b^4*c^2*d^9*f - 675*B^2*C*a^2*b^8*c^8*d^3*f + 560*B*C^2*a^5*b^5*c^8*d^3*f + 496*B*C^2*a^7*b^3*c^2*d^9*f + 496*B*C^2*a^4*b^6*c^3*d^8*f - 468*B*C^2*a^2*b^8*c^7*d^4*f + 456*B^2*C*a^7*b^3*c^3*d^8*f - 452*B^2*C*a^4*b^6*c^8*d^3*f - 416*B*C^2*a^3*b^7*c^2*d^9*f + 378*B*C^2*a^4*b^6*c^5*d^6*f + 376*B*C^2*a^3*b^7*c^8*d^3*f - 360*B^2*C*a^2*b^8*c^6*d^5*f + 355*B*C^2*a^2*b^8*c^9*d^2*f + 346*B^2*C*a^6*b^4*c^6*d^5*f - 320*B^2*C*a^4*b^6*c^2*d^9*f + 268*B^2*C*a^2*b^8*c^2*d^9*f + 216*B^2*C*a^3*b^7*c^7*d^4*f - 203*B*C^2*a^8*b^2*c^3*d^8*f - 184*B*C^2*a^7*b^3*c^6*d^5*f + 170*B*C^2*a^6*b^4*c^7*d^4*f + 160*B^2*C*a^7*b^3*c^5*d^6*f - 160*B*C^2*a^5*b^5*c^2*d^9*f - 140*B^2*C*a^8*b^2*c^4*d^7*f - 136*B*C^2*a^2*b^8*c^3*d^8*f + 112*B^2*C*a^3*b^7*c^9*d^2*f + 91*B^2*C*a^8*b^2*c^2*d^9*f + 88*B*C^2*a^7*b^3*c^4*d^7*f + 72*B^2*C*a^6*b^4*c^8*d^3*f - 64*B^2*C*a^3*b^7*c^3*d^8*f - 60*B*C^2*a^6*b^4*c^3*d^8*f + 56*B*C^2*a^4*b^6*c^9*d^2*f + 52*B*C^2*a^5*b^5*c^6*d^5*f - 48*B^2*C*a^7*b^3*c^7*d^4*f + 48*B^2*C*a^5*b^5*c^9*d^2*f + 44*B*C^2*a^8*b^2*c^5*d^6*f - 36*B*C^2*a^6*b^4*c^9*d^2*f + 12*B^2*C*a^8*b^2*c^6*d^5*f - 2958*A^2*C*a^4*b^6*c^4*d^7*f - 1932*A^2*C*a^2*b^8*c^4*d^7*f + 1848*A^2*C*a^3*b^7*c^5*d^6*f + 1728*A^2*C*a^3*b^7*c^3*d^8

*f + 1524*A^2*C*a^5*b^5*c^5*d^6*f + 1374*A*C^2*a^4*b^6*c^4*d^7*f - 1272*A*C
^2*a^3*b^7*c^5*d^6*f - 1236*A*C^2*a^5*b^5*c^5*d^6*f + 1116*A*C^2*a^2*b^8*c^
4*d^7*f - 1110*A^2*C*a^4*b^6*c^6*d^5*f + 1038*A*C^2*a^4*b^6*c^6*d^5*f - 768
*A^2*C*a^2*b^8*c^2*d^9*f - 696*A^2*C*a^3*b^7*c^7*d^4*f - 666*A*C^2*a^6*b^4*
c^4*d^7*f + 564*A^2*C*a^2*b^8*c^6*d^5*f - 564*A*C^2*a^5*b^5*c^7*d^4*f - 555
*A*C^2*a^2*b^8*c^8*d^3*f + 519*A^2*C*a^2*b^8*c^8*d^3*f - 480*A*C^2*a^3*b^7*
c^3*d^8*f + 456*A*C^2*a^5*b^5*c^3*d^8*f - 420*A*C^2*a^6*b^4*c^2*d^9*f + 408
*A*C^2*a^3*b^7*c^7*d^4*f + 408*A*C^2*a^2*b^8*c^2*d^9*f + 348*A^2*C*a^6*b^4*
c^2*d^9*f - 348*A*C^2*a^2*b^8*c^6*d^5*f + 342*A*C^2*a^6*b^4*c^6*d^5*f - 336
*A*C^2*a^4*b^6*c^8*d^3*f + 324*A^2*C*a^5*b^5*c^7*d^4*f - 312*A^2*C*a^4*b^6*
c^2*d^9*f + 264*A^2*C*a^4*b^6*c^8*d^3*f + 240*A*C^2*a^7*b^3*c^5*d^6*f + 195
*A*C^2*a^8*b^2*c^2*d^9*f - 174*A^2*C*a^6*b^4*c^6*d^5*f + 144*A*C^2*a^3*b^7*
c^9*d^2*f - 123*A^2*C*a^8*b^2*c^2*d^9*f + 120*A*C^2*a^7*b^3*c^3*d^8*f + 108
*A*C^2*a^6*b^4*c^8*d^3*f - 102*A^2*C*a^6*b^4*c^4*d^7*f - 96*A^2*C*a^8*b^2*c
^4*d^7*f + 72*A^2*C*a^7*b^3*c^3*d^8*f + 72*A*C^2*a^5*b^5*c^9*d^2*f + 48*A^2
*C*a^7*b^3*c^5*d^6*f - 48*A^2*C*a^3*b^7*c^9*d^2*f - 48*A*C^2*a^4*b^6*c^2*d^
9*f - 24*A^2*C*a^5*b^5*c^3*d^8*f - 12*A*C^2*a^8*b^2*c^4*d^7*f + 2736*A^2*B*
a^3*b^7*c^6*d^5*f + 2464*A^2*B*a^4*b^6*c^3*d^8*f - 2298*A*B^2*a^4*b^6*c^4*d
^7*f - 2252*A^2*B*a^2*b^8*c^5*d^6*f - 1692*A^2*B*a^5*b^5*c^4*d^7*f - 1592*A
*B^2*a^2*b^8*c^4*d^7*f - 1338*A*B^2*a^4*b^6*c^6*d^5*f + 1320*A*B^2*a^3*b^7*
c^5*d^6*f + 1212*A*B^2*a^5*b^5*c^5*d^6*f - 1056*A*B^2*a^5*b^5*c^3*d^8*f + 1
024*A^2*B*a^3*b^7*c^4*d^7*f - 1022*A^2*B*a^4*b^6*c^7*d^4*f - 880*A^2*B*a^5*
b^5*c^2*d^9*f - 846*A^2*B*a^4*b^6*c^5*d^6*f - 840*A*B^2*a^3*b^7*c^7*d^4*f +
 760*A*B^2*a^6*b^4*c^2*d^9*f - 704*A^2*B*a^3*b^7*c^2*d^9*f + 688*A*B^2*a^3*
b^7*c^3*d^8*f + 660*A^2*B*a^6*b^4*c^3*d^8*f - 612*A^2*B*a^2*b^8*c^7*d^4*f +
 462*A*B^2*a^6*b^4*c^4*d^7*f + 459*A*B^2*a^2*b^8*c^8*d^3*f - 412*A*B^2*a^2*
b^8*c^2*d^9*f - 408*A*B^2*a^7*b^3*c^3*d^8*f + 388*A^2*B*a^5*b^5*c^6*d^5*f +
 296*A^2*B*a^2*b^8*c^3*d^8*f + 288*A*B^2*a^2*b^8*c^6*d^5*f + 284*A*B^2*a^5*
b^5*c^7*d^4*f + 236*A*B^2*a^4*b^6*c^8*d^3*f - 226*A*B^2*a^6*b^4*c^6*d^5*f +
 212*A*B^2*a^4*b^6*c^2*d^9*f + 202*A^2*B*a^6*b^4*c^5*d^6*f - 152*A^2*B*a^7*
b^3*c^4*d^7*f + 88*A^2*B*a^3*b^7*c^8*d^3*f + 79*A^2*B*a^2*b^8*c^9*d^2*f - 7
0*A^2*B*a^6*b^4*c^7*d^4*f + 68*A*B^2*a^8*b^2*c^4*d^7*f + 64*A^2*B*a^7*b^3*c
^2*d^9*f - 64*A*B^2*a^3*b^7*c^9*d^2*f + 56*A^2*B*a^7*b^3*c^6*d^5*f + 56*A^2
*B*a^5*b^5*c^8*d^3*f + 37*A^2*B*a^8*b^2*c^3*d^8*f - 28*A^2*B*a^8*b^2*c^5*d^
6*f - 28*A^2*B*a^4*b^6*c^9*d^2*f + 17*A*B^2*a^8*b^2*c^2*d^9*f - 16*A*B^2*a^
7*b^3*c^5*d^6*f + 24*A*B*C*b^10*c*d^10*f - 6*A*B*C*a^10*c*d^10*f + 48*A*B*C
*a*b^9*d^11*f + 4*A*B*C*a^9*b*d^11*f + 432*B^2*C*a*b^9*c^7*d^4*f - 376*B*C^
2*a^6*b^4*c*d^10*f - 354*B*C^2*a*b^9*c^8*d^3*f + 352*B^2*C*a^5*b^5*c*d^10*f
 + 320*B^2*C*a*b^9*c^5*d^6*f + 256*B^2*C*a^3*b^7*c*d^10*f - 232*B^2*C*a^7*b
^3*c*d^10*f - 210*B^2*C*a*b^9*c^9*d^2*f - 152*B*C^2*a^4*b^6*c*d^10*f + 85*B
*C^2*a^8*b^2*c*d^10*f + 72*B^2*C*a*b^9*c^3*d^8*f - 48*B*C^2*a*b^9*c^6*d^5*f
 - 40*B*C^2*a^3*b^7*c^10*d*f + 40*B*C^2*a^2*b^8*c*d^10*f + 37*B^2*C*a^2*b^8
*c^10*d*f + 22*B^2*C*a^9*b*c^3*d^8*f - 18*B*C^2*a^9*b*c^2*d^9*f + 16*B*C^2*
a*b^9*c^2*d^9*f - 12*B^2*C*a^4*b^6*c^10*d*f + 8*B*C^2*a^9*b*c^4*d^7*f + 8*B
*C^2*a*b^9*c^4*d^7*f - 984*A^2*C*a*b^9*c^7*d^4*f + 672*A^2*C*a*b^9*c^3*d^8*

f + 552*A*C^2*a*b^9*c^7*d^4*f - 504*A^2*C*a^5*b^5*c*d^10*f - 408*A^2*C*a*b^9*c^5*d^6*f + 408*A*C^2*a*b^9*c^5*d^6*f + 336*A*C^2*a^5*b^5*c*d^10*f - 216*A*C^2*a^7*b^3*c*d^10*f + 192*A*C^2*a^3*b^7*c*d^10*f - 162*A*C^2*a*b^9*c^9*d^2*f + 120*A^2*C*a^7*b^3*c*d^10*f + 96*A^2*C*a^3*b^7*c*d^10*f + 90*A^2*C*a*b^9*c^9*d^2*f + 66*A^2*C*a^9*b*c^3*d^8*f - 66*A*C^2*a^9*b*c^3*d^8*f + 57*A*C^2*a^2*b^8*c^10*d*f - 48*A*C^2*a*b^9*c^3*d^8*f - 9*A^2*C*a^2*b^8*c^10*d*f + 1736*A^2*B*a*b^9*c^4*d^7*f + 1248*A^2*B*a*b^9*c^6*d^5*f - 1008*A*B^2*a*b^9*c^7*d^4*f + 772*A^2*B*a^4*b^6*c*d^10*f - 688*A*B^2*a^5*b^5*c*d^10*f - 608*A*B^2*a*b^9*c^5*d^6*f + 436*A^2*B*a^2*b^8*c*d^10*f - 426*A^2*B*a*b^9*c^8*d^3*f + 312*A*B^2*a*b^9*c^3*d^8*f + 304*A^2*B*a*b^9*c^2*d^9*f - 244*A^2*B*a^6*b^4*c*d^10*f - 160*A*B^2*a^3*b^7*c*d^10*f + 114*A*B^2*a*b^9*c^9*d^2*f + 88*A*B^2*a^7*b^3*c*d^10*f - 22*A*B^2*a^9*b*c^3*d^8*f - 18*A^2*B*a^9*b*c^2*d^9*f + 13*A^2*B*a^8*b^2*c*d^10*f - 13*A*B^2*a^2*b^8*c^10*d*f + 8*A^2*B*a^9*b*c^4*d^7*f + 8*A^2*B*a^3*b^7*c^10*d*f + 111*B^2*C*b^10*c^8*d^3*f - 39*B*C^2*b^10*c^9*d^2*f + 24*B*C^2*b^10*c^7*d^4*f - 4*B^2*C*b^10*c^2*d^9*f - 4*B*C^2*b^10*c^5*d^6*f + 432*A^2*C*b^10*c^6*d^5*f + 192*A^2*C*b^10*c^4*d^7*f - 111*A^2*C*b^10*c^8*d^3*f + 111*A*C^2*b^10*c^8*d^3*f - 72*A*C^2*b^10*c^6*d^5*f + 12*A*C^2*b^10*c^4*d^7*f - 3*B^2*C*a^10*c^2*d^9*f - B*C^2*a^10*c^3*d^8*f + 456*A^2*B*b^10*c^7*d^4*f - 288*A^2*B*b^10*c^3*d^8*f + 252*A*B^2*b^10*c^6*d^5*f + 192*A*B^2*b^10*c^4*d^7*f - 183*A*B^2*b^10*c^8*d^3*f - 148*A^2*B*b^10*c^5*d^6*f + 112*B^2*C*a^6*b^4*d^11*f + 76*A*B^2*b^10*c^2*d^9*f - 64*B*C^2*a^7*b^3*d^11*f + 16*B^2*C*a^4*b^6*d^11*f - 16*B^2*C*a^2*b^8*d^11*f + 16*B*C^2*a^5*b^5*d^11*f + 16*B*C^2*a^3*b^7*d^11*f - 9*A^2*C*a^10*c^2*d^9*f + 9*A*C^2*a^10*c^2*d^9*f - 3*A^2*B*b^10*c^9*d^2*f - B^2*C*a^8*b^2*d^11*f + 96*A^2*C*a^4*b^6*d^11*f - 84*A^2*C*a^6*b^4*d^11*f + 72*A*C^2*a^6*b^4*d^11*f - 24*A*C^2*a^4*b^6*d^11*f - 24*A*C^2*a^2*b^8*d^11*f - 21*A*C^2*a^8*b^2*d^11*f + 12*A^2*C*a^2*b^8*d^11*f + 9*A^2*C*a^8*b^2*d^11*f + 3*A*B^2*a^10*c^2*d^9*f - A^2*B*a^10*c^3*d^8*f - B*C^2*a^2*b^8*c^11*f + 176*A*B^2*a^4*b^6*d^11*f + 136*A^2*B*a^5*b^5*d^11*f - 128*A^2*B*a^3*b^7*d^11*f + 112*A*B^2*a^2*b^8*d^11*f - 64*A*B^2*a^6*b^4*d^11*f - 16*A^2*B*a^7*b^3*d^11*f - A^2*B*a^2*b^8*c^11*f - 2*C^3*a^9*b*c*d^10*f - 2*B^3*a*b^9*c^10*d*f - 264*A^3*a*b^9*c*d^10*f + 2*A^3*a^9*b*c*d^10*f - 9*B^2*C*b^10*c^10*d*f + 9*A^2*C*b^10*c^10*d*f - 9*A*C^2*b^10*c^10*d*f + 3*B*C^2*a^10*c*d^10*f - 132*A^2*B*b^10*c*d^10*f - 3*A*B^2*b^10*c^10*d*f - 2*B*C^2*a^9*b*d^11*f + 3*A^2*B*a^10*c*d^10*f - 2*B^2*C*a*b^9*c^11*f - 120*A^2*B*a*b^9*d^11*f - 6*A^2*C*a*b^9*c^11*f + 6*A*C^2*a*b^9*c^11*f - 2*A^2*B*a^9*b*d^11*f + 2*A*B^2*a*b^9*c^11*f + 520*C^3*a^3*b^7*c^5*d^6*f + 460*C^3*a^5*b^5*c^5*d^6*f - 418*C^3*a^4*b^6*c^6*d^5*f + 406*C^3*a^6*b^4*c^4*d^7*f + 268*C^3*a^5*b^5*c^7*d^4*f - 266*C^3*a^6*b^4*c^6*d^5*f + 233*C^3*a^2*b^8*c^8*d^3*f - 176*C^3*a^7*b^3*c^5*d^6*f + 164*C^3*a^6*b^4*c^2*d^9*f + 140*C^3*a^2*b^8*c^6*d^5*f + 136*C^3*a^4*b^6*c^2*d^9*f - 128*C^3*a^3*b^7*c^9*d^2*f + 128*C^3*a^3*b^7*c^3*d^8*f - 108*C^3*a^6*b^4*c^8*d^3*f - 104*C^3*a^7*b^3*c^3*d^8*f - 104*C^3*a^5*b^5*c^3*d^8*f + 100*C^3*a^4*b^6*c^8*d^3*f - 89*C^3*a^8*b^2*c^2*d^9*f - 72*C^3*a^5*b^5*c^9*d^2*f + 40*C^3*a^8*b^2*c^4*d^7*f - 40*C^3*a^3*b^7*c^7*d^4*f - 28*C^3*a^2*b^8*c^4*d^7*f - 16*C^3*a^2*b^8*c^2*d^9*f - 2*C^3*a^4*b^6*c^4*d^7*f + 828*B^3*a^5*b^5*c^4*d^7*f +

408*B^3*a^2*b^8*c^5*d^6*f + 390*B^3*a^4*b^6*c^7*d^4*f - 372*B^3*a^4*b^6*c^3
*d^8*f - 336*B^3*a^3*b^7*c^6*d^5*f - 314*B^3*a^6*b^4*c^5*d^6*f + 288*B^3*a^
3*b^7*c^4*d^7*f + 216*B^3*a^2*b^8*c^7*d^4*f - 176*B^3*a^7*b^3*c^2*d^9*f + 1
28*B^3*a^3*b^7*c^2*d^9*f + 108*B^3*a^5*b^5*c^6*d^5*f + 88*B^3*a^7*b^3*c^4*d
^7*f + 72*B^3*a^5*b^5*c^2*d^9*f - 68*B^3*a^2*b^8*c^3*d^8*f - 65*B^3*a^2*b^8
*c^9*d^2*f - 56*B^3*a^5*b^5*c^8*d^3*f + 40*B^3*a^7*b^3*c^6*d^5*f + 37*B^3*a
^8*b^2*c^3*d^8*f + 30*B^3*a^4*b^6*c^5*d^6*f - 28*B^3*a^8*b^2*c^5*d^6*f + 24
*B^3*a^3*b^7*c^8*d^3*f - 4*B^3*a^4*b^6*c^9*d^2*f - 2*B^3*a^6*b^4*c^7*d^4*f
+ 1586*A^3*a^4*b^6*c^4*d^7*f - 1376*A^3*a^3*b^7*c^3*d^8*f - 1096*A^3*a^3*b^
7*c^5*d^6*f + 844*A^3*a^2*b^8*c^4*d^7*f - 748*A^3*a^5*b^5*c^5*d^6*f + 490*A
^3*a^4*b^6*c^6*d^5*f + 376*A^3*a^2*b^8*c^2*d^9*f + 362*A^3*a^6*b^4*c^4*d^7*
f - 356*A^3*a^2*b^8*c^6*d^5*f - 328*A^3*a^5*b^5*c^3*d^8*f + 328*A^3*a^3*b^7
*c^7*d^4*f + 224*A^3*a^4*b^6*c^2*d^9*f - 197*A^3*a^2*b^8*c^8*d^3*f - 112*A^
3*a^7*b^3*c^5*d^6*f + 98*A^3*a^6*b^4*c^6*d^5*f - 92*A^3*a^6*b^4*c^2*d^9*f -
88*A^3*a^7*b^3*c^3*d^8*f + 68*A^3*a^8*b^2*c^4*d^7*f + 32*A^3*a^3*b^7*c^9*d
^2*f - 28*A^3*a^5*b^5*c^7*d^4*f - 28*A^3*a^4*b^6*c^8*d^3*f + 17*A^3*a^8*b^2
*c^2*d^9*f + 104*C^3*a^7*b^3*c*d^10*f + 54*C^3*a*b^9*c^9*d^2*f - 40*C^3*a*b
^9*c^7*d^4*f - 35*C^3*a^2*b^8*c^10*d*f + 22*C^3*a^9*b*c^3*d^8*f + 16*C^3*a^
5*b^5*c*d^10*f - 16*C^3*a^3*b^7*c*d^10*f + 8*C^3*a*b^9*c^5*d^6*f - 2*A*B*C*
b^10*c^11*f + 198*B^3*a*b^9*c^8*d^3*f + 192*B^3*a^6*b^4*c*d^10*f - 128*B^3*
a*b^9*c^4*d^7*f - 80*B^3*a^2*b^8*c*d^10*f - 56*B^3*a*b^9*c^2*d^9*f - 24*B^3
*a*b^9*c^6*d^5*f - 18*B^3*a^9*b*c^2*d^9*f - 16*B^3*a^4*b^6*c*d^10*f + 13*B^
3*a^8*b^2*c*d^10*f + 8*B^3*a^9*b*c^4*d^7*f + 8*B^3*a^3*b^7*c^10*d*f - 624*A
^3*a*b^9*c^3*d^8*f + 472*A^3*a*b^9*c^7*d^4*f - 272*A^3*a^3*b^7*c*d^10*f + 1
52*A^3*a^5*b^5*c*d^10*f - 22*A^3*a^9*b*c^3*d^8*f + 18*A^3*a*b^9*c^9*d^2*f -
13*A^3*a^2*b^8*c^10*d*f - 8*A^3*a^7*b^3*c*d^10*f - 8*A^3*a*b^9*c^5*d^6*f +
A*B^2*a^8*b^2*d^11*f - C^3*b^10*c^8*d^3*f - 60*B^3*b^10*c^7*d^4*f - 32*B^3
*b^10*c^5*d^6*f + 21*B^3*b^10*c^9*d^2*f - 12*B^3*b^10*c^3*d^8*f - 3*C^3*a^1
0*c^2*d^9*f - 360*A^3*b^10*c^6*d^5*f - 204*A^3*b^10*c^4*d^7*f + 11*C^3*a^8*
b^2*d^11*f - 8*C^3*a^6*b^4*d^11*f - 4*C^3*a^4*b^6*d^11*f - B^3*a^10*c^3*d^8
*f - 64*B^3*a^5*b^5*d^11*f - 32*B^3*a^3*b^7*d^11*f + 3*A^3*a^10*c^2*d^9*f -
68*A^3*a^4*b^6*d^11*f + 20*A^3*a^6*b^4*d^11*f + 12*A^3*a^2*b^8*d^11*f - B^
3*a^2*b^8*c^11*f + 3*C^3*b^10*c^10*d*f + 3*B^3*a^10*c*d^10*f - 3*A^3*b^10*c
^10*d*f - 2*C^3*a*b^9*c^11*f - 2*B^3*a^9*b*d^11*f + 2*A^3*a*b^9*c^11*f - 36
*A^2*C*b^10*d^11*f + 3*A^2*C*a^10*d^11*f - 3*A*C^2*a^10*d^11*f - A*B^2*a^10
*d^11*f + 36*A^3*b^10*d^11*f - A^3*a^10*d^11*f + A^3*b^10*c^8*d^3*f + A^3*a
^8*b^2*d^11*f + B^2*C*a^10*d^11*f + B*C^2*b^10*c^11*f + A^2*B*b^10*c^11*f +
C^3*a^10*d^11*f + B^3*b^10*c^11*f - 6*A*B^2*C*a*b^7*c^7*d + 4*A*B^2*C*a*b^
7*c*d^7 + 168*A^2*B*C*a^3*b^5*c^2*d^6 + 144*A*B*C^2*a^4*b^4*c^3*d^5 - 129*A
^2*B*C*a^4*b^4*c^3*d^5 - 96*A*B*C^2*a^3*b^5*c^2*d^6 + 84*A*B*C^2*a^2*b^6*c^
3*d^5 + 72*A^2*B*C*a^3*b^5*c^4*d^4 - 72*A^2*B*C*a^2*b^6*c^3*d^5 + 64*A*B^2*
C*a^4*b^4*c^4*d^4 - 60*A*B*C^2*a^3*b^5*c^4*d^4 + 57*A^2*B*C*a^2*b^6*c^5*d^3
- 56*A*B^2*C*a^3*b^5*c^5*d^3 - 39*A*B^2*C*a^4*b^4*c^2*d^6 - 38*A*B^2*C*a^5
*b^3*c^3*d^5 + 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^4*b^4*c^5*d^3 - 30
*A*B*C^2*a^2*b^6*c^5*d^3 + 27*A*B^2*C*a^2*b^6*c^6*d^2 - 24*A*B^2*C*a^2*b^6*

c^2*d^6 - 24*A*B*C^2*a^5*b^3*c^4*d^4 + 24*A*B*C^2*a^3*b^5*c^6*d^2 + 18*A^2*
B*C*a^5*b^3*c^2*d^6 - 18*A^2*B*C*a^4*b^4*c^5*d^3 - 15*A*B^2*C*a^2*b^6*c^4*d
^4 + 12*A^2*B*C*a^5*b^3*c^4*d^4 - 12*A^2*B*C*a^3*b^5*c^6*d^2 + 9*A*B^2*C*a^
6*b^2*c^2*d^6 + 6*A*B*C^2*a^6*b^2*c^3*d^5 - 3*A^2*B*C*a^6*b^2*c^3*d^5 + 60*
A^2*B*C*a*b^7*c^2*d^6 - 51*A^2*B*C*a^4*b^4*c*d^7 + 48*A*B*C^2*a*b^7*c^6*d^2
 - 42*A^2*B*C*a^2*b^6*c*d^7 - 42*A^2*B*C*a*b^7*c^6*d^2 + 36*A*B*C^2*a^4*b^4
*c*d^7 + 36*A*B*C^2*a^2*b^6*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4 - 30*A^2*B*C*a
*b^7*c^4*d^4 + 24*A*B^2*C*a*b^7*c^3*d^5 - 24*A*B*C^2*a*b^7*c^2*d^6 + 18*A*B
^2*C*a^5*b^3*c*d^7 - 18*A*B*C^2*a^6*b^2*c*d^7 + 12*A*B^2*C*a^3*b^5*c*d^7 +
9*A^2*B*C*a^6*b^2*c*d^7 + 6*A*B^2*C*a*b^7*c^5*d^3 - 6*A*B*C^2*a^2*b^6*c^7*d
 + 3*A^2*B*C*a^2*b^6*c^7*d - 18*B^3*C*a*b^7*c^6*d^2 - 18*B*C^3*a*b^7*c^6*d^
2 - 14*B^3*C*a*b^7*c^4*d^4 - 14*B*C^3*a*b^7*c^4*d^4 - 10*B^3*C*a^2*b^6*c*d^
7 - 10*B*C^3*a^2*b^6*c*d^7 + 9*B^3*C*a^6*b^2*c*d^7 + 9*B*C^3*a^6*b^2*c*d^7
 - 7*B^3*C*a^4*b^4*c*d^7 - 7*B*C^3*a^4*b^4*c*d^7 + 6*B^2*C^2*a*b^7*c^7*d - 4
*B^3*C*a*b^7*c^2*d^6 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a*b^7*c^2*d^6 + 3*B^
3*C*a^2*b^6*c^7*d + 3*B*C^3*a^2*b^6*c^7*d + 144*A^3*C*a*b^7*c^3*d^5 + 62*A^
3*C*a*b^7*c^5*d^3 + 48*A*C^3*a*b^7*c^3*d^5 - 36*A^2*C^2*a*b^7*c*d^7 + 26*A*
C^3*a*b^7*c^5*d^3 + 20*A^3*C*a^3*b^5*c*d^7 + 18*A^2*C^2*a*b^7*c^7*d - 18*A*
C^3*a^5*b^3*c*d^7 - 6*A^3*C*a^5*b^3*c*d^7 - 4*A*C^3*a^3*b^5*c*d^7 - 32*A^3*
B*a*b^7*c^2*d^6 - 32*A*B^3*a*b^7*c^2*d^6 + 22*A^3*B*a^4*b^4*c*d^7 + 22*A*B^
3*a^4*b^4*c*d^7 + 16*A^3*B*a^2*b^6*c*d^7 + 16*A*B^3*a^2*b^6*c*d^7 + 12*A^3*
B*a*b^7*c^6*d^2 + 12*A*B^3*a*b^7*c^6*d^2 + 8*A^3*B*a*b^7*c^4*d^4 - 8*A^2*B^
2*a*b^7*c*d^7 + 8*A*B^3*a*b^7*c^4*d^4 + 57*A^2*B*C*b^8*c^5*d^3 + 36*A^2*B*C
*b^8*c^3*d^5 - 30*A*B*C^2*b^8*c^5*d^3 - 18*A*B*C^2*b^8*c^3*d^5 - 9*A*B^2*C*
b^8*c^4*d^4 - 3*A*B^2*C*b^8*c^6*d^2 - 2*A*B^2*C*b^8*c^2*d^6 + 36*A^2*B*C*a^
3*b^5*d^8 + 24*A*B*C^2*a^5*b^3*d^8 - 18*A^2*B*C*a^5*b^3*d^8 - 12*A*B*C^2*a^
3*b^5*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 3*A*B^2*C*a^4*b^4*d^8 - 2*A*B^2*C*a^2*b
^6*d^8 + 34*B^2*C^2*a^5*b^3*c^3*d^5 + 28*B^2*C^2*a^3*b^5*c^5*d^3 + 24*B^2*C
^2*a^4*b^4*c^2*d^6 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*d^
5 + 12*B^2*C^2*a^2*b^6*c^2*d^6 - 9*B^2*C^2*a^6*b^2*c^2*d^6 + 9*B^2*C^2*a^4*
b^4*c^6*d^2 + 9*B^2*C^2*a^2*b^6*c^4*d^4 - 3*B^2*C^2*a^2*b^6*c^6*d^2 + 159*A
^2*C^2*a^2*b^6*c^4*d^4 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^5*b^3*c
^3*d^5 + 78*A^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2*C
^2*a^6*b^2*c^2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^4*b^4*c^2*d^
6 + 9*A^2*C^2*a^4*b^4*c^6*d^2 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a^2
*b^6*c^4*d^4 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^4*b^4*c^2*d^6 + 28
*A^2*B^2*a^3*b^5*c^5*d^3 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^2*b^6*c
^6*d^2 + 4*A^2*B^2*a^5*b^3*c^3*d^5 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a*b^7*
c^7*d + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a*b^7*c^7*d + 12*A^2*B*C*b^8*c*d^7 +
 6*A*B*C^2*b^8*c^7*d - 6*A*B*C^2*b^8*c*d^7 - 3*A^2*B*C*b^8*c^7*d + 24*A^2*B
*C*a*b^7*d^8 - 12*A*B*C^2*a*b^7*d^8 - 53*B^3*C*a^4*b^4*c^3*d^5 - 53*B*C^3*a
^4*b^4*c^3*d^5 - 32*B^3*C*a^2*b^6*c^3*d^5 - 32*B*C^3*a^2*b^6*c^3*d^5 - 18*B
^3*C*a^4*b^4*c^5*d^3 - 18*B*C^3*a^4*b^4*c^5*d^3 + 16*B^3*C*a^3*b^5*c^4*d^4
+ 16*B*C^3*a^3*b^5*c^4*d^4 + 12*B^3*C*a^5*b^3*c^4*d^4 - 12*B^3*C*a^3*b^5*c^
6*d^2 + 12*B^2*C^2*a*b^7*c^3*d^5 + 12*B*C^3*a^5*b^3*c^4*d^4 - 12*B*C^3*a^3*

b^5*c^6*d^2 + 8*B^3*C*a^3*b^5*c^2*d^6 + 8*B*C^3*a^3*b^5*c^2*d^6 - 6*B^3*C*a
^5*b^3*c^2*d^6 - 6*B^2*C^2*a^5*b^3*c*d^7 + 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*C^
3*a^5*b^3*c^2*d^6 - 3*B^3*C*a^6*b^2*c^3*d^5 - 3*B*C^3*a^6*b^2*c^3*d^5 - 175
*A^3*C*a^2*b^6*c^4*d^4 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a*b^7*c^3*
d^5 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^5*b^3*c^3*d^5 - 73*A*C^3*a^2*b
^6*c^4*d^4 - 66*A^2*C^2*a*b^7*c^5*d^3 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C^3
*a^4*b^4*c^4*d^4 - 30*A^3*C*a^5*b^3*c^3*d^5 + 30*A^3*C*a^4*b^4*c^4*d^4 + 27
*A*C^3*a^6*b^2*c^2*d^6 + 21*A*C^3*a^4*b^4*c^2*d^6 + 18*A^2*C^2*a^5*b^3*c*d^
7 - 18*A*C^3*a^4*b^4*c^6*d^2 - 16*A*C^3*a^2*b^6*c^2*d^6 - 15*A^3*C*a^4*b^4*
c^2*d^6 + 15*A^3*C*a^2*b^6*c^6*d^2 - 12*A^2*C^2*a^3*b^5*c*d^7 + 9*A^3*C*a^6
*b^2*c^2*d^6 + 9*A*C^3*a^2*b^6*c^6*d^2 - 80*A^3*B*a^3*b^5*c^2*d^6 - 80*A*B^
3*a^3*b^5*c^2*d^6 + 38*A^3*B*a^4*b^4*c^3*d^5 + 38*A*B^3*a^4*b^4*c^3*d^5 - 3
6*A^2*B^2*a*b^7*c^3*d^5 - 28*A^3*B*a^3*b^5*c^4*d^4 - 28*A^3*B*a^2*b^6*c^5*d
^3 - 28*A*B^3*a^3*b^5*c^4*d^4 - 28*A*B^3*a^2*b^6*c^5*d^3 + 20*A^3*B*a^2*b^6
*c^3*d^5 + 20*A*B^3*a^2*b^6*c^3*d^5 - 12*A^3*B*a^5*b^3*c^2*d^6 - 12*A^2*B^2
*a^5*b^3*c*d^7 - 12*A^2*B^2*a^3*b^5*c*d^7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A
*B^3*a^5*b^3*c^2*d^6 + 6*B^2*C^2*b^8*c^6*d^2 + 3*B^2*C^2*b^8*c^4*d^4 + 36*A
^2*C^2*b^8*c^4*d^4 + 27*A^2*C^2*b^8*c^2*d^6 - 18*A^2*C^2*b^8*c^6*d^2 + 33*A
^2*B^2*b^8*c^4*d^4 + 28*A^2*B^2*b^8*c^2*d^6 + 9*B^2*C^2*a^4*b^4*d^8 + 6*A^2
*B^2*b^8*c^6*d^2 + 4*B^2*C^2*a^2*b^6*d^8 + 3*B^2*C^2*a^6*b^2*d^8 - 30*A^2*C
^2*a^4*b^4*d^8 + 9*A^2*C^2*a^6*b^2*d^8 + 16*A^2*B^2*a^2*b^6*d^8 + 3*A^2*B^2
*a^4*b^4*d^8 + 6*C^4*a^5*b^3*c*d^7 + 4*C^4*a^3*b^5*c*d^7 - 2*C^4*a*b^7*c^5*
d^3 - 12*B^4*a^5*b^3*c*d^7 + 12*B^4*a*b^7*c^3*d^5 + 8*B^4*a*b^7*c^5*d^3 - 4
*B^4*a^3*b^5*c*d^7 - 48*A^4*a*b^7*c^3*d^5 - 20*A^4*a*b^7*c^5*d^3 - 8*A^4*a^
3*b^5*c*d^7 - 63*A^3*C*b^8*c^4*d^4 - 54*A^3*C*b^8*c^2*d^6 + 9*A^3*C*b^8*c^6
*d^2 + 9*A*C^3*b^8*c^6*d^2 - 3*A*C^3*b^8*c^4*d^4 - 28*A^3*B*b^8*c^5*d^3 - 2
8*A*B^3*b^8*c^5*d^3 - 18*A^3*B*b^8*c^3*d^5 - 18*A*B^3*b^8*c^3*d^5 - 10*B^3*
C*a^5*b^3*d^8 - 10*B*C^3*a^5*b^3*d^8 - 4*B^3*C*a^3*b^5*d^8 - 4*B*C^3*a^3*b^
5*d^8 + 23*A^3*C*a^4*b^4*d^8 - 18*A^3*C*a^2*b^6*d^8 + 11*A*C^3*a^4*b^4*d^8
- 9*A*C^3*a^6*b^2*d^8 + 6*A*C^3*a^2*b^6*d^8 - 3*A^3*C*a^6*b^2*d^8 - 20*A^3*
B*a^3*b^5*d^8 - 20*A*B^3*a^3*b^5*d^8 + 4*A^3*B*a^5*b^3*d^8 + 4*A*B^3*a^5*b^
3*d^8 + B^3*C*a^2*b^6*c^5*d^3 + B*C^3*a^2*b^6*c^5*d^3 + 6*C^4*a*b^7*c^7*d +
4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 3*B^3*C*b^8*c^7*d - 3*B*C^3*b^8*c
^7*d - 6*A^3*B*b^8*c*d^7 - 6*A*B^3*b^8*c*d^7 - 12*A^3*B*a*b^7*d^8 - 12*A*B^
3*a*b^7*d^8 + 30*C^4*a^5*b^3*c^3*d^5 + 19*C^4*a^2*b^6*c^4*d^4 - 9*C^4*a^6*b
^2*c^2*d^6 + 9*C^4*a^4*b^4*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*b^6*
c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^4*b^4*c^2*d^6 + 3*C^4*a^2*b^6*c^6
*d^2 + 28*B^4*a^3*b^5*c^5*d^3 + 27*B^4*a^4*b^4*c^2*d^6 - 17*B^4*a^4*b^4*c^4
*d^4 - 10*B^4*a^2*b^6*c^4*d^4 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c^2*d
^6 - 6*B^4*a^2*b^6*c^6*d^2 + 4*B^4*a^5*b^3*c^3*d^5 + 70*A^4*a^2*b^6*c^4*d^4
+ 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^4*b^4*c^2*d^6
+ B^2*C^2*b^8*c^2*d^6 - 18*A^3*C*b^8*d^8 + B^3*C*b^8*c^5*d^3 + B*C^3*b^8*c
^5*d^3 + 6*B^4*b^8*c^6*d^2 + 3*B^4*b^8*c^4*d^4 + 30*A^4*b^8*c^4*d^4 + 27*A^
4*b^8*c^2*d^6 + 3*C^4*a^6*b^2*d^8 + 8*B^4*a^4*b^4*d^8 + 4*B^4*a^2*b^6*d^8 +
12*A^4*a^2*b^6*d^8 - 5*A^4*a^4*b^4*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2*b^8

*d^8 + 9*A^4*b^8*d^8 + B^4*b^8*c^2*d^6 + C^4*a^4*b^4*d^8, f, k)*(root(640*a
^15*b*c^7*d^13*f^4 + 640*a*b^15*c^13*d^7*f^4 + 480*a^15*b*c^9*d^11*f^4 + 48
0*a^15*b*c^5*d^15*f^4 + 480*a*b^15*c^15*d^5*f^4 + 480*a*b^15*c^11*d^9*f^4 +
192*a^15*b*c^11*d^9*f^4 + 192*a^15*b*c^3*d^17*f^4 + 192*a^11*b^5*c*d^19*f^
4 + 192*a^5*b^11*c^19*d*f^4 + 192*a*b^15*c^17*d^3*f^4 + 192*a*b^15*c^9*d^11
*f^4 + 128*a^13*b^3*c*d^19*f^4 + 128*a^9*b^7*c*d^19*f^4 + 128*a^7*b^9*c^19*
d*f^4 + 128*a^3*b^13*c^19*d*f^4 + 32*a^15*b*c^13*d^7*f^4 + 32*a^9*b^7*c^19*
d*f^4 + 32*a^7*b^9*c*d^19*f^4 + 32*a*b^15*c^7*d^13*f^4 + 32*a^15*b*c*d^19*f
^4 + 32*a*b^15*c^19*d*f^4 - 47088*a^8*b^8*c^10*d^10*f^4 + 42432*a^9*b^7*c^9
*d^11*f^4 + 42432*a^7*b^9*c^11*d^9*f^4 + 39328*a^9*b^7*c^11*d^9*f^4 + 39328
*a^7*b^9*c^9*d^11*f^4 - 36912*a^8*b^8*c^12*d^8*f^4 - 36912*a^8*b^8*c^8*d^12
*f^4 - 34256*a^10*b^6*c^10*d^10*f^4 - 34256*a^6*b^10*c^10*d^10*f^4 - 31152*
a^10*b^6*c^8*d^12*f^4 - 31152*a^6*b^10*c^12*d^8*f^4 + 28128*a^9*b^7*c^7*d^1
3*f^4 + 28128*a^7*b^9*c^13*d^7*f^4 + 24160*a^11*b^5*c^9*d^11*f^4 + 24160*a^
5*b^11*c^11*d^9*f^4 - 23088*a^10*b^6*c^12*d^8*f^4 - 23088*a^6*b^10*c^8*d^12
*f^4 + 22272*a^9*b^7*c^13*d^7*f^4 + 22272*a^7*b^9*c^7*d^13*f^4 + 19072*a^11
*b^5*c^11*d^9*f^4 + 19072*a^5*b^11*c^9*d^11*f^4 + 18624*a^11*b^5*c^7*d^13*f
^4 + 18624*a^5*b^11*c^13*d^7*f^4 - 17328*a^8*b^8*c^14*d^6*f^4 - 17328*a^8*b
^8*c^6*d^14*f^4 - 17232*a^10*b^6*c^6*d^14*f^4 - 17232*a^6*b^10*c^14*d^6*f^4
 - 13520*a^12*b^4*c^8*d^12*f^4 - 13520*a^4*b^12*c^12*d^8*f^4 - 12464*a^12*b
^4*c^10*d^10*f^4 - 12464*a^4*b^12*c^10*d^10*f^4 + 10880*a^9*b^7*c^5*d^15*f^
4 + 10880*a^7*b^9*c^15*d^5*f^4 - 9072*a^10*b^6*c^14*d^6*f^4 - 9072*a^6*b^10
*c^6*d^14*f^4 + 8928*a^11*b^5*c^13*d^7*f^4 + 8928*a^5*b^11*c^7*d^13*f^4 - 8
880*a^12*b^4*c^6*d^14*f^4 - 8880*a^4*b^12*c^14*d^6*f^4 + 8480*a^11*b^5*c^5*
d^15*f^4 + 8480*a^5*b^11*c^15*d^5*f^4 + 7200*a^9*b^7*c^15*d^5*f^4 + 7200*a^
7*b^9*c^5*d^15*f^4 - 6912*a^12*b^4*c^12*d^8*f^4 - 6912*a^4*b^12*c^8*d^12*f^
4 + 6400*a^13*b^3*c^9*d^11*f^4 + 6400*a^3*b^13*c^11*d^9*f^4 + 5920*a^13*b^3
*c^7*d^13*f^4 + 5920*a^3*b^13*c^13*d^7*f^4 - 5392*a^10*b^6*c^4*d^16*f^4 - 5
392*a^6*b^10*c^16*d^4*f^4 - 4428*a^8*b^8*c^16*d^4*f^4 - 4428*a^8*b^8*c^4*d^
16*f^4 + 4128*a^13*b^3*c^11*d^9*f^4 + 4128*a^3*b^13*c^9*d^11*f^4 - 3328*a^1
2*b^4*c^4*d^16*f^4 - 3328*a^4*b^12*c^16*d^4*f^4 + 3264*a^13*b^3*c^5*d^15*f^
4 + 3264*a^3*b^13*c^15*d^5*f^4 - 2480*a^14*b^2*c^8*d^12*f^4 - 2480*a^2*b^14
*c^12*d^8*f^4 + 2240*a^11*b^5*c^15*d^5*f^4 + 2240*a^5*b^11*c^5*d^15*f^4 - 2
128*a^12*b^4*c^14*d^6*f^4 - 2128*a^4*b^12*c^6*d^14*f^4 + 2112*a^9*b^7*c^3*d
^17*f^4 + 2112*a^7*b^9*c^17*d^3*f^4 + 2048*a^11*b^5*c^3*d^17*f^4 + 2048*a^5
*b^11*c^17*d^3*f^4 - 2000*a^14*b^2*c^6*d^14*f^4 - 2000*a^2*b^14*c^14*d^6*f^
4 - 1792*a^10*b^6*c^16*d^4*f^4 - 1792*a^6*b^10*c^4*d^16*f^4 - 1776*a^14*b^2
*c^10*d^10*f^4 - 1776*a^2*b^14*c^10*d^10*f^4 + 1472*a^13*b^3*c^13*d^7*f^4 +
1472*a^3*b^13*c^7*d^13*f^4 + 1088*a^9*b^7*c^17*d^3*f^4 + 1088*a^7*b^9*c^3*
d^17*f^4 + 992*a^13*b^3*c^3*d^17*f^4 + 992*a^3*b^13*c^17*d^3*f^4 - 912*a^14
*b^2*c^4*d^16*f^4 - 912*a^2*b^14*c^16*d^4*f^4 - 768*a^10*b^6*c^2*d^18*f^4 -
768*a^6*b^10*c^18*d^2*f^4 - 688*a^14*b^2*c^12*d^8*f^4 - 688*a^2*b^14*c^8*d
^12*f^4 - 592*a^12*b^4*c^2*d^18*f^4 - 592*a^4*b^12*c^18*d^2*f^4 - 472*a^8*b
^8*c^18*d^2*f^4 - 472*a^8*b^8*c^2*d^18*f^4 - 280*a^12*b^4*c^16*d^4*f^4 - 28
0*a^4*b^12*c^4*d^16*f^4 + 224*a^13*b^3*c^15*d^5*f^4 + 224*a^11*b^5*c^17*d^3

*f^4 + 224*a^5*b^11*c^3*d^17*f^4 + 224*a^3*b^13*c^5*d^15*f^4 - 208*a^14*b^2
*c^2*d^18*f^4 - 208*a^2*b^14*c^18*d^2*f^4 - 112*a^14*b^2*c^14*d^6*f^4 - 112
*a^10*b^6*c^18*d^2*f^4 - 112*a^6*b^10*c^2*d^18*f^4 - 112*a^2*b^14*c^6*d^14*
f^4 - 80*b^16*c^14*d^6*f^4 - 60*b^16*c^16*d^4*f^4 - 60*b^16*c^12*d^8*f^4 -
24*b^16*c^18*d^2*f^4 - 24*b^16*c^10*d^10*f^4 - 4*b^16*c^8*d^12*f^4 - 80*a^1
6*c^6*d^14*f^4 - 60*a^16*c^8*d^12*f^4 - 60*a^16*c^4*d^16*f^4 - 24*a^16*c^10
*d^10*f^4 - 24*a^16*c^2*d^18*f^4 - 4*a^16*c^12*d^8*f^4 - 24*a^12*b^4*d^20*f
^4 - 16*a^14*b^2*d^20*f^4 - 16*a^10*b^6*d^20*f^4 - 4*a^8*b^8*d^20*f^4 - 24*
a^4*b^12*c^20*f^4 - 16*a^6*b^10*c^20*f^4 - 16*a^2*b^14*c^20*f^4 - 4*a^8*b^8
*c^20*f^4 - 4*b^16*c^20*f^4 - 4*a^16*d^20*f^4 + 56*A*C*a*b^11*c^13*d*f^2 -
48*A*C*a^11*b*c*d^13*f^2 + 48*A*C*a*b^11*c*d^13*f^2 + 5904*B*C*a^6*b^6*c^7*
d^7*f^2 - 5016*B*C*a^5*b^7*c^8*d^6*f^2 - 4608*B*C*a^7*b^5*c^6*d^8*f^2 - 451
2*B*C*a^5*b^7*c^6*d^8*f^2 - 4384*B*C*a^7*b^5*c^8*d^6*f^2 + 3056*B*C*a^8*b^4
*c^7*d^7*f^2 + 2256*B*C*a^4*b^8*c^7*d^7*f^2 - 1824*B*C*a^3*b^9*c^8*d^6*f^2
+ 1632*B*C*a^9*b^3*c^4*d^10*f^2 - 1400*B*C*a^8*b^4*c^3*d^11*f^2 - 1320*B*C*
a^4*b^8*c^11*d^3*f^2 - 1248*B*C*a^3*b^9*c^6*d^8*f^2 + 1152*B*C*a^3*b^9*c^10
*d^4*f^2 - 1072*B*C*a^9*b^3*c^6*d^8*f^2 + 1068*B*C*a^6*b^6*c^9*d^5*f^2 - 10
04*B*C*a^4*b^8*c^5*d^9*f^2 - 968*B*C*a^6*b^6*c^3*d^11*f^2 - 864*B*C*a^8*b^4
*c^5*d^9*f^2 - 828*B*C*a^4*b^8*c^9*d^5*f^2 - 792*B*C*a^4*b^8*c^3*d^11*f^2 -
792*B*C*a^2*b^10*c^11*d^3*f^2 - 776*B*C*a^9*b^3*c^8*d^6*f^2 + 688*B*C*a^7*
b^5*c^4*d^10*f^2 - 672*B*C*a^10*b^2*c^3*d^11*f^2 - 592*B*C*a^2*b^10*c^9*d^5
*f^2 + 544*B*C*a^10*b^2*c^7*d^7*f^2 - 492*B*C*a^2*b^10*c^5*d^9*f^2 + 480*B*
C*a^5*b^7*c^10*d^4*f^2 - 392*B*C*a^10*b^2*c^5*d^9*f^2 + 332*B*C*a^8*b^4*c^9
*d^5*f^2 - 328*B*C*a^6*b^6*c^11*d^3*f^2 + 320*B*C*a^9*b^3*c^2*d^12*f^2 + 27
2*B*C*a^3*b^9*c^12*d^2*f^2 - 248*B*C*a^5*b^7*c^4*d^10*f^2 - 248*B*C*a^2*b^1
0*c^3*d^11*f^2 - 208*B*C*a^7*b^5*c^10*d^4*f^2 - 192*B*C*a^5*b^7*c^2*d^12*f^
2 + 144*B*C*a^2*b^10*c^7*d^7*f^2 - 96*B*C*a^3*b^9*c^4*d^10*f^2 + 88*B*C*a^5
*b^7*c^12*d^2*f^2 - 72*B*C*a^8*b^4*c^11*d^3*f^2 + 48*B*C*a^9*b^3*c^10*d^4*f
^2 - 48*B*C*a^7*b^5*c^12*d^2*f^2 - 48*B*C*a^7*b^5*c^2*d^12*f^2 - 48*B*C*a^3
*b^9*c^2*d^12*f^2 - 12*B*C*a^10*b^2*c^9*d^5*f^2 + 4*B*C*a^6*b^6*c^5*d^9*f^2
+ 5824*A*C*a^7*b^5*c^5*d^9*f^2 - 4378*A*C*a^8*b^4*c^6*d^8*f^2 + 4296*A*C*a
^5*b^7*c^5*d^9*f^2 - 3912*A*C*a^6*b^6*c^6*d^8*f^2 - 3672*A*C*a^5*b^7*c^9*d^
5*f^2 + 3594*A*C*a^4*b^8*c^8*d^6*f^2 + 3236*A*C*a^6*b^6*c^8*d^6*f^2 + 2816*
A*C*a^9*b^3*c^5*d^9*f^2 + 2624*A*C*a^3*b^9*c^5*d^9*f^2 + 2432*A*C*a^7*b^5*c
^7*d^7*f^2 - 2366*A*C*a^8*b^4*c^4*d^10*f^2 + 2298*A*C*a^4*b^8*c^10*d^4*f^2
+ 1872*A*C*a^3*b^9*c^7*d^7*f^2 + 1848*A*C*a^6*b^6*c^10*d^4*f^2 - 1644*A*C*a
^6*b^6*c^4*d^10*f^2 - 1488*A*C*a^7*b^5*c^9*d^5*f^2 - 1408*A*C*a^3*b^9*c^9*d
^5*f^2 - 1308*A*C*a^4*b^8*c^6*d^8*f^2 + 1248*A*C*a^5*b^7*c^7*d^7*f^2 - 1012
*A*C*a^10*b^2*c^6*d^8*f^2 + 1008*A*C*a^7*b^5*c^3*d^11*f^2 + 992*A*C*a^5*b^7
*c^3*d^11*f^2 + 928*A*C*a^3*b^9*c^3*d^11*f^2 + 848*A*C*a^9*b^3*c^7*d^7*f^2
+ 636*A*C*a^2*b^10*c^8*d^6*f^2 - 628*A*C*a^10*b^2*c^4*d^10*f^2 - 600*A*C*a^
2*b^10*c^6*d^8*f^2 - 576*A*C*a^5*b^7*c^11*d^3*f^2 + 572*A*C*a^2*b^10*c^10*d
^4*f^2 + 464*A*C*a^8*b^4*c^8*d^6*f^2 + 304*A*C*a^6*b^6*c^2*d^12*f^2 - 304*A
*C*a^4*b^8*c^4*d^10*f^2 + 296*A*C*a^4*b^8*c^2*d^12*f^2 + 260*A*C*a^8*b^4*c^
10*d^4*f^2 - 232*A*C*a^9*b^3*c^9*d^5*f^2 - 232*A*C*a^2*b^10*c^12*d^2*f^2 +

228*A*C*a^10*b^2*c^2*d^12*f^2 - 188*A*C*a^2*b^10*c^4*d^10*f^2 + 144*A*C*a^3
*b^9*c^11*d^3*f^2 + 116*A*C*a^6*b^6*c^12*d^2*f^2 + 112*A*C*a^9*b^3*c^3*d^11
*f^2 - 112*A*C*a^7*b^5*c^11*d^3*f^2 + 92*A*C*a^10*b^2*c^8*d^6*f^2 + 74*A*C*
a^4*b^8*c^12*d^2*f^2 + 62*A*C*a^8*b^4*c^2*d^12*f^2 + 40*A*C*a^2*b^10*c^2*d^
12*f^2 - 7008*A*B*a^6*b^6*c^7*d^7*f^2 - 4032*A*B*a^4*b^8*c^7*d^7*f^2 + 3952
*A*B*a^7*b^5*c^8*d^6*f^2 + 3648*A*B*a^5*b^7*c^8*d^6*f^2 - 3392*A*B*a^8*b^4*
c^7*d^7*f^2 + 3264*A*B*a^7*b^5*c^6*d^8*f^2 - 2992*A*B*a^5*b^7*c^4*d^10*f^2
- 2368*A*B*a^7*b^5*c^4*d^10*f^2 - 2304*A*B*a^3*b^9*c^4*d^10*f^2 - 1968*A*B*
a^6*b^6*c^9*d^5*f^2 - 1872*A*B*a^9*b^3*c^4*d^10*f^2 - 1728*A*B*a^2*b^10*c^7
*d^7*f^2 + 1712*A*B*a^8*b^4*c^3*d^11*f^2 + 1536*A*B*a^5*b^7*c^6*d^8*f^2 - 1
536*A*B*a^3*b^9*c^10*d^4*f^2 - 1392*A*B*a^5*b^7*c^2*d^12*f^2 + 1328*A*B*a^6
*b^6*c^3*d^11*f^2 - 1104*A*B*a^3*b^9*c^2*d^12*f^2 - 1056*A*B*a^3*b^9*c^6*d^
8*f^2 + 976*A*B*a^9*b^3*c^6*d^8*f^2 + 960*A*B*a^4*b^8*c^11*d^3*f^2 + 936*A*
B*a^8*b^4*c^5*d^9*f^2 - 912*A*B*a^5*b^7*c^10*d^4*f^2 + 848*A*B*a^9*b^3*c^8*
d^6*f^2 - 816*A*B*a^7*b^5*c^2*d^12*f^2 + 816*A*B*a^4*b^8*c^3*d^11*f^2 + 768
*A*B*a^10*b^2*c^3*d^11*f^2 + 672*A*B*a^3*b^9*c^8*d^6*f^2 - 632*A*B*a^8*b^4*
c^9*d^5*f^2 - 608*A*B*a^2*b^10*c^9*d^5*f^2 - 552*A*B*a^4*b^8*c^9*d^5*f^2 -
544*A*B*a^10*b^2*c^7*d^7*f^2 - 480*A*B*a^2*b^10*c^5*d^9*f^2 + 464*A*B*a^10*
b^2*c^5*d^9*f^2 - 464*A*B*a^9*b^3*c^2*d^12*f^2 + 432*A*B*a^2*b^10*c^11*d^3*
f^2 - 368*A*B*a^3*b^9*c^12*d^2*f^2 - 256*A*B*a^6*b^6*c^5*d^9*f^2 - 208*A*B*
a^5*b^7*c^12*d^2*f^2 + 176*A*B*a^4*b^8*c^5*d^9*f^2 + 112*A*B*a^7*b^5*c^10*d
^4*f^2 + 112*A*B*a^6*b^6*c^11*d^3*f^2 - 16*A*B*a^2*b^10*c^3*d^11*f^2 - 576*
B*C*a*b^11*c^8*d^6*f^2 + 400*B*C*a^11*b*c^4*d^10*f^2 - 288*B*C*a*b^11*c^6*d
^8*f^2 - 176*B*C*a^11*b*c^6*d^8*f^2 + 128*B*C*a*b^11*c^10*d^4*f^2 - 108*B*C
*a^4*b^8*c*d^13*f^2 - 104*B*C*a*b^11*c^4*d^10*f^2 - 92*B*C*a^4*b^8*c^13*d*f
^2 - 60*B*C*a^8*b^4*c*d^13*f^2 - 60*B*C*a^6*b^6*c*d^13*f^2 + 48*B*C*a^11*b*
c^2*d^12*f^2 - 40*B*C*a^2*b^10*c*d^13*f^2 - 28*B*C*a^2*b^10*c^13*d*f^2 - 24
*B*C*a*b^11*c^12*d^2*f^2 + 20*B*C*a^10*b^2*c*d^13*f^2 - 16*B*C*a*b^11*c^2*d
^12*f^2 + 12*B*C*a^6*b^6*c^13*d*f^2 + 912*A*C*a*b^11*c^7*d^7*f^2 + 808*A*C*
a*b^11*c^5*d^9*f^2 + 432*A*C*a^11*b*c^5*d^9*f^2 + 336*A*C*a*b^11*c^3*d^11*f
^2 + 224*A*C*a*b^11*c^11*d^3*f^2 - 112*A*C*a^11*b*c^3*d^11*f^2 + 112*A*C*a^
3*b^9*c*d^13*f^2 - 88*A*C*a^9*b^3*c*d^13*f^2 + 80*A*C*a^3*b^9*c^13*d*f^2 +
56*A*C*a^5*b^7*c*d^13*f^2 + 48*A*C*a*b^11*c^9*d^5*f^2 - 40*A*C*a^5*b^7*c^13
*d*f^2 - 16*A*C*a^11*b*c^7*d^7*f^2 + 16*A*C*a^7*b^5*c*d^13*f^2 - 496*A*B*a*
b^11*c^4*d^10*f^2 - 400*A*B*a^11*b*c^4*d^10*f^2 + 288*A*B*a*b^11*c^8*d^6*f^
2 - 288*A*B*a*b^11*c^6*d^8*f^2 - 272*A*B*a*b^11*c^2*d^12*f^2 + 240*A*B*a^6*
b^6*c*d^13*f^2 - 224*A*B*a*b^11*c^10*d^4*f^2 + 192*A*B*a^8*b^4*c*d^13*f^2 +
192*A*B*a^4*b^8*c*d^13*f^2 + 176*A*B*a^11*b*c^6*d^8*f^2 + 104*A*B*a^4*b^8*
c^13*d*f^2 - 48*A*B*a^11*b*c^2*d^12*f^2 + 16*A*B*a^10*b^2*c*d^13*f^2 + 16*A
*B*a^2*b^10*c^13*d*f^2 + 16*A*B*a^2*b^10*c*d^13*f^2 - 112*B*C*b^12*c^11*d^3
*f^2 + 4*B*C*b^12*c^5*d^9*f^2 + 150*A*C*b^12*c^10*d^4*f^2 - 80*B*C*a^12*c^3
*d^11*f^2 + 66*A*C*b^12*c^8*d^6*f^2 - 30*A*C*b^12*c^12*d^2*f^2 + 24*B*C*a^1
2*c^5*d^9*f^2 - 12*A*C*b^12*c^4*d^10*f^2 - 576*A*B*b^12*c^7*d^7*f^2 - 432*A
*B*b^12*c^9*d^5*f^2 - 400*A*B*b^12*c^5*d^9*f^2 - 144*A*B*b^12*c^3*d^11*f^2
- 96*B*C*a^7*b^5*d^14*f^2 - 72*B*C*a^5*b^7*d^14*f^2 - 66*A*C*a^12*c^4*d^10*

f^2 + 54*A*C*a^12*c^2*d^12*f^2 - 32*A*B*b^12*c^11*d^3*f^2 - 24*B*C*a^9*b^3*
d^14*f^2 - 16*B*C*a^3*b^9*d^14*f^2 + 2*A*C*a^12*c^6*d^8*f^2 + 116*A*C*a^6*b
^6*d^14*f^2 + 100*A*C*a^4*b^8*d^14*f^2 + 80*A*B*a^12*c^3*d^11*f^2 + 24*A*C*
a^2*b^10*d^14*f^2 - 24*A*B*a^12*c^5*d^9*f^2 + 22*A*C*a^8*b^4*d^14*f^2 + 16*
B*C*a^3*b^9*c^14*f^2 + 8*A*C*a^10*b^2*d^14*f^2 - 192*A*B*a^5*b^7*d^14*f^2 -
176*A*B*a^3*b^9*d^14*f^2 - 48*A*B*a^7*b^5*d^14*f^2 - 28*A*C*a^2*b^10*c^14*
f^2 + 2*A*C*a^4*b^8*c^14*f^2 - 16*A*B*a^3*b^9*c^14*f^2 + 2508*C^2*a^6*b^6*c
^6*d^8*f^2 + 2376*C^2*a^5*b^7*c^9*d^5*f^2 + 2357*C^2*a^8*b^4*c^6*d^8*f^2 -
2048*C^2*a^7*b^5*c^5*d^9*f^2 + 1304*C^2*a^3*b^9*c^9*d^5*f^2 + 1303*C^2*a^8*
b^4*c^4*d^10*f^2 + 1212*C^2*a^6*b^6*c^4*d^10*f^2 - 1203*C^2*a^4*b^8*c^8*d^6
*f^2 - 1192*C^2*a^9*b^3*c^5*d^9*f^2 + 1062*C^2*a^4*b^8*c^6*d^8*f^2 + 984*C^
2*a^7*b^5*c^9*d^5*f^2 - 952*C^2*a^6*b^6*c^8*d^6*f^2 + 768*C^2*a^5*b^7*c^7*d
^7*f^2 - 681*C^2*a^4*b^8*c^10*d^4*f^2 - 672*C^2*a^5*b^7*c^5*d^9*f^2 - 480*C
^2*a^6*b^6*c^10*d^4*f^2 + 458*C^2*a^10*b^2*c^6*d^8*f^2 - 448*C^2*a^7*b^5*c^
7*d^7*f^2 + 422*C^2*a^4*b^8*c^4*d^10*f^2 + 372*C^2*a^2*b^10*c^6*d^8*f^2 + 3
60*C^2*a^5*b^7*c^11*d^3*f^2 + 312*C^2*a^3*b^9*c^7*d^7*f^2 + 278*C^2*a^10*b^
2*c^4*d^10*f^2 - 232*C^2*a^9*b^3*c^7*d^7*f^2 + 194*C^2*a^2*b^10*c^12*d^2*f^
2 + 176*C^2*a^9*b^3*c^9*d^5*f^2 + 152*C^2*a^5*b^7*c^3*d^11*f^2 + 124*C^2*a^
2*b^10*c^4*d^10*f^2 - 120*C^2*a^7*b^5*c^3*d^11*f^2 - 114*C^2*a^10*b^2*c^2*d
^12*f^2 - 102*C^2*a^2*b^10*c^8*d^6*f^2 + 101*C^2*a^4*b^8*c^12*d^2*f^2 + 100
*C^2*a^6*b^6*c^2*d^12*f^2 - 88*C^2*a^3*b^9*c^5*d^9*f^2 + 77*C^2*a^8*b^4*c^2
*d^12*f^2 + 72*C^2*a^3*b^9*c^11*d^3*f^2 - 64*C^2*a^10*b^2*c^8*d^6*f^2 + 64*
C^2*a^3*b^9*c^3*d^11*f^2 - 58*C^2*a^2*b^10*c^10*d^4*f^2 + 56*C^2*a^7*b^5*c^
11*d^3*f^2 + 56*C^2*a^6*b^6*c^12*d^2*f^2 + 40*C^2*a^9*b^3*c^3*d^11*f^2 + 36
*C^2*a^8*b^4*c^12*d^2*f^2 + 32*C^2*a^4*b^8*c^2*d^12*f^2 + 26*C^2*a^8*b^4*c^
10*d^4*f^2 + 16*C^2*a^2*b^10*c^2*d^12*f^2 + 2*C^2*a^8*b^4*c^8*d^6*f^2 + 227
7*B^2*a^4*b^8*c^8*d^6*f^2 + 2144*B^2*a^7*b^5*c^5*d^9*f^2 - 2112*B^2*a^5*b^7
*c^9*d^5*f^2 + 2028*B^2*a^6*b^6*c^8*d^6*f^2 - 1671*B^2*a^8*b^4*c^6*d^8*f^2
+ 1275*B^2*a^4*b^8*c^10*d^4*f^2 + 1176*B^2*a^5*b^7*c^5*d^9*f^2 + 1096*B^2*a
^9*b^3*c^5*d^9*f^2 - 1044*B^2*a^6*b^6*c^6*d^8*f^2 + 984*B^2*a^6*b^6*c^10*d^
4*f^2 - 968*B^2*a^3*b^9*c^9*d^5*f^2 - 888*B^2*a^7*b^5*c^9*d^5*f^2 + 672*B^2
*a^7*b^5*c^7*d^7*f^2 + 664*B^2*a^3*b^9*c^5*d^9*f^2 - 649*B^2*a^8*b^4*c^4*d^
10*f^2 + 618*B^2*a^2*b^10*c^8*d^6*f^2 + 514*B^2*a^4*b^8*c^4*d^10*f^2 + 460*
B^2*a^6*b^6*c^2*d^12*f^2 + 422*B^2*a^8*b^4*c^8*d^6*f^2 + 406*B^2*a^2*b^10*c
^10*d^4*f^2 - 382*B^2*a^10*b^2*c^6*d^8*f^2 + 368*B^2*a^4*b^8*c^2*d^12*f^2 -
312*B^2*a^5*b^7*c^11*d^3*f^2 + 312*B^2*a^3*b^9*c^7*d^7*f^2 + 248*B^2*a^9*b
^3*c^7*d^7*f^2 + 245*B^2*a^8*b^4*c^2*d^12*f^2 - 192*B^2*a^5*b^7*c^7*d^7*f^2
- 184*B^2*a^9*b^3*c^3*d^11*f^2 + 182*B^2*a^10*b^2*c^2*d^12*f^2 + 176*B^2*a
^3*b^9*c^3*d^11*f^2 + 174*B^2*a^4*b^8*c^6*d^8*f^2 - 170*B^2*a^10*b^2*c^4*d^
10*f^2 - 152*B^2*a^9*b^3*c^9*d^5*f^2 + 152*B^2*a^2*b^10*c^4*d^10*f^2 + 142*
B^2*a^8*b^4*c^10*d^4*f^2 - 90*B^2*a^2*b^10*c^12*d^2*f^2 + 88*B^2*a^2*b^10*c
^2*d^12*f^2 + 84*B^2*a^10*b^2*c^8*d^6*f^2 + 84*B^2*a^2*b^10*c^6*d^8*f^2 + 6
0*B^2*a^6*b^6*c^12*d^2*f^2 - 56*B^2*a^7*b^5*c^11*d^3*f^2 + 53*B^2*a^4*b^8*c
^12*d^2*f^2 + 24*B^2*a^7*b^5*c^3*d^11*f^2 + 24*B^2*a^6*b^6*c^4*d^10*f^2 + 2
4*B^2*a^3*b^9*c^11*d^3*f^2 - 8*B^2*a^5*b^7*c^3*d^11*f^2 + 4566*A^2*a^4*b^8*

c^6*d^8*f^2 + 4284*A^2*a^6*b^6*c^6*d^8*f^2 - 3776*A^2*a^7*b^5*c^5*d^9*f^2 - 3624*A^2*a^5*b^7*c^5*d^9*f^2 + 3122*A^2*a^4*b^8*c^4*d^10*f^2 + 3108*A^2*a^2*b^10*c^6*d^8*f^2 + 2741*A^2*a^8*b^4*c^6*d^8*f^2 + 2592*A^2*a^6*b^6*c^4*d^10*f^2 - 2536*A^2*a^3*b^9*c^5*d^9*f^2 + 2224*A^2*a^2*b^10*c^4*d^10*f^2 - 2184*A^2*a^3*b^9*c^7*d^7*f^2 - 2016*A^2*a^5*b^7*c^7*d^7*f^2 - 1984*A^2*a^7*b^5*c^7*d^7*f^2 + 1626*A^2*a^2*b^10*c^8*d^6*f^2 - 1624*A^2*a^9*b^3*c^5*d^9*f^2 + 1603*A^2*a^8*b^4*c^4*d^10*f^2 + 1296*A^2*a^5*b^7*c^9*d^5*f^2 - 1144*A^2*a^5*b^7*c^3*d^11*f^2 - 992*A^2*a^3*b^9*c^3*d^11*f^2 + 968*A^2*a^4*b^8*c^2*d^12*f^2 - 888*A^2*a^7*b^5*c^3*d^11*f^2 + 849*A^2*a^4*b^8*c^8*d^6*f^2 + 808*A^2*a^2*b^10*c^2*d^12*f^2 - 616*A^2*a^9*b^3*c^7*d^7*f^2 + 554*A^2*a^10*b^2*c^6*d^8*f^2 + 504*A^2*a^7*b^5*c^9*d^5*f^2 - 504*A^2*a^6*b^6*c^10*d^4*f^2 + 460*A^2*a^6*b^6*c^2*d^12*f^2 + 350*A^2*a^10*b^2*c^4*d^10*f^2 + 350*A^2*a^2*b^10*c^10*d^4*f^2 - 321*A^2*a^4*b^8*c^10*d^4*f^2 + 216*A^2*a^5*b^7*c^11*d^3*f^2 - 216*A^2*a^3*b^9*c^11*d^3*f^2 + 182*A^2*a^2*b^10*c^12*d^2*f^2 - 152*A^2*a^9*b^3*c^3*d^11*f^2 - 124*A^2*a^6*b^6*c^8*d^6*f^2 - 114*A^2*a^10*b^2*c^2*d^12*f^2 + 104*A^2*a^3*b^9*c^9*d^5*f^2 + 77*A^2*a^8*b^4*c^2*d^12*f^2 + 74*A^2*a^8*b^4*c^8*d^6*f^2 - 70*A^2*a^8*b^4*c^10*d^4*f^2 + 56*A^2*a^9*b^3*c^9*d^5*f^2 + 56*A^2*a^7*b^5*c^11*d^3*f^2 + 41*A^2*a^4*b^8*c^12*d^2*f^2 - 28*A^2*a^10*b^2*c^8*d^6*f^2 - 28*A^2*a^6*b^6*c^12*d^2*f^2 + 12*B*C*b^12*c^13*d*f^2 + 24*B*C*a^12*c*d^13*f^2 - 24*A*B*b^12*c^13*d*f^2 - 24*A*B*b^12*c*d^13*f^2 - 16*B*C*a^11*b*d^14*f^2 - 24*A*B*a^12*c*d^13*f^2 - 16*B*C*a*b^11*c^14*f^2 - 48*A*B*a*b^11*d^14*f^2 + 16*A*B*a^11*b*d^14*f^2 + 16*A*B*a*b^11*c^14*f^2 - 216*C^2*a^11*b*c^5*d^9*f^2 + 216*C^2*a*b^11*c^9*d^5*f^2 + 56*C^2*a^11*b*c^3*d^11*f^2 + 56*C^2*a^9*b^3*c*d^13*f^2 + 56*C^2*a^5*b^7*c*d^13*f^2 + 40*C^2*a^7*b^5*c*d^13*f^2 - 40*C^2*a*b^11*c^11*d^3*f^2 + 32*C^2*a^5*b^7*c^13*d*f^2 - 24*C^2*a*b^11*c^7*d^7*f^2 - 16*C^2*a^3*b^9*c^13*d*f^2 + 16*C^2*a^3*b^9*c*d^13*f^2 + 8*C^2*a^11*b*c^7*d^7*f^2 - 8*C^2*a*b^11*c^5*d^9*f^2 + 264*B^2*a*b^11*c^7*d^7*f^2 + 224*B^2*a*b^11*c^5*d^9*f^2 + 168*B^2*a^11*b*c^5*d^9*f^2 - 112*B^2*a^9*b^3*c*d^13*f^2 - 104*B^2*a^11*b*c^3*d^11*f^2 - 104*B^2*a^7*b^5*c*d^13*f^2 + 96*B^2*a*b^11*c^3*d^11*f^2 + 88*B^2*a*b^11*c^11*d^3*f^2 - 72*B^2*a*b^11*c^9*d^5*f^2 - 64*B^2*a^5*b^7*c*d^13*f^2 + 32*B^2*a^3*b^9*c^13*d*f^2 - 24*B^2*a^11*b*c^7*d^7*f^2 - 24*B^2*a^5*b^7*c^13*d*f^2 + 16*B^2*a^3*b^9*c*d^13*f^2 - 888*A^2*a*b^11*c^7*d^7*f^2 - 800*A^2*a*b^11*c^5*d^9*f^2 - 336*A^2*a*b^11*c^3*d^11*f^2 - 264*A^2*a*b^11*c^9*d^5*f^2 - 216*A^2*a^11*b*c^5*d^9*f^2 - 184*A^2*a*b^11*c^11*d^3*f^2 - 128*A^2*a^3*b^9*c*d^13*f^2 - 112*A^2*a^5*b^7*c*d^13*f^2 - 64*A^2*a^3*b^9*c^13*d*f^2 + 56*A^2*a^11*b*c^3*d^11*f^2 - 56*A^2*a^7*b^5*c*d^13*f^2 + 32*A^2*a^9*b^3*c*d^13*f^2 + 8*A^2*a^11*b*c^7*d^7*f^2 + 8*A^2*a^5*b^7*c^13*d*f^2 + 24*C^2*a^11*b*c*d^13*f^2 - 16*C^2*a*b^11*c^13*d*f^2 - 40*B^2*a^11*b*c*d^13*f^2 + 24*B^2*a*b^11*c^13*d*f^2 + 16*B^2*a*b^11*c*d^13*f^2 - 48*A^2*a*b^11*c*d^13*f^2 - 40*A^2*a*b^11*c^13*d*f^2 + 24*A^2*a^11*b*c*d^13*f^2 - 6*A*C*a^12*d^14*f^2 + 2*A*C*b^12*c^14*f^2 + 33*C^2*b^12*c^12*d^2*f^2 - 27*C^2*b^12*c^10*d^4*f^2 + 3*C^2*b^12*c^8*d^6*f^2 + 117*B^2*b^12*c^10*d^4*f^2 + 111*B^2*b^12*c^8*d^6*f^2 + 72*B^2*b^12*c^6*d^8*f^2 + 33*C^2*a^12*c^4*d^10*f^2 - 27*C^2*a^12*c^2*d^12*f^2 + 24*B^2*b^12*c^4*d^10*f^2 + 4*B^2*b^12*c^2*d^12*f^2 - 3*B^2*b^12*c^12*d^2*f^2

- C^2*a^12*c^6*d^8*f^2 + 720*A^2*b^12*c^6*d^8*f^2 + 552*A^2*b^12*c^4*d^10*
f^2 + 471*A^2*b^12*c^8*d^6*f^2 + 216*A^2*b^12*c^2*d^12*f^2 + 93*A^2*b^12*c^
10*d^4*f^2 + 33*B^2*a^12*c^2*d^12*f^2 + 33*A^2*b^12*c^12*d^2*f^2 + 31*C^2*a
^8*b^4*d^14*f^2 - 27*B^2*a^12*c^4*d^10*f^2 + 20*C^2*a^6*b^6*d^14*f^2 + 4*C^
2*a^4*b^8*d^14*f^2 + 3*B^2*a^12*c^6*d^8*f^2 + 2*C^2*a^10*b^2*d^14*f^2 + 80*
B^2*a^6*b^6*d^14*f^2 + 64*B^2*a^4*b^8*d^14*f^2 + 33*A^2*a^12*c^4*d^10*f^2 +
31*B^2*a^8*b^4*d^14*f^2 - 27*A^2*a^12*c^2*d^12*f^2 + 16*B^2*a^2*b^10*d^14*
f^2 + 14*C^2*a^2*b^10*c^14*f^2 + 14*B^2*a^10*b^2*d^14*f^2 - C^2*a^4*b^8*c^1
4*f^2 - A^2*a^12*c^6*d^8*f^2 + 120*A^2*a^2*b^10*d^14*f^2 + 112*A^2*a^4*b^8*
d^14*f^2 - 17*A^2*a^8*b^4*d^14*f^2 - 10*B^2*a^2*b^10*c^14*f^2 - 10*A^2*a^10
*b^2*d^14*f^2 + 8*A^2*a^6*b^6*d^14*f^2 + 3*B^2*a^4*b^8*c^14*f^2 + 14*A^2*a^
2*b^10*c^14*f^2 - A^2*a^4*b^8*c^14*f^2 + 3*C^2*a^12*d^14*f^2 - C^2*b^12*c^1
4*f^2 + 36*A^2*b^12*d^14*f^2 + 3*B^2*b^12*c^14*f^2 - B^2*a^12*d^14*f^2 + 3*
A^2*a^12*d^14*f^2 - A^2*b^12*c^14*f^2 - 44*A*B*C*a*b^9*c^10*d*f + 3816*A*B*
C*a^5*b^5*c^4*d^7*f + 2920*A*B*C*a^2*b^8*c^5*d^6*f - 2736*A*B*C*a^3*b^7*c^6
*d^5*f - 2672*A*B*C*a^4*b^6*c^3*d^8*f + 1996*A*B*C*a^4*b^6*c^7*d^4*f - 1412
*A*B*C*a^6*b^4*c^5*d^6*f + 1120*A*B*C*a^3*b^7*c^2*d^9*f + 1080*A*B*C*a^2*b^
8*c^7*d^4*f + 1040*A*B*C*a^5*b^5*c^2*d^9*f + 684*A*B*C*a^4*b^6*c^5*d^6*f +
592*A*B*C*a^3*b^7*c^4*d^7*f - 560*A*B*C*a^7*b^3*c^2*d^9*f - 448*A*B*C*a^2*b
^8*c^3*d^8*f - 400*A*B*C*a^5*b^5*c^8*d^3*f - 398*A*B*C*a^2*b^8*c^9*d^2*f -
312*A*B*C*a^6*b^4*c^3*d^8*f + 166*A*B*C*a^8*b^2*c^3*d^8*f + 136*A*B*C*a^5*b
^5*c^6*d^5*f + 128*A*B*C*a^7*b^3*c^6*d^5*f - 100*A*B*C*a^6*b^4*c^7*d^4*f +
64*A*B*C*a^7*b^3*c^4*d^7*f - 64*A*B*C*a^4*b^6*c^9*d^2*f - 32*A*B*C*a^3*b^7*
c^8*d^3*f - 16*A*B*C*a^8*b^2*c^5*d^6*f - 1312*A*B*C*a*b^9*c^4*d^7*f + 996*A
*B*C*a*b^9*c^8*d^3*f + 728*A*B*C*a^6*b^4*c*d^10*f - 624*A*B*C*a*b^9*c^6*d^5
*f - 584*A*B*C*a^2*b^8*c*d^10*f - 512*A*B*C*a^4*b^6*c*d^10*f - 320*A*B*C*a*
b^9*c^2*d^9*f - 98*A*B*C*a^8*b^2*c*d^10*f + 36*A*B*C*a^9*b*c^2*d^9*f + 32*A
*B*C*a^3*b^7*c^10*d*f - 16*A*B*C*a^9*b*c^4*d^7*f + 46*B*C^2*a*b^9*c^10*d*f
- 16*B^2*C*a*b^9*c*d^10*f - 2*B^2*C*a^9*b*c*d^10*f + 312*A^2*C*a*b^9*c*d^10
*f - 48*A*C^2*a*b^9*c*d^10*f - 6*A^2*C*a^9*b*c*d^10*f + 6*A*C^2*a^9*b*c*d^1
0*f + 208*A*B^2*a*b^9*c*d^10*f - 2*A^2*B*a*b^9*c^10*d*f + 2*A*B^2*a^9*b*c*d
^10*f - 480*A*B*C*b^10*c^7*d^4*f + 78*A*B*C*b^10*c^9*d^2*f - 64*A*B*C*b^10*
c^5*d^6*f + 2*A*B*C*a^10*c^3*d^8*f - 224*A*B*C*a^5*b^5*d^11*f + 80*A*B*C*a^
7*b^3*d^11*f - 32*A*B*C*a^3*b^7*d^11*f + 2*A*B*C*a^2*b^8*c^11*f - 1692*B*C^
2*a^5*b^5*c^4*d^7*f - 1500*B^2*C*a^5*b^5*c^5*d^6*f - 1464*B^2*C*a^3*b^7*c^5
*d^6*f + 1426*B*C^2*a^6*b^4*c^5*d^6*f - 1158*B^2*C*a^6*b^4*c^4*d^7*f + 1152
*B*C^2*a^3*b^7*c^6*d^5*f + 1026*B^2*C*a^4*b^6*c^6*d^5*f - 974*B*C^2*a^4*b^6
*c^7*d^4*f + 960*B^2*C*a^5*b^5*c^3*d^8*f - 884*B*C^2*a^2*b^8*c^5*d^6*f - 76
4*B^2*C*a^5*b^5*c^7*d^4*f + 752*B^2*C*a^2*b^8*c^4*d^7*f - 752*B*C^2*a^3*b^7
*c^4*d^7*f + 738*B^2*C*a^4*b^6*c^4*d^7*f - 688*B^2*C*a^6*b^4*c^2*d^9*f - 67
5*B^2*C*a^2*b^8*c^8*d^3*f + 560*B*C^2*a^5*b^5*c^8*d^3*f + 496*B*C^2*a^7*b^3
*c^2*d^9*f + 496*B*C^2*a^4*b^6*c^3*d^8*f - 468*B*C^2*a^2*b^8*c^7*d^4*f + 45
6*B^2*C*a^7*b^3*c^3*d^8*f - 452*B^2*C*a^4*b^6*c^8*d^3*f - 416*B*C^2*a^3*b^7
*c^2*d^9*f + 378*B*C^2*a^4*b^6*c^5*d^6*f + 376*B*C^2*a^3*b^7*c^8*d^3*f - 36
0*B^2*C*a^2*b^8*c^6*d^5*f + 355*B*C^2*a^2*b^8*c^9*d^2*f + 346*B^2*C*a^6*b^4

*c^6*d^5*f - 320*B^2*C*a^4*b^6*c^2*d^9*f + 268*B^2*C*a^2*b^8*c^2*d^9*f + 21
6*B^2*C*a^3*b^7*c^7*d^4*f - 203*B*C^2*a^8*b^2*c^3*d^8*f - 184*B*C^2*a^7*b^3
*c^6*d^5*f + 170*B*C^2*a^6*b^4*c^7*d^4*f + 160*B^2*C*a^7*b^3*c^5*d^6*f - 16
0*B*C^2*a^5*b^5*c^2*d^9*f - 140*B^2*C*a^8*b^2*c^4*d^7*f - 136*B*C^2*a^2*b^8
*c^3*d^8*f + 112*B^2*C*a^3*b^7*c^9*d^2*f + 91*B^2*C*a^8*b^2*c^2*d^9*f + 88*
B*C^2*a^7*b^3*c^4*d^7*f + 72*B^2*C*a^6*b^4*c^8*d^3*f - 64*B^2*C*a^3*b^7*c^3
*d^8*f - 60*B*C^2*a^6*b^4*c^3*d^8*f + 56*B*C^2*a^4*b^6*c^9*d^2*f + 52*B*C^2
*a^5*b^5*c^6*d^5*f - 48*B^2*C*a^7*b^3*c^7*d^4*f + 48*B^2*C*a^5*b^5*c^9*d^2*
f + 44*B*C^2*a^8*b^2*c^5*d^6*f - 36*B*C^2*a^6*b^4*c^9*d^2*f + 12*B^2*C*a^8*
b^2*c^6*d^5*f - 2958*A^2*C*a^4*b^6*c^4*d^7*f - 1932*A^2*C*a^2*b^8*c^4*d^7*f
 + 1848*A^2*C*a^3*b^7*c^5*d^6*f + 1728*A^2*C*a^3*b^7*c^3*d^8*f + 1524*A^2*C
*a^5*b^5*c^5*d^6*f + 1374*A*C^2*a^4*b^6*c^4*d^7*f - 1272*A*C^2*a^3*b^7*c^5*
d^6*f - 1236*A*C^2*a^5*b^5*c^5*d^6*f + 1116*A*C^2*a^2*b^8*c^4*d^7*f - 1110*
A^2*C*a^4*b^6*c^6*d^5*f + 1038*A*C^2*a^4*b^6*c^6*d^5*f - 768*A^2*C*a^2*b^8*
c^2*d^9*f - 696*A^2*C*a^3*b^7*c^7*d^4*f - 666*A*C^2*a^6*b^4*c^4*d^7*f + 564
*A^2*C*a^2*b^8*c^6*d^5*f - 564*A*C^2*a^5*b^5*c^7*d^4*f - 555*A*C^2*a^2*b^8*
c^8*d^3*f + 519*A^2*C*a^2*b^8*c^8*d^3*f - 480*A*C^2*a^3*b^7*c^3*d^8*f + 456
*A*C^2*a^5*b^5*c^3*d^8*f - 420*A*C^2*a^6*b^4*c^2*d^9*f + 408*A*C^2*a^3*b^7*
c^7*d^4*f + 408*A*C^2*a^2*b^8*c^2*d^9*f + 348*A^2*C*a^6*b^4*c^2*d^9*f - 348
*A*C^2*a^2*b^8*c^6*d^5*f + 342*A*C^2*a^6*b^4*c^6*d^5*f - 336*A*C^2*a^4*b^6*
c^8*d^3*f + 324*A^2*C*a^5*b^5*c^7*d^4*f - 312*A^2*C*a^4*b^6*c^2*d^9*f + 264
*A^2*C*a^4*b^6*c^8*d^3*f + 240*A*C^2*a^7*b^3*c^5*d^6*f + 195*A*C^2*a^8*b^2*
c^2*d^9*f - 174*A^2*C*a^6*b^4*c^6*d^5*f + 144*A*C^2*a^3*b^7*c^9*d^2*f - 123
*A^2*C*a^8*b^2*c^2*d^9*f + 120*A*C^2*a^7*b^3*c^3*d^8*f + 108*A*C^2*a^6*b^4*
c^8*d^3*f - 102*A^2*C*a^6*b^4*c^4*d^7*f - 96*A^2*C*a^8*b^2*c^4*d^7*f + 72*A
^2*C*a^7*b^3*c^3*d^8*f + 72*A*C^2*a^5*b^5*c^9*d^2*f + 48*A^2*C*a^7*b^3*c^5*
d^6*f - 48*A^2*C*a^3*b^7*c^9*d^2*f - 48*A*C^2*a^4*b^6*c^2*d^9*f - 24*A^2*C*
a^5*b^5*c^3*d^8*f - 12*A*C^2*a^8*b^2*c^4*d^7*f + 2736*A^2*B*a^3*b^7*c^6*d^5
*f + 2464*A^2*B*a^4*b^6*c^3*d^8*f - 2298*A*B^2*a^4*b^6*c^4*d^7*f - 2252*A^2
*B*a^2*b^8*c^5*d^6*f - 1692*A^2*B*a^5*b^5*c^4*d^7*f - 1592*A*B^2*a^2*b^8*c^
4*d^7*f - 1338*A*B^2*a^4*b^6*c^6*d^5*f + 1320*A*B^2*a^3*b^7*c^5*d^6*f + 121
2*A*B^2*a^5*b^5*c^5*d^6*f - 1056*A*B^2*a^5*b^5*c^3*d^8*f + 1024*A^2*B*a^3*b
^7*c^4*d^7*f - 1022*A^2*B*a^4*b^6*c^7*d^4*f - 880*A^2*B*a^5*b^5*c^2*d^9*f -
 846*A^2*B*a^4*b^6*c^5*d^6*f - 840*A*B^2*a^3*b^7*c^7*d^4*f + 760*A*B^2*a^6*
b^4*c^2*d^9*f - 704*A^2*B*a^3*b^7*c^2*d^9*f + 688*A*B^2*a^3*b^7*c^3*d^8*f +
 660*A^2*B*a^6*b^4*c^3*d^8*f - 612*A^2*B*a^2*b^8*c^7*d^4*f + 462*A*B^2*a^6*
b^4*c^4*d^7*f + 459*A*B^2*a^2*b^8*c^8*d^3*f - 412*A*B^2*a^2*b^8*c^2*d^9*f -
 408*A*B^2*a^7*b^3*c^3*d^8*f + 388*A^2*B*a^5*b^5*c^6*d^5*f + 296*A^2*B*a^2*
b^8*c^3*d^8*f + 288*A*B^2*a^2*b^8*c^6*d^5*f + 284*A*B^2*a^5*b^5*c^7*d^4*f +
 236*A*B^2*a^4*b^6*c^8*d^3*f - 226*A*B^2*a^6*b^4*c^6*d^5*f + 212*A*B^2*a^4*
b^6*c^2*d^9*f + 202*A^2*B*a^6*b^4*c^5*d^6*f - 152*A^2*B*a^7*b^3*c^4*d^7*f +
 88*A^2*B*a^3*b^7*c^8*d^3*f + 79*A^2*B*a^2*b^8*c^9*d^2*f - 70*A^2*B*a^6*b^4
*c^7*d^4*f + 68*A*B^2*a^8*b^2*c^4*d^7*f + 64*A^2*B*a^7*b^3*c^2*d^9*f - 64*A
*B^2*a^3*b^7*c^9*d^2*f + 56*A^2*B*a^7*b^3*c^6*d^5*f + 56*A^2*B*a^5*b^5*c^8*
d^3*f + 37*A^2*B*a^8*b^2*c^3*d^8*f - 28*A^2*B*a^8*b^2*c^5*d^6*f - 28*A^2*B*

a^4*b^6*c^9*d^2*f + 17*A*B^2*a^8*b^2*c^2*d^9*f - 16*A*B^2*a^7*b^3*c^5*d^6*f + 24*A*B*C*b^10*c*d^10*f - 6*A*B*C*a^10*c*d^10*f + 48*A*B*C*a*b^9*d^11*f + 4*A*B*C*a^9*b*d^11*f + 432*B^2*C*a*b^9*c^7*d^4*f - 376*B*C^2*a^6*b^4*c*d^10*f - 354*B*C^2*a*b^9*c^8*d^3*f + 352*B^2*C*a^5*b^5*c*d^10*f + 320*B^2*C*a*b^9*c^5*d^6*f + 256*B^2*C*a^3*b^7*c*d^10*f - 232*B^2*C*a^7*b^3*c*d^10*f - 210*B^2*C*a*b^9*c^9*d^2*f - 152*B*C^2*a^4*b^6*c*d^10*f + 85*B*C^2*a^8*b^2*c*d^10*f + 72*B^2*C*a*b^9*c^3*d^8*f - 48*B*C^2*a*b^9*c^6*d^5*f - 40*B*C^2*a^3*b^7*c^10*d*f + 40*B*C^2*a^2*b^8*c*d^10*f + 37*B^2*C*a^2*b^8*c^10*d*f + 22*B^2*C*a^9*b*c^3*d^8*f - 18*B*C^2*a^9*b*c^2*d^9*f + 16*B*C^2*a*b^9*c^2*d^9*f - 12*B^2*C*a^4*b^6*c^10*d*f + 8*B*C^2*a^9*b*c^4*d^7*f + 8*B*C^2*a*b^9*c^4*d^7*f - 984*A^2*C*a*b^9*c^7*d^4*f + 672*A^2*C*a*b^9*c^3*d^8*f + 552*A*C^2*a*b^9*c^7*d^4*f - 504*A^2*C*a^5*b^5*c*d^10*f - 408*A^2*C*a*b^9*c^5*d^6*f + 408*A*C^2*a*b^9*c^5*d^6*f + 336*A*C^2*a^5*b^5*c*d^10*f - 216*A*C^2*a^7*b^3*c*d^10*f + 192*A*C^2*a^3*b^7*c*d^10*f - 162*A*C^2*a*b^9*c^9*d^2*f + 120*A^2*C*a^7*b^3*c*d^10*f + 96*A^2*C*a^3*b^7*c*d^10*f + 90*A^2*C*a*b^9*c^9*d^2*f + 66*A^2*C*a^9*b*c^3*d^8*f - 66*A*C^2*a^9*b*c^3*d^8*f + 57*A*C^2*a^2*b^8*c^10*d*f - 48*A*C^2*a*b^9*c^3*d^8*f - 9*A^2*C*a^2*b^8*c^10*d*f + 1736*A^2*B*a*b^9*c^4*d^7*f + 1248*A^2*B*a*b^9*c^6*d^5*f - 1008*A*B^2*a*b^9*c^7*d^4*f + 772*A^2*B*a^4*b^6*c*d^10*f - 688*A*B^2*a^5*b^5*c*d^10*f - 608*A*B^2*a*b^9*c^5*d^6*f + 436*A^2*B*a^2*b^8*c*d^10*f - 426*A^2*B*a*b^9*c^8*d^3*f + 312*A*B^2*a*b^9*c^3*d^8*f + 304*A^2*B*a*b^9*c^2*d^9*f - 244*A^2*B*a^6*b^4*c*d^10*f - 160*A*B^2*a^3*b^7*c*d^10*f + 114*A*B^2*a*b^9*c^9*d^2*f + 88*A*B^2*a^7*b^3*c*d^10*f - 22*A*B^2*a^9*b*c^3*d^8*f - 18*A^2*B*a^9*b*c^2*d^9*f + 13*A^2*B*a^8*b^2*c*d^10*f - 13*A*B^2*a^2*b^8*c^10*d*f + 8*A^2*B*a^9*b*c^4*d^7*f + 8*A^2*B*a^3*b^7*c^10*d*f + 111*B^2*C*b^10*c^8*d^3*f - 39*B*C^2*b^10*c^9*d^2*f + 24*B*C^2*b^10*c^7*d^4*f - 4*B^2*C*b^10*c^2*d^9*f - 4*B*C^2*b^10*c^5*d^6*f + 432*A^2*C*b^10*c^6*d^5*f + 192*A^2*C*b^10*c^4*d^7*f - 111*A^2*C*b^10*c^8*d^3*f + 111*A*C^2*b^10*c^8*d^3*f - 72*A*C^2*b^10*c^6*d^5*f + 12*A*C^2*b^10*c^4*d^7*f - 3*B^2*C*a^10*c^2*d^9*f - B*C^2*a^10*c^3*d^8*f + 456*A^2*B*b^10*c^7*d^4*f - 288*A^2*B*b^10*c^3*d^8*f + 252*A*B^2*b^10*c^6*d^5*f + 192*A*B^2*b^10*c^4*d^7*f - 183*A*B^2*b^10*c^8*d^3*f - 148*A^2*B*b^10*c^5*d^6*f + 112*B^2*C*a^6*b^4*d^11*f + 76*A*B^2*b^10*c^2*d^9*f - 64*B*C^2*a^7*b^3*d^11*f + 16*B^2*C*a^4*b^6*d^11*f - 16*B^2*C*a^2*b^8*d^11*f + 16*B*C^2*a^5*b^5*d^11*f + 16*B*C^2*a^3*b^7*d^11*f - 9*A^2*C*a^10*c^2*d^9*f + 9*A*C^2*a^10*c^2*d^9*f - 3*A^2*B*b^10*c^9*d^2*f - B^2*C*a^8*b^2*d^11*f + 96*A^2*C*a^4*b^6*d^11*f - 84*A^2*C*a^6*b^4*d^11*f + 72*A*C^2*a^6*b^4*d^11*f - 24*A*C^2*a^4*b^6*d^11*f - 24*A*C^2*a^2*b^8*d^11*f - 21*A*C^2*a^8*b^2*d^11*f + 12*A^2*C*a^2*b^8*d^11*f + 9*A^2*C*a^8*b^2*d^11*f + 3*A*B^2*a^10*c^2*d^9*f - A^2*B*a^10*c^3*d^8*f - B*C^2*a^2*b^8*c^11*f + 176*A*B^2*a^4*b^6*d^11*f + 136*A^2*B*a^5*b^5*d^11*f - 128*A^2*B*a^3*b^7*d^11*f + 112*A*B^2*a^2*b^8*d^11*f - 64*A*B^2*a^6*b^4*d^11*f - 16*A^2*B*a^7*b^3*d^11*f - A^2*B*a^2*b^8*c^11*f - 2*C^3*a^9*b*c*d^10*f - 2*B^3*a*b^9*c^10*d*f - 264*A^3*a*b^9*c*d^10*f + 2*A^3*a^9*b*c*d^10*f - 9*B^2*C*b^10*c^10*d*f + 9*A^2*C*b^10*c^10*d*f - 9*A*C^2*b^10*c^10*d*f + 3*B*C^2*a^10*c*d^10*f - 132*A^2*B*b^10*c*d^10*f - 3*A*B^2*b^10*c^10*d*f - 2*B*C^2*a^9*b*d^11*f + 3*A^2*B*a^10*c*d^10*f - 2*B^2*C*a*b^9*c^11*f -

120*A^2*B*a*b^9*d^11*f - 6*A^2*C*a*b^9*c^11*f + 6*A*C^2*a*b^9*c^11*f - 2*A
^2*B*a^9*b*d^11*f + 2*A*B^2*a*b^9*c^11*f + 520*C^3*a^3*b^7*c^5*d^6*f + 460*
C^3*a^5*b^5*c^5*d^6*f - 418*C^3*a^4*b^6*c^6*d^5*f + 406*C^3*a^6*b^4*c^4*d^7
*f + 268*C^3*a^5*b^5*c^7*d^4*f - 266*C^3*a^6*b^4*c^6*d^5*f + 233*C^3*a^2*b^
8*c^8*d^3*f - 176*C^3*a^7*b^3*c^5*d^6*f + 164*C^3*a^6*b^4*c^2*d^9*f + 140*C
^3*a^2*b^8*c^6*d^5*f + 136*C^3*a^4*b^6*c^2*d^9*f - 128*C^3*a^3*b^7*c^9*d^2*
f + 128*C^3*a^3*b^7*c^3*d^8*f - 108*C^3*a^6*b^4*c^8*d^3*f - 104*C^3*a^7*b^3
*c^3*d^8*f - 104*C^3*a^5*b^5*c^3*d^8*f + 100*C^3*a^4*b^6*c^8*d^3*f - 89*C^3
*a^8*b^2*c^2*d^9*f - 72*C^3*a^5*b^5*c^9*d^2*f + 40*C^3*a^8*b^2*c^4*d^7*f -
40*C^3*a^3*b^7*c^7*d^4*f - 28*C^3*a^2*b^8*c^4*d^7*f - 16*C^3*a^2*b^8*c^2*d^
9*f - 2*C^3*a^4*b^6*c^4*d^7*f + 828*B^3*a^5*b^5*c^4*d^7*f + 408*B^3*a^2*b^8
*c^5*d^6*f + 390*B^3*a^4*b^6*c^7*d^4*f - 372*B^3*a^4*b^6*c^3*d^8*f - 336*B^
3*a^3*b^7*c^6*d^5*f - 314*B^3*a^6*b^4*c^5*d^6*f + 288*B^3*a^3*b^7*c^4*d^7*f
 + 216*B^3*a^2*b^8*c^7*d^4*f - 176*B^3*a^7*b^3*c^2*d^9*f + 128*B^3*a^3*b^7*
c^2*d^9*f + 108*B^3*a^5*b^5*c^6*d^5*f + 88*B^3*a^7*b^3*c^4*d^7*f + 72*B^3*a
^5*b^5*c^2*d^9*f - 68*B^3*a^2*b^8*c^3*d^8*f - 65*B^3*a^2*b^8*c^9*d^2*f - 56
*B^3*a^5*b^5*c^8*d^3*f + 40*B^3*a^7*b^3*c^6*d^5*f + 37*B^3*a^8*b^2*c^3*d^8*
f + 30*B^3*a^4*b^6*c^5*d^6*f - 28*B^3*a^8*b^2*c^5*d^6*f + 24*B^3*a^3*b^7*c^
8*d^3*f - 4*B^3*a^4*b^6*c^9*d^2*f - 2*B^3*a^6*b^4*c^7*d^4*f + 1586*A^3*a^4*
b^6*c^4*d^7*f - 1376*A^3*a^3*b^7*c^3*d^8*f - 1096*A^3*a^3*b^7*c^5*d^6*f + 8
44*A^3*a^2*b^8*c^4*d^7*f - 748*A^3*a^5*b^5*c^5*d^6*f + 490*A^3*a^4*b^6*c^6*
d^5*f + 376*A^3*a^2*b^8*c^2*d^9*f + 362*A^3*a^6*b^4*c^4*d^7*f - 356*A^3*a^2
*b^8*c^6*d^5*f - 328*A^3*a^5*b^5*c^3*d^8*f + 328*A^3*a^3*b^7*c^7*d^4*f + 22
4*A^3*a^4*b^6*c^2*d^9*f - 197*A^3*a^2*b^8*c^8*d^3*f - 112*A^3*a^7*b^3*c^5*d
^6*f + 98*A^3*a^6*b^4*c^6*d^5*f - 92*A^3*a^6*b^4*c^2*d^9*f - 88*A^3*a^7*b^3
*c^3*d^8*f + 68*A^3*a^8*b^2*c^4*d^7*f + 32*A^3*a^3*b^7*c^9*d^2*f - 28*A^3*a
^5*b^5*c^7*d^4*f - 28*A^3*a^4*b^6*c^8*d^3*f + 17*A^3*a^8*b^2*c^2*d^9*f + 10
4*C^3*a^7*b^3*c*d^10*f + 54*C^3*a*b^9*c^9*d^2*f - 40*C^3*a*b^9*c^7*d^4*f -
35*C^3*a^2*b^8*c^10*d*f + 22*C^3*a^9*b*c^3*d^8*f + 16*C^3*a^5*b^5*c*d^10*f
 - 16*C^3*a^3*b^7*c*d^10*f + 8*C^3*a*b^9*c^5*d^6*f - 2*A*B*C*b^10*c^11*f + 1
98*B^3*a*b^9*c^8*d^3*f + 192*B^3*a^6*b^4*c*d^10*f - 128*B^3*a*b^9*c^4*d^7*f
 - 80*B^3*a^2*b^8*c*d^10*f - 56*B^3*a*b^9*c^2*d^9*f - 24*B^3*a*b^9*c^6*d^5*
f - 18*B^3*a^9*b*c^2*d^9*f - 16*B^3*a^4*b^6*c*d^10*f + 13*B^3*a^8*b^2*c*d^1
0*f + 8*B^3*a^9*b*c^4*d^7*f + 8*B^3*a^3*b^7*c^10*d*f - 624*A^3*a*b^9*c^3*d^
8*f + 472*A^3*a*b^9*c^7*d^4*f - 272*A^3*a^3*b^7*c*d^10*f + 152*A^3*a^5*b^5*
c*d^10*f - 22*A^3*a^9*b*c^3*d^8*f + 18*A^3*a*b^9*c^9*d^2*f - 13*A^3*a^2*b^8
*c^10*d*f - 8*A^3*a^7*b^3*c*d^10*f - 8*A^3*a*b^9*c^5*d^6*f + A*B^2*a^8*b^2*
d^11*f - C^3*b^10*c^8*d^3*f - 60*B^3*b^10*c^7*d^4*f - 32*B^3*b^10*c^5*d^6*f
 + 21*B^3*b^10*c^9*d^2*f - 12*B^3*b^10*c^3*d^8*f - 3*C^3*a^10*c^2*d^9*f - 3
60*A^3*b^10*c^6*d^5*f - 204*A^3*b^10*c^4*d^7*f + 11*C^3*a^8*b^2*d^11*f - 8*
C^3*a^6*b^4*d^11*f - 4*C^3*a^4*b^6*d^11*f - B^3*a^10*c^3*d^8*f - 64*B^3*a^5
*b^5*d^11*f - 32*B^3*a^3*b^7*d^11*f + 3*A^3*a^10*c^2*d^9*f - 68*A^3*a^4*b^6
*d^11*f + 20*A^3*a^6*b^4*d^11*f + 12*A^3*a^2*b^8*d^11*f - B^3*a^2*b^8*c^11*
f + 3*C^3*b^10*c^10*d*f + 3*B^3*a^10*c*d^10*f - 3*A^3*b^10*c^10*d*f - 2*C^3
*a*b^9*c^11*f - 2*B^3*a^9*b*d^11*f + 2*A^3*a*b^9*c^11*f - 36*A^2*C*b^10*d^1

1*f + 3*A^2*C*a^10*d^11*f - 3*A*C^2*a^10*d^11*f - A*B^2*a^10*d^11*f + 36*A^
3*b^10*d^11*f - A^3*a^10*d^11*f + A^3*b^10*c^8*d^3*f + A^3*a^8*b^2*d^11*f +
B^2*C*a^10*d^11*f + B*C^2*b^10*c^11*f + A^2*B*b^10*c^11*f + C^3*a^10*d^11*
f + B^3*b^10*c^11*f - 6*A*B^2*C*a*b^7*c^7*d + 4*A*B^2*C*a*b^7*c*d^7 + 168*A
^2*B*C*a^3*b^5*c^2*d^6 + 144*A*B*C^2*a^4*b^4*c^3*d^5 - 129*A^2*B*C*a^4*b^4*
c^3*d^5 - 96*A*B*C^2*a^3*b^5*c^2*d^6 + 84*A*B*C^2*a^2*b^6*c^3*d^5 + 72*A^2*
B*C*a^3*b^5*c^4*d^4 - 72*A^2*B*C*a^2*b^6*c^3*d^5 + 64*A*B^2*C*a^4*b^4*c^4*d
^4 - 60*A*B*C^2*a^3*b^5*c^4*d^4 + 57*A^2*B*C*a^2*b^6*c^5*d^3 - 56*A*B^2*C*a
^3*b^5*c^5*d^3 - 39*A*B^2*C*a^4*b^4*c^2*d^6 - 38*A*B^2*C*a^5*b^3*c^3*d^5 +
36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^4*b^4*c^5*d^3 - 30*A*B*C^2*a^2*b^
6*c^5*d^3 + 27*A*B^2*C*a^2*b^6*c^6*d^2 - 24*A*B^2*C*a^2*b^6*c^2*d^6 - 24*A*
B*C^2*a^5*b^3*c^4*d^4 + 24*A*B*C^2*a^3*b^5*c^6*d^2 + 18*A^2*B*C*a^5*b^3*c^2
*d^6 - 18*A^2*B*C*a^4*b^4*c^5*d^3 - 15*A*B^2*C*a^2*b^6*c^4*d^4 + 12*A^2*B*C
*a^5*b^3*c^4*d^4 - 12*A^2*B*C*a^3*b^5*c^6*d^2 + 9*A*B^2*C*a^6*b^2*c^2*d^6 +
6*A*B*C^2*a^6*b^2*c^3*d^5 - 3*A^2*B*C*a^6*b^2*c^3*d^5 + 60*A^2*B*C*a*b^7*c
^2*d^6 - 51*A^2*B*C*a^4*b^4*c*d^7 + 48*A*B*C^2*a*b^7*c^6*d^2 - 42*A^2*B*C*a
^2*b^6*c*d^7 - 42*A^2*B*C*a*b^7*c^6*d^2 + 36*A*B*C^2*a^4*b^4*c*d^7 + 36*A*B
*C^2*a^2*b^6*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d^4 - 30*A^2*B*C*a*b^7*c^4*d^4 +
24*A*B^2*C*a*b^7*c^3*d^5 - 24*A*B*C^2*a*b^7*c^2*d^6 + 18*A*B^2*C*a^5*b^3*c*
d^7 - 18*A*B*C^2*a^6*b^2*c*d^7 + 12*A*B^2*C*a^3*b^5*c*d^7 + 9*A^2*B*C*a^6*b
^2*c*d^7 + 6*A*B^2*C*a*b^7*c^5*d^3 - 6*A*B*C^2*a^2*b^6*c^7*d + 3*A^2*B*C*a^
2*b^6*c^7*d - 18*B^3*C*a*b^7*c^6*d^2 - 18*B*C^3*a*b^7*c^6*d^2 - 14*B^3*C*a*
b^7*c^4*d^4 - 14*B*C^3*a*b^7*c^4*d^4 - 10*B^3*C*a^2*b^6*c*d^7 - 10*B*C^3*a^
2*b^6*c*d^7 + 9*B^3*C*a^6*b^2*c*d^7 + 9*B*C^3*a^6*b^2*c*d^7 - 7*B^3*C*a^4*b
^4*c*d^7 - 7*B*C^3*a^4*b^4*c*d^7 + 6*B^2*C^2*a*b^7*c^7*d - 4*B^3*C*a*b^7*c^
2*d^6 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a*b^7*c^2*d^6 + 3*B^3*C*a^2*b^6*c^7
*d + 3*B*C^3*a^2*b^6*c^7*d + 144*A^3*C*a*b^7*c^3*d^5 + 62*A^3*C*a*b^7*c^5*d
^3 + 48*A*C^3*a*b^7*c^3*d^5 - 36*A^2*C^2*a*b^7*c*d^7 + 26*A*C^3*a*b^7*c^5*d
^3 + 20*A^3*C*a^3*b^5*c*d^7 + 18*A^2*C^2*a*b^7*c^7*d - 18*A*C^3*a^5*b^3*c*d
^7 - 6*A^3*C*a^5*b^3*c*d^7 - 4*A*C^3*a^3*b^5*c*d^7 - 32*A^3*B*a*b^7*c^2*d^6
- 32*A*B^3*a*b^7*c^2*d^6 + 22*A^3*B*a^4*b^4*c*d^7 + 22*A*B^3*a^4*b^4*c*d^7
+ 16*A^3*B*a^2*b^6*c*d^7 + 16*A*B^3*a^2*b^6*c*d^7 + 12*A^3*B*a*b^7*c^6*d^2
+ 12*A*B^3*a*b^7*c^6*d^2 + 8*A^3*B*a*b^7*c^4*d^4 - 8*A^2*B^2*a*b^7*c*d^7 +
8*A*B^3*a*b^7*c^4*d^4 + 57*A^2*B*C*b^8*c^5*d^3 + 36*A^2*B*C*b^8*c^3*d^5 -
30*A*B*C^2*b^8*c^5*d^3 - 18*A*B*C^2*b^8*c^3*d^5 - 9*A*B^2*C*b^8*c^4*d^4 - 3
*A*B^2*C*b^8*c^6*d^2 - 2*A*B^2*C*b^8*c^2*d^6 + 36*A^2*B*C*a^3*b^5*d^8 + 24*
A*B*C^2*a^5*b^3*d^8 - 18*A^2*B*C*a^5*b^3*d^8 - 12*A*B*C^2*a^3*b^5*d^8 - 3*A
*B^2*C*a^6*b^2*d^8 - 3*A*B^2*C*a^4*b^4*d^8 - 2*A*B^2*C*a^2*b^6*d^8 + 34*B^2
*C^2*a^5*b^3*c^3*d^5 + 28*B^2*C^2*a^3*b^5*c^5*d^3 + 24*B^2*C^2*a^4*b^4*c^2*
d^6 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*d^5 + 12*B^2*C^2*
a^2*b^6*c^2*d^6 - 9*B^2*C^2*a^6*b^2*c^2*d^6 + 9*B^2*C^2*a^4*b^4*c^6*d^2 + 9
*B^2*C^2*a^2*b^6*c^4*d^4 - 3*B^2*C^2*a^2*b^6*c^6*d^2 + 159*A^2*C^2*a^2*b^6*
c^4*d^4 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^5*b^3*c^3*d^5 + 78*A^2
*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2*C^2*a^6*b^2*c^2*
d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^4*b^4*c^2*d^6 + 9*A^2*C^2*a

^4*b^4*c^6*d^2 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a^2*b^6*c^4*d^4 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^4*b^4*c^2*d^6 + 28*A^2*B^2*a^3*b^5*c^5*d^3 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^2*b^6*c^6*d^2 + 4*A^2*B^2*a^5*b^3*c^3*d^5 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a*b^7*c^7*d + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a*b^7*c^7*d + 12*A^2*B*C*b^8*c*d^7 + 6*A*B*C^2*b^8*c^7*d - 6*A*B*C^2*b^8*c*d^7 - 3*A^2*B*C*b^8*c^7*d + 24*A^2*B*C*a*b^7*d^8 - 12*A*B*C^2*a*b^7*d^8 - 53*B^3*C*a^4*b^4*c^3*d^5 - 53*B*C^3*a^4*b^4*c^3*d^5 - 32*B^3*C*a^2*b^6*c^3*d^5 - 32*B*C^3*a^2*b^6*c^3*d^5 - 18*B^3*C*a^4*b^4*c^5*d^3 - 18*B*C^3*a^4*b^4*c^5*d^3 + 16*B^3*C*a^3*b^5*c^4*d^4 + 16*B*C^3*a^3*b^5*c^4*d^4 + 12*B^3*C*a^5*b^3*c^4*d^4 - 12*B^3*C*a^3*b^5*c^6*d^2 + 12*B^2*C^2*a*b^7*c^3*d^5 + 12*B*C^3*a^5*b^3*c^4*d^4 - 12*B*C^3*a^3*b^5*c^6*d^2 + 8*B^3*C*a^3*b^5*c^2*d^6 + 8*B*C^3*a^3*b^5*c^2*d^6 - 6*B^3*C*a^5*b^3*c^2*d^6 - 6*B^2*C^2*a^5*b^3*c*d^7 + 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*C^3*a^5*b^3*c^2*d^6 - 3*B^3*C*a^6*b^2*c^3*d^5 - 3*B*C^3*a^6*b^2*c^3*d^5 - 175*A^3*C*a^2*b^6*c^4*d^4 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a*b^7*c^3*d^5 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^5*b^3*c^3*d^5 - 73*A*C^3*a^2*b^6*c^4*d^4 - 66*A^2*C^2*a*b^7*c^5*d^3 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C^3*a^4*b^4*c^4*d^4 - 30*A^3*C*a^5*b^3*c^3*d^5 + 30*A^3*C*a^4*b^4*c^4*d^4 + 27*A*C^3*a^6*b^2*c^2*d^6 + 21*A*C^3*a^4*b^4*c^2*d^6 + 18*A^2*C^2*a^5*b^3*c*d^7 - 18*A*C^3*a^4*b^4*c^6*d^2 - 16*A*C^3*a^2*b^6*c^2*d^6 - 15*A^3*C*a^4*b^4*c^2*d^6 + 15*A^3*C*a^2*b^6*c^6*d^2 - 12*A^2*C^2*a^3*b^5*c*d^7 + 9*A^3*C*a^6*b^2*c^2*d^6 + 9*A*C^3*a^2*b^6*c^6*d^2 - 80*A^3*B*a^3*b^5*c^2*d^6 - 80*A*B^3*a^3*b^5*c^2*d^6 + 38*A^3*B*a^4*b^4*c^3*d^5 + 38*A*B^3*a^4*b^4*c^3*d^5 - 36*A^2*B^2*a*b^7*c^3*d^5 - 28*A^3*B*a^3*b^5*c^4*d^4 - 28*A^3*B*a^2*b^6*c^5*d^3 - 28*A*B^3*a^3*b^5*c^4*d^4 - 28*A*B^3*a^2*b^6*c^5*d^3 + 20*A^3*B*a^2*b^6*c^3*d^5 + 20*A*B^3*a^2*b^6*c^3*d^5 - 12*A^3*B*a^5*b^3*c^2*d^6 - 12*A^2*B^2*a^5*b^3*c*d^7 - 12*A^2*B^2*a^3*b^5*c*d^7 - 12*A^2*B^2*a*b^7*c^5*d^3 - 12*A*B^3*a^5*b^3*c^2*d^6 + 6*B^2*C^2*b^8*c^6*d^2 + 3*B^2*C^2*b^8*c^4*d^4 + 36*A^2*C^2*b^8*c^4*d^4 + 27*A^2*C^2*b^8*c^2*d^6 - 18*A^2*C^2*b^8*c^6*d^2 + 33*A^2*B^2*b^8*c^4*d^4 + 28*A^2*B^2*b^8*c^2*d^6 + 9*B^2*C^2*a^4*b^4*d^8 + 6*A^2*B^2*b^8*c^6*d^2 + 4*B^2*C^2*a^2*b^6*d^8 + 3*B^2*C^2*a^6*b^2*d^8 - 30*A^2*C^2*a^4*b^4*d^8 + 9*A^2*C^2*a^6*b^2*d^8 + 16*A^2*B^2*a^2*b^6*d^8 + 3*A^2*B^2*a^4*b^4*d^8 + 6*C^4*a^5*b^3*c*d^7 + 4*C^4*a^3*b^5*c*d^7 - 2*C^4*a*b^7*c^5*d^3 - 12*B^4*a^5*b^3*c*d^7 + 12*B^4*a*b^7*c^3*d^5 + 8*B^4*a*b^7*c^5*d^3 - 4*B^4*a^3*b^5*c*d^7 - 48*A^4*a*b^7*c^3*d^5 - 20*A^4*a*b^7*c^5*d^3 - 8*A^4*a^3*b^5*c*d^7 - 63*A^3*C*b^8*c^4*d^4 - 54*A^3*C*b^8*c^2*d^6 + 9*A^3*C*b^8*c^6*d^2 + 9*A*C^3*b^8*c^6*d^2 - 3*A*C^3*b^8*c^4*d^4 - 28*A^3*B*b^8*c^5*d^3 - 28*A*B^3*b^8*c^5*d^3 - 18*A^3*B*b^8*c^3*d^5 - 18*A*B^3*b^8*c^3*d^5 - 10*B^3*C*a^5*b^3*d^8 - 10*B*C^3*a^5*b^3*d^8 - 4*B^3*C*a^3*b^5*d^8 - 4*B*C^3*a^3*b^5*d^8 + 23*A^3*C*a^4*b^4*d^8 - 18*A^3*C*a^2*b^6*d^8 + 11*A*C^3*a^4*b^4*d^8 - 9*A*C^3*a^6*b^2*d^8 + 6*A*C^3*a^2*b^6*d^8 - 3*A^3*C*a^6*b^2*d^8 - 20*A^3*B*a^3*b^5*d^8 - 20*A*B^3*a^3*b^5*d^8 + 4*A^3*B*a^5*b^3*d^8 + 4*A*B^3*a^5*b^3*d^8 + B^3*C*a^2*b^6*c^5*d^3 + B*C^3*a^2*b^6*c^5*d^3 + 6*C^4*a*b^7*c^7*d + 4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 3*B^3*C*b^8*c^7*d - 3*B*C^3*b^8*c^7*d - 6*A^3*B*b^8*c*d^7 - 6*A*B^3*b^8*c*d^7 - 12*A^3*B*a*b^7*d^8 - 12*A*B^3*a*b^7*d^8 + 3

0*C^4*a^5*b^3*c^3*d^5 + 19*C^4*a^2*b^6*c^4*d^4 - 9*C^4*a^6*b^2*c^2*d^6 + 9*
C^4*a^4*b^4*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5 + 4*C^4*a^2*b^6*c^2*d^6 - 3*C^4
*a^4*b^4*c^4*d^4 - 3*C^4*a^4*b^4*c^2*d^6 + 3*C^4*a^2*b^6*c^6*d^2 + 28*B^4*a
^3*b^5*c^5*d^3 + 27*B^4*a^4*b^4*c^2*d^6 - 17*B^4*a^4*b^4*c^4*d^4 - 10*B^4*a
^2*b^6*c^4*d^4 + 8*B^4*a^3*b^5*c^3*d^5 + 8*B^4*a^2*b^6*c^2*d^6 - 6*B^4*a^2*
b^6*c^6*d^2 + 4*B^4*a^5*b^3*c^3*d^5 + 70*A^4*a^2*b^6*c^4*d^4 + 58*A^4*a^2*b
^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^4*a^4*b^4*c^2*d^6 + B^2*C^2*b^8*
c^2*d^6 - 18*A^3*C*b^8*d^8 + B^3*C*b^8*c^5*d^3 + B*C^3*b^8*c^5*d^3 + 6*B^4*
b^8*c^6*d^2 + 3*B^4*b^8*c^4*d^4 + 30*A^4*b^8*c^4*d^4 + 27*A^4*b^8*c^2*d^6 +
 3*C^4*a^6*b^2*d^8 + 8*B^4*a^4*b^4*d^8 + 4*B^4*a^2*b^6*d^8 + 12*A^4*a^2*b^6
*d^8 - 5*A^4*a^4*b^4*d^8 + 9*A^2*C^2*b^8*d^8 + 9*A^2*B^2*b^8*d^8 + 9*A^4*b^
8*d^8 + B^4*b^8*c^2*d^6 + C^4*a^4*b^4*d^8, f, k)*((4*a^7*b^8*d^19 + 4*a^9*b
^6*d^19 - 4*a^11*b^4*d^19 - 4*a^13*b^2*d^19 + 4*b^15*c^7*d^12 + 12*b^15*c^9
*d^10 + 8*b^15*c^11*d^8 - 8*b^15*c^13*d^6 - 12*b^15*c^15*d^4 - 4*b^15*c^17*
d^2 - 20*a*b^14*c^6*d^13 - 44*a*b^14*c^8*d^11 + 32*a*b^14*c^10*d^9 + 168*a*
b^14*c^12*d^7 + 172*a*b^14*c^14*d^5 + 68*a*b^14*c^16*d^3 + 16*a^3*b^12*c^18
*d + 8*a^5*b^10*c^18*d - 20*a^6*b^9*c*d^18 - 4*a^8*b^7*c*d^18 + 60*a^10*b^5
*c*d^18 + 52*a^12*b^3*c*d^18 + 32*a^14*b*c^3*d^16 + 48*a^14*b*c^5*d^14 + 32
*a^14*b*c^7*d^12 + 8*a^14*b*c^9*d^10 + 36*a^2*b^13*c^5*d^14 + 32*a^2*b^13*c
^7*d^12 - 292*a^2*b^13*c^9*d^10 - 768*a^2*b^13*c^11*d^8 - 772*a^2*b^13*c^13
*d^6 - 352*a^2*b^13*c^15*d^4 - 60*a^2*b^13*c^17*d^2 - 20*a^3*b^12*c^4*d^15
+ 64*a^3*b^12*c^6*d^13 + 668*a^3*b^12*c^8*d^11 + 1648*a^3*b^12*c^10*d^9 + 1
892*a^3*b^12*c^12*d^7 + 1088*a^3*b^12*c^14*d^5 + 276*a^3*b^12*c^16*d^3 - 20
*a^4*b^11*c^3*d^16 - 104*a^4*b^11*c^5*d^14 - 640*a^4*b^11*c^7*d^12 - 2028*a
^4*b^11*c^9*d^10 - 3092*a^4*b^11*c^11*d^8 - 2368*a^4*b^11*c^13*d^6 - 856*a^
4*b^11*c^15*d^4 - 108*a^4*b^11*c^17*d^2 + 36*a^5*b^10*c^2*d^17 + 8*a^5*b^10
*c^4*d^15 + 112*a^5*b^10*c^6*d^13 + 1404*a^5*b^10*c^8*d^11 + 3404*a^5*b^10*
c^10*d^9 + 3552*a^5*b^10*c^12*d^7 + 1752*a^5*b^10*c^14*d^5 + 348*a^5*b^10*c
^16*d^3 + 64*a^6*b^9*c^3*d^16 + 392*a^6*b^9*c^5*d^14 + 32*a^6*b^9*c^7*d^12
- 1864*a^6*b^9*c^9*d^10 - 3296*a^6*b^9*c^11*d^8 - 2360*a^6*b^9*c^13*d^6 - 7
04*a^6*b^9*c^15*d^4 - 52*a^6*b^9*c^17*d^2 - 32*a^7*b^8*c^2*d^17 - 568*a^7*b
^8*c^4*d^15 - 1504*a^7*b^8*c^6*d^13 - 976*a^7*b^8*c^8*d^11 + 1120*a^7*b^8*c
^10*d^9 + 1912*a^7*b^8*c^12*d^7 + 928*a^7*b^8*c^14*d^5 + 140*a^7*b^8*c^16*d
^3 + 472*a^8*b^7*c^3*d^16 + 2016*a^8*b^7*c^5*d^14 + 3076*a^8*b^7*c^7*d^12 +
 1724*a^8*b^7*c^9*d^10 - 288*a^8*b^7*c^11*d^8 - 664*a^8*b^7*c^13*d^6 - 188*
a^8*b^7*c^15*d^4 - 240*a^9*b^6*c^2*d^17 - 1472*a^9*b^6*c^4*d^15 - 3316*a^9*
b^6*c^6*d^13 - 3484*a^9*b^6*c^8*d^11 - 1592*a^9*b^6*c^10*d^9 - 104*a^9*b^6*
c^12*d^7 + 92*a^9*b^6*c^14*d^5 + 704*a^10*b^5*c^3*d^16 + 2308*a^10*b^5*c^5*
d^14 + 3392*a^10*b^5*c^7*d^12 + 2468*a^10*b^5*c^9*d^10 + 832*a^10*b^5*c^11*
d^8 + 92*a^10*b^5*c^13*d^6 - 240*a^11*b^4*c^2*d^17 - 1108*a^11*b^4*c^4*d^15
 - 2112*a^11*b^4*c^6*d^13 - 2028*a^11*b^4*c^8*d^11 - 976*a^11*b^4*c^10*d^9
- 188*a^11*b^4*c^12*d^7 + 348*a^12*b^3*c^3*d^16 + 872*a^12*b^3*c^5*d^14 + 1
048*a^12*b^3*c^7*d^12 + 612*a^12*b^3*c^9*d^10 + 140*a^12*b^3*c^11*d^8 - 68*
a^13*b^2*c^2*d^17 - 232*a^13*b^2*c^4*d^15 - 328*a^13*b^2*c^6*d^13 - 212*a^1
3*b^2*c^8*d^11 - 52*a^13*b^2*c^10*d^9 + 8*a*b^14*c^18*d + 8*a^14*b*c*d^18)/

(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 + 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 + 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2 - 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^9*c^13*d - 6*a^9*b*c*d^13) + (tan(e + f*x)*(6*a^14*b*d^19 + 6*b^15*c^18*d + 8*a^6*b^9*d^19 + 22*a^8*b^7*d^19 + 26*a^10*b^5*d^19 + 18*a^12*b^3*d^19 + 8*b^15*c^6*d^13 + 38*b^15*c^8*d^11 + 78*b^15*c^10*d^9 + 92*b^15*c^12*d^7 + 68*b^15*c^14*d^5 + 30*b^15*c^16*d^3 - 48*a*b^14*c^5*d^14 - 224*a*b^14*c^7*d^12 - 448*a*b^14*c^9*d^10 - 512*a*b^14*c^11*d^8 - 368*a*b^14*c^13*d^6 - 160*a*b^14*c^15*d^4 - 32*a*b^14*c^17*d^2 + 10*a^2*b^13*c^18*d + 2*a^4*b^11*c^18*d - 48*a^5*b^10*c*d^18 - 2*a^6*b^9*c^18*d - 128*a^7*b^8*c*d^18 - 144*a^9*b^6*c*d^18 - 96*a^11*b^4*c*d^18 - 32*a^13*b^2*c*d^18 + 22*a^14*b*c^2*d^17 + 28*a^14*b*c^4*d^15 + 12*a^14*b*c^6*d^13 - 2*a^14*b*c^8*d^11 - 2*a^14*b*c^10*d^9 + 120*a^2*b^13*c^4*d^15 + 568*a^2*b^13*c^6*d^13 + 1138*a^2*b^13*c^8*d^11 + 1282*a^2*b^13*c^10*d^9 + 908*a^2*b^13*c^12*d^7 + 412*a^2*b^13*c^14*d^5 + 106*a^2*b^13*c^16*d^3 - 160*a^3*b^12*c^3*d^16 - 832*a^3*b^12*c^5*d^14 - 1776*a^3*b^12*c^7*d^12 - 2032*a^3*b^12*c^9*d^10 - 1408*a^3*b^12*c^11*d^8 - 672*a^3*b^12*c^13*d^6 - 240*a^3*b^12*c^15*d^4 - 48*a^3*b^12*c^17*d^2 + 120*a^4*b^11*c^2*d^17 + 820*a^4*b^11*c^4*d^15 + 2044*a^4*b^11*c^6*d^13 + 2434*a^4*b^11*c^8*d^11 + 1498*a^4*b^11*c^10*d^9 + 552*a^4*b^11*c^12*d^7 + 208*a^4*b^11*c^14*d^5 + 66*a^4*b^11*c^16*d^3 - 608*a^5*b^10*c^3*d^16 - 1904*a^5*b^10*c^5*d^14 - 2384*a^5*b^10*c^7*d^12 - 976*a^5*b^10*c^9*d^10 + 448*a^5*b^10*c^11*d^8 + 496*a^5*b^10*c^13*d^6 + 112*a^5*b^10*c^15*d^4 + 344*a^6*b^9*c^2*d^17 + 1428*a^6*b^9*c^4*d^15 + 1988*a^6*b^9*c^6*d^13 + 214*a^6*b^9*c^8*d^11 - 2058*a^6*b^9*c^10*d^9 - 2000*a^6*b^9*c^12*d^7 - 688*a^6*b^9*c^14*d^5 - 66*a^6*b^9*c^16*d^3 - 848*a^7*b^8*c^3*d^16 - 1520*a^7*b^8*c^5*d^14 + 80*a^7*b^8*c^7*d^12 + 3056*a^7*b^8*c^9*d^10 + 3536*a^7*b^8*c^11*d^8 + 1648*a^7*b^8*c^13*d^6 + 304*a^7*b^8*c^15*d^4 + 16*a^7*b^8*c^17*d^2 + 406*a^8*b^7*c^2*d^17 + 1072*a^8*b^7*c^4*d^15 + 200*a^8*b^7*c^6*d^13 - 2626*a^8*b^7*c^8*d^11 - 4042*a^8*b^7*c^10*d^9 - 2540*a^8*b^7*c^12*d^7 - 692*a^8*b^7*c^14*d^5 - 56*a^8*b^7*c^16*d^3 - 624*a^9*b^6*c^3*d^16 - 544*a^9*b^6*c^5*d^14 + 1296*a^9*b^6*c^7*d^12 + 3184*a^9*b^6*c^9*d^10 + 2672*a^9*b^6*c^11*d^8 + 960*a^9*b^6*c^13

*d^6 + 112*a^9*b^6*c^15*d^4 + 282*a^10*b^5*c^2*d^17 + 568*a^10*b^5*c^4*d^15
 - 168*a^10*b^5*c^6*d^13 - 1622*a^10*b^5*c^8*d^11 - 1862*a^10*b^5*c^10*d^9
- 860*a^10*b^5*c^12*d^7 - 140*a^10*b^5*c^14*d^5 - 336*a^11*b^4*c^3*d^16 - 2
72*a^11*b^4*c^5*d^14 + 352*a^11*b^4*c^7*d^12 + 768*a^11*b^4*c^9*d^10 + 496*
a^11*b^4*c^11*d^8 + 112*a^11*b^4*c^13*d^6 + 122*a^12*b^3*c^2*d^17 + 252*a^1
2*b^3*c^4*d^15 + 148*a^12*b^3*c^6*d^13 - 118*a^12*b^3*c^8*d^11 - 174*a^12*b
^3*c^10*d^9 - 56*a^12*b^3*c^12*d^7 - 112*a^13*b^2*c^3*d^16 - 128*a^13*b^2*c
^5*d^14 - 32*a^13*b^2*c^7*d^12 + 32*a^13*b^2*c^9*d^10 + 16*a^13*b^2*c^11*d^
8))/(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 +
 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10
*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^
2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d
^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c
*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c
^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72
*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*
c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 +
 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*
b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d
^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*
a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4
*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2
- 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*
b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^
10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^
9*c^13*d - 6*a^9*b*c*d^13)) - (C*b^13*c^15*d - A*b^13*c^15*d - B*a^12*b*d^1
6 + 12*A*a^3*b^10*d^16 + 20*A*a^5*b^8*d^16 - 4*A*a^9*b^4*d^16 + 4*A*a^11*b^
2*d^16 - 8*B*a^4*b^9*d^16 - 16*B*a^6*b^7*d^16 - B*a^8*b^5*d^16 + 6*B*a^10*b
^3*d^16 - 12*A*b^13*c^3*d^13 - 48*A*b^13*c^5*d^11 - 76*A*b^13*c^7*d^9 - 45*
A*b^13*c^9*d^7 + 5*A*b^13*c^11*d^5 + 9*A*b^13*c^13*d^3 + 4*C*a^5*b^8*d^16 +
 12*C*a^7*b^6*d^16 + 4*C*a^9*b^4*d^16 - 4*C*a^11*b^2*d^16 + 4*B*b^13*c^4*d^
12 + 16*B*b^13*c^6*d^10 + 35*B*b^13*c^8*d^8 + 33*B*b^13*c^10*d^6 + 5*B*b^13
*c^12*d^4 - 5*B*b^13*c^14*d^2 + 4*C*b^13*c^7*d^9 - 3*C*b^13*c^9*d^7 - 17*C*
b^13*c^11*d^5 - 9*C*b^13*c^13*d^3 + 36*A*a*b^12*c^2*d^14 + 176*A*a*b^12*c^4
*d^12 + 380*A*a*b^12*c^6*d^10 + 396*A*a*b^12*c^8*d^8 + 176*A*a*b^12*c^10*d^
6 + 20*A*a*b^12*c^12*d^4 - 36*A*a^2*b^11*c*d^15 - 2*A*a^2*b^11*c^15*d - 92*
A*a^4*b^9*c*d^15 - A*a^4*b^9*c^15*d - 56*A*a^6*b^7*c*d^15 + 3*A*a^8*b^5*c*d
^15 + 2*A*a^10*b^3*c*d^15 - 3*A*a^12*b*c^3*d^13 - 3*A*a^12*b*c^5*d^11 - A*a
^12*b*c^7*d^9 - 4*B*a*b^12*c^3*d^13 - 24*B*a*b^12*c^5*d^11 - 116*B*a*b^12*c
^7*d^9 - 196*B*a*b^12*c^9*d^7 - 120*B*a*b^12*c^11*d^5 - 20*B*a*b^12*c^13*d^
3 + 20*B*a^3*b^10*c*d^15 + 68*B*a^5*b^8*c*d^15 + 56*B*a^7*b^6*c*d^15 + 4*B*
a^9*b^4*c*d^15 - 4*B*a^11*b^2*c*d^15 - 3*B*a^12*b*c^2*d^14 - 3*B*a^12*b*c^4
*d^12 - B*a^12*b*c^6*d^10 - 8*C*a*b^12*c^4*d^12 - 56*C*a*b^12*c^6*d^10 - 60
*C*a*b^12*c^8*d^8 + 28*C*a*b^12*c^10*d^6 + 52*C*a*b^12*c^12*d^4 + 12*C*a*b^
12*c^14*d^2 + 2*C*a^2*b^11*c^15*d - 4*C*a^4*b^9*c*d^15 + C*a^4*b^9*c^15*d -

40*C*a^6*b^7*c*d^15 - 51*C*a^8*b^5*c*d^15 - 14*C*a^10*b^3*c*d^15 + 3*C*a^1
2*b*c^3*d^13 + 3*C*a^12*b*c^5*d^11 + C*a^12*b*c^7*d^9 - 260*A*a^2*b^11*c^3*
d^13 - 780*A*a^2*b^11*c^5*d^11 - 1144*A*a^2*b^11*c^7*d^9 - 798*A*a^2*b^11*c
^9*d^7 - 202*A*a^2*b^11*c^11*d^5 + 6*A*a^2*b^11*c^13*d^3 + 204*A*a^3*b^10*c
^2*d^14 + 876*A*a^3*b^10*c^4*d^12 + 1812*A*a^3*b^10*c^6*d^10 + 1872*A*a^3*b
^10*c^8*d^8 + 872*A*a^3*b^10*c^10*d^6 + 136*A*a^3*b^10*c^12*d^4 + 8*A*a^3*b
^10*c^14*d^2 - 608*A*a^4*b^9*c^3*d^13 - 1866*A*a^4*b^9*c^5*d^11 - 2802*A*a^
4*b^9*c^7*d^9 - 2007*A*a^4*b^9*c^9*d^7 - 585*A*a^4*b^9*c^11*d^5 - 31*A*a^4*
b^9*c^13*d^3 + 264*A*a^5*b^8*c^2*d^14 + 1180*A*a^5*b^8*c^4*d^12 + 2528*A*a^
5*b^8*c^6*d^10 + 2628*A*a^5*b^8*c^8*d^8 + 1200*A*a^5*b^8*c^10*d^6 + 172*A*a
^5*b^8*c^12*d^4 + 8*A*a^5*b^8*c^14*d^2 - 356*A*a^6*b^7*c^3*d^13 - 1320*A*a^
6*b^7*c^5*d^11 - 2188*A*a^6*b^7*c^7*d^9 - 1588*A*a^6*b^7*c^9*d^7 - 448*A*a^
6*b^7*c^11*d^5 - 28*A*a^6*b^7*c^13*d^3 + 24*A*a^7*b^6*c^2*d^14 + 368*A*a^7*
b^6*c^4*d^12 + 1112*A*a^7*b^6*c^6*d^10 + 1272*A*a^7*b^6*c^8*d^8 + 560*A*a^7
*b^6*c^10*d^6 + 56*A*a^7*b^6*c^12*d^4 + 33*A*a^8*b^5*c^3*d^13 - 165*A*a^8*b
^5*c^5*d^11 - 487*A*a^8*b^5*c^7*d^9 - 362*A*a^8*b^5*c^9*d^7 - 70*A*a^8*b^5*
c^11*d^5 - 68*A*a^9*b^4*c^2*d^14 - 108*A*a^9*b^4*c^4*d^12 + 28*A*a^9*b^4*c^
6*d^10 + 128*A*a^9*b^4*c^8*d^8 + 56*A*a^9*b^4*c^10*d^6 + 26*A*a^10*b^3*c^3*
d^13 + 18*A*a^10*b^3*c^5*d^11 - 34*A*a^10*b^3*c^7*d^9 - 28*A*a^10*b^3*c^9*d
^7 + 4*A*a^11*b^2*c^2*d^14 + 4*A*a^11*b^2*c^4*d^12 + 12*A*a^11*b^2*c^6*d^10
 + 8*A*a^11*b^2*c^8*d^8 - 12*B*a^2*b^11*c^2*d^14 - 44*B*a^2*b^11*c^4*d^12 +
 48*B*a^2*b^11*c^6*d^10 + 302*B*a^2*b^11*c^8*d^8 + 342*B*a^2*b^11*c^10*d^6
+ 118*B*a^2*b^11*c^12*d^4 - 2*B*a^2*b^11*c^14*d^2 + 132*B*a^3*b^10*c^3*d^13
 + 284*B*a^3*b^10*c^5*d^11 + 28*B*a^3*b^10*c^7*d^9 - 424*B*a^3*b^10*c^9*d^7
 - 336*B*a^3*b^10*c^11*d^5 - 56*B*a^3*b^10*c^13*d^3 - 132*B*a^4*b^9*c^2*d^1
4 - 558*B*a^4*b^9*c^4*d^12 - 694*B*a^4*b^9*c^6*d^10 - 27*B*a^4*b^9*c^8*d^8
+ 411*B*a^4*b^9*c^10*d^6 + 181*B*a^4*b^9*c^12*d^4 + 3*B*a^4*b^9*c^14*d^2 +
496*B*a^5*b^8*c^3*d^13 + 1196*B*a^5*b^8*c^5*d^11 + 1032*B*a^5*b^8*c^7*d^9 +
 84*B*a^5*b^8*c^9*d^7 - 216*B*a^5*b^8*c^11*d^5 - 36*B*a^5*b^8*c^13*d^3 - 24
4*B*a^6*b^7*c^2*d^14 - 1064*B*a^6*b^7*c^4*d^12 - 1596*B*a^6*b^7*c^6*d^10 -
828*B*a^6*b^7*c^8*d^8 + 68*B*a^6*b^7*c^12*d^4 + 488*B*a^7*b^6*c^3*d^13 + 12
24*B*a^7*b^6*c^5*d^11 + 1208*B*a^7*b^6*c^7*d^9 + 416*B*a^7*b^6*c^9*d^7 - 10
3*B*a^8*b^5*c^2*d^14 - 581*B*a^8*b^5*c^4*d^12 - 959*B*a^8*b^5*c^6*d^10 - 58
2*B*a^8*b^5*c^8*d^8 - 102*B*a^8*b^5*c^10*d^6 + 132*B*a^9*b^4*c^3*d^13 + 356
*B*a^9*b^4*c^5*d^11 + 332*B*a^9*b^4*c^7*d^9 + 104*B*a^9*b^4*c^9*d^7 + 18*B*
a^10*b^3*c^2*d^14 - 30*B*a^10*b^3*c^4*d^12 - 90*B*a^10*b^3*c^6*d^10 - 48*B*
a^10*b^3*c^8*d^8 + 4*B*a^11*b^2*c^3*d^13 + 20*B*a^11*b^2*c^5*d^11 + 12*B*a^
11*b^2*c^7*d^9 + 20*C*a^2*b^11*c^3*d^13 + 156*C*a^2*b^11*c^5*d^11 + 328*C*a
^2*b^11*c^7*d^9 + 234*C*a^2*b^11*c^9*d^7 + 10*C*a^2*b^11*c^11*d^5 - 30*C*a^
2*b^11*c^13*d^3 - 12*C*a^3*b^10*c^2*d^14 - 168*C*a^3*b^10*c^4*d^12 - 636*C*
a^3*b^10*c^6*d^10 - 828*C*a^3*b^10*c^8*d^8 - 344*C*a^3*b^10*c^10*d^6 + 20*C
*a^3*b^10*c^12*d^4 + 16*C*a^3*b^10*c^14*d^2 + 56*C*a^4*b^9*c^3*d^13 + 570*C
*a^4*b^9*c^5*d^11 + 1218*C*a^4*b^9*c^7*d^9 + 951*C*a^4*b^9*c^9*d^7 + 225*C*
a^4*b^9*c^11*d^5 - 17*C*a^4*b^9*c^13*d^3 + 36*C*a^5*b^8*c^2*d^14 - 172*C*a^
5*b^8*c^4*d^12 - 1004*C*a^5*b^8*c^6*d^10 - 1452*C*a^5*b^8*c^8*d^8 - 732*C*a

^5*b^8*c^10*d^6 - 76*C*a^5*b^8*c^12*d^4 + 4*C*a^5*b^8*c^14*d^2 - 124*C*a^6*
b^7*c^3*d^13 + 336*C*a^6*b^7*c^5*d^11 + 1132*C*a^6*b^7*c^7*d^9 + 964*C*a^6*
b^7*c^9*d^7 + 256*C*a^6*b^7*c^11*d^5 + 4*C*a^6*b^7*c^13*d^3 + 144*C*a^7*b^6
*c^2*d^14 + 196*C*a^7*b^6*c^4*d^12 - 296*C*a^7*b^6*c^6*d^10 - 708*C*a^7*b^6
*c^8*d^8 - 392*C*a^7*b^6*c^10*d^6 - 44*C*a^7*b^6*c^12*d^4 - 237*C*a^8*b^5*c
^3*d^13 - 171*C*a^8*b^5*c^5*d^11 + 223*C*a^8*b^5*c^7*d^9 + 266*C*a^8*b^5*c^
9*d^7 + 58*C*a^8*b^5*c^11*d^5 + 92*C*a^9*b^4*c^2*d^14 + 204*C*a^9*b^4*c^4*d
^12 + 116*C*a^9*b^4*c^6*d^10 - 32*C*a^9*b^4*c^8*d^8 - 32*C*a^9*b^4*c^10*d^6
 - 74*C*a^10*b^3*c^3*d^13 - 90*C*a^10*b^3*c^5*d^11 - 14*C*a^10*b^3*c^7*d^9
+ 16*C*a^10*b^3*c^9*d^7 - 4*C*a^11*b^2*c^2*d^14 - 4*C*a^11*b^2*c^4*d^12 - 1
2*C*a^11*b^2*c^6*d^10 - 8*C*a^11*b^2*c^8*d^8 - A*a^12*b*c*d^15 + C*a^12*b*c
*d^15)/(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^1
4 + 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a
^10*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12
*d^2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^1
1*d^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^
3*c*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*
b*c^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 +
 72*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b
^7*c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^
3 + 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a
^4*b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^
3*d^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 -
64*a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*
b^4*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d
^2 - 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a
^7*b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4
*d^10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a
*b^9*c^13*d - 6*a^9*b*c*d^13) + (tan(e + f*x)*(3*C*a^12*b*d^16 - 3*A*a^12*b
*d^16 + 3*B*b^13*c^15*d + 24*A*a^4*b^9*d^16 + 56*A*a^6*b^7*d^16 + 25*A*a^8*
b^5*d^16 - 10*A*a^10*b^3*d^16 - 16*B*a^5*b^8*d^16 - 48*B*a^7*b^6*d^16 - 36*
B*a^9*b^4*d^16 - 4*B*a^11*b^2*d^16 - 24*A*b^13*c^4*d^12 - 104*A*b^13*c^6*d^
10 - 199*A*b^13*c^8*d^8 - 189*A*b^13*c^10*d^6 - 77*A*b^13*c^12*d^4 - 7*A*b^
13*c^14*d^2 + 4*C*a^6*b^7*d^16 + 23*C*a^8*b^5*d^16 + 22*C*a^10*b^3*d^16 + 8
*B*b^13*c^5*d^11 + 24*B*b^13*c^7*d^9 + 51*B*b^13*c^9*d^7 + 65*B*b^13*c^11*d
^5 + 33*B*b^13*c^13*d^3 - 4*C*b^13*c^6*d^10 + 7*C*b^13*c^8*d^8 + 21*C*b^13*
c^10*d^6 + 5*C*b^13*c^12*d^4 - 5*C*b^13*c^14*d^2 + 48*A*a*b^12*c^3*d^13 + 2
08*A*a*b^12*c^5*d^11 + 472*A*a*b^12*c^7*d^9 + 572*A*a*b^12*c^9*d^7 + 324*A*
a*b^12*c^11*d^5 + 68*A*a*b^12*c^13*d^3 - 48*A*a^3*b^10*c*d^15 + 4*A*a^3*b^1
0*c^15*d - 144*A*a^5*b^8*c*d^15 - 104*A*a^7*b^6*c*d^15 + 4*A*a^9*b^4*c*d^15
 + 12*A*a^11*b^2*c*d^15 - A*a^12*b*c^2*d^14 + 7*A*a^12*b*c^4*d^12 + 5*A*a^1
2*b*c^6*d^10 + 64*B*a*b^12*c^6*d^10 + 100*B*a*b^12*c^8*d^8 - 4*B*a*b^12*c^1
0*d^6 - 52*B*a*b^12*c^12*d^4 - 12*B*a*b^12*c^14*d^2 + 2*B*a^2*b^11*c^15*d +
 24*B*a^4*b^9*c*d^15 - B*a^4*b^9*c^15*d + 120*B*a^6*b^7*c*d^15 + 147*B*a^8*
b^5*c*d^15 + 58*B*a^10*b^3*c*d^15 + 13*B*a^12*b*c^3*d^13 + 5*B*a^12*b*c^5*d

^11 - B*a^12*b*c^7*d^9 + 8*C*a*b^12*c^5*d^11 - 88*C*a*b^12*c^7*d^9 - 236*C*
a*b^12*c^9*d^7 - 180*C*a*b^12*c^11*d^5 - 44*C*a*b^12*c^13*d^3 - 4*C*a^3*b^1
0*c^15*d + 24*C*a^5*b^8*c*d^15 + 8*C*a^7*b^6*c*d^15 - 28*C*a^9*b^4*c*d^15 -
12*C*a^11*b^2*c*d^15 + C*a^12*b*c^2*d^14 - 7*C*a^12*b*c^4*d^12 - 5*C*a^12*
b*c^6*d^10 - 24*A*a^2*b^11*c^4*d^12 - 316*A*a^2*b^11*c^6*d^10 - 838*A*a^2*b
^11*c^8*d^8 - 858*A*a^2*b^11*c^10*d^6 - 346*A*a^2*b^11*c^12*d^4 - 34*A*a^2*
b^11*c^14*d^2 - 192*A*a^3*b^10*c^3*d^13 - 200*A*a^3*b^10*c^5*d^11 + 472*A*a
^3*b^10*c^7*d^9 + 1148*A*a^3*b^10*c^9*d^7 + 756*A*a^3*b^10*c^11*d^5 + 140*A
*a^3*b^10*c^13*d^3 + 200*A*a^4*b^9*c^2*d^14 + 790*A*a^4*b^9*c^4*d^12 + 906*
A*a^4*b^9*c^6*d^10 - 177*A*a^4*b^9*c^8*d^8 - 795*A*a^4*b^9*c^10*d^6 - 353*A
*a^4*b^9*c^12*d^4 - 27*A*a^4*b^9*c^14*d^2 - 936*A*a^5*b^8*c^3*d^13 - 2016*A
*a^5*b^8*c^5*d^11 - 1512*A*a^5*b^8*c^7*d^9 + 72*A*a^5*b^8*c^9*d^7 + 432*A*a
^5*b^8*c^11*d^5 + 72*A*a^5*b^8*c^13*d^3 + 468*A*a^6*b^7*c^2*d^14 + 1768*A*a
^6*b^7*c^4*d^12 + 2524*A*a^6*b^7*c^6*d^10 + 1252*A*a^6*b^7*c^8*d^8 - 84*A*a
^6*b^7*c^12*d^4 - 952*A*a^7*b^6*c^3*d^13 - 2264*A*a^7*b^6*c^5*d^11 - 2088*A
*a^7*b^6*c^7*d^9 - 672*A*a^7*b^6*c^9*d^7 + 283*A*a^8*b^5*c^2*d^14 + 1137*A*
a^8*b^5*c^4*d^12 + 1651*A*a^8*b^5*c^6*d^10 + 898*A*a^8*b^5*c^8*d^8 + 126*A*
a^8*b^5*c^10*d^6 - 268*A*a^9*b^4*c^3*d^13 - 716*A*a^9*b^4*c^5*d^11 - 612*A*
a^9*b^4*c^7*d^9 - 168*A*a^9*b^4*c^9*d^7 + 14*A*a^10*b^3*c^2*d^14 + 166*A*a^
10*b^3*c^4*d^12 + 250*A*a^10*b^3*c^6*d^10 + 108*A*a^10*b^3*c^8*d^8 - 12*A*a
^11*b^2*c^3*d^13 - 60*A*a^11*b^2*c^5*d^11 - 36*A*a^11*b^2*c^7*d^9 - 32*B*a^
2*b^11*c^3*d^13 - 280*B*a^2*b^11*c^5*d^11 - 612*B*a^2*b^11*c^7*d^9 - 474*B*
a^2*b^11*c^9*d^7 - 70*B*a^2*b^11*c^11*d^5 + 42*B*a^2*b^11*c^13*d^3 + 16*B*a
^3*b^10*c^2*d^14 + 240*B*a^3*b^10*c^4*d^12 + 968*B*a^3*b^10*c^6*d^10 + 1348
*B*a^3*b^10*c^8*d^8 + 668*B*a^3*b^10*c^10*d^6 + 60*B*a^3*b^10*c^12*d^4 - 4*
B*a^3*b^10*c^14*d^2 + 8*B*a^4*b^9*c^3*d^13 - 814*B*a^4*b^9*c^5*d^11 - 2034*
B*a^4*b^9*c^7*d^9 - 1731*B*a^4*b^9*c^9*d^7 - 513*B*a^4*b^9*c^11*d^5 - 19*B*
a^4*b^9*c^13*d^3 - 128*B*a^5*b^8*c^2*d^14 + 144*B*a^5*b^8*c^4*d^12 + 1472*B
*a^5*b^8*c^6*d^10 + 2232*B*a^5*b^8*c^8*d^8 + 1176*B*a^5*b^8*c^10*d^6 + 168*
B*a^5*b^8*c^12*d^4 + 8*B*a^5*b^8*c^14*d^2 + 460*B*a^6*b^7*c^3*d^13 - 152*B*
a^6*b^7*c^5*d^11 - 1596*B*a^6*b^7*c^7*d^9 - 1524*B*a^6*b^7*c^9*d^7 - 448*B*
a^6*b^7*c^11*d^5 - 28*B*a^6*b^7*c^13*d^3 - 408*B*a^7*b^6*c^2*d^14 - 576*B*a
^7*b^6*c^4*d^12 + 328*B*a^7*b^6*c^6*d^10 + 1048*B*a^7*b^6*c^8*d^8 + 560*B*a
^7*b^6*c^10*d^6 + 56*B*a^7*b^6*c^12*d^4 + 617*B*a^8*b^5*c^3*d^13 + 587*B*a^
8*b^5*c^5*d^11 - 159*B*a^8*b^5*c^7*d^9 - 346*B*a^8*b^5*c^9*d^7 - 70*B*a^8*b
^5*c^11*d^5 - 316*B*a^9*b^4*c^2*d^14 - 564*B*a^9*b^4*c^4*d^12 - 268*B*a^9*b
^4*c^6*d^10 + 72*B*a^9*b^4*c^8*d^8 + 56*B*a^9*b^4*c^10*d^6 + 210*B*a^10*b^3
*c^3*d^13 + 218*B*a^10*b^3*c^5*d^11 + 38*B*a^10*b^3*c^7*d^9 - 28*B*a^10*b^3
*c^9*d^7 - 52*B*a^11*b^2*c^2*d^14 - 84*B*a^11*b^2*c^4*d^12 - 28*B*a^11*b^2*
c^6*d^10 + 8*B*a^11*b^2*c^8*d^8 - 36*C*a^2*b^11*c^4*d^12 + 52*C*a^2*b^11*c^
6*d^10 + 382*C*a^2*b^11*c^8*d^8 + 474*C*a^2*b^11*c^10*d^6 + 190*C*a^2*b^11*
c^12*d^4 + 10*C*a^2*b^11*c^14*d^2 + 96*C*a^3*b^10*c^3*d^13 + 344*C*a^3*b^10
*c^5*d^11 + 104*C*a^3*b^10*c^7*d^9 - 524*C*a^3*b^10*c^9*d^7 - 468*C*a^3*b^1
0*c^11*d^5 - 92*C*a^3*b^10*c^13*d^3 - 92*C*a^4*b^9*c^2*d^14 - 646*C*a^4*b^9
*c^4*d^12 - 942*C*a^4*b^9*c^6*d^10 - 87*C*a^4*b^9*c^8*d^8 + 543*C*a^4*b^9*c

^10*d^6 + 257*C*a^4*b^9*c^12*d^4 + 15*C*a^4*b^9*c^14*d^2 + 504*C*a^5*b^8*c^3*d^13 + 1512*C*a^5*b^8*c^5*d^11 + 1416*C*a^5*b^8*c^7*d^9 + 144*C*a^5*b^8*c^9*d^7 - 288*C*a^5*b^8*c^11*d^5 - 48*C*a^5*b^8*c^13*d^3 - 204*C*a^6*b^7*c^2*d^14 - 1324*C*a^6*b^7*c^4*d^12 - 2188*C*a^6*b^7*c^6*d^10 - 1168*C*a^6*b^7*c^8*d^8 - 24*C*a^6*b^7*c^10*d^6 + 72*C*a^6*b^7*c^12*d^4 + 568*C*a^7*b^6*c^3*d^13 + 1688*C*a^7*b^6*c^5*d^11 + 1704*C*a^7*b^6*c^7*d^9 + 576*C*a^7*b^6*c^9*d^7 - 79*C*a^8*b^5*c^2*d^14 - 801*C*a^8*b^5*c^4*d^12 - 1387*C*a^8*b^5*c^6*d^10 - 802*C*a^8*b^5*c^8*d^8 - 114*C*a^8*b^5*c^10*d^6 + 172*C*a^9*b^4*c^3*d^13 + 572*C*a^9*b^4*c^5*d^11 + 516*C*a^9*b^4*c^7*d^9 + 144*C*a^9*b^4*c^9*d^7 + 34*C*a^10*b^3*c^2*d^14 - 94*C*a^10*b^3*c^4*d^12 - 202*C*a^10*b^3*c^6*d^10 - 96*C*a^10*b^3*c^8*d^8 + 12*C*a^11*b^2*c^3*d^13 + 60*C*a^11*b^2*c^5*d^11 + 36*C*a^11*b^2*c^7*d^9 + 4*A*a*b^12*c^15*d + 7*B*a^12*b*c*d^15 - 4*C*a*b^12*c^15*d))/(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 + 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 + 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2 - 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^9*c^13*d - 6*a^9*b*c*d^13)) + (36*A^2*a^3*b^8*d^13 - 4*A^2*a^5*b^6*d^13 - 3*A^2*a^7*b^4*d^13 + 16*B^2*a^3*b^8*d^13 + 16*B^2*a^5*b^6*d^13 + B^2*a^7*b^4*d^13 + 2*B^2*a^9*b^2*d^13 + 156*A^2*b^11*c^3*d^10 + 204*A^2*b^11*c^5*d^8 + 85*A^2*b^11*c^7*d^6 + 3*A^2*b^11*c^11*d^2 + 8*C^2*a^5*b^6*d^13 + 9*C^2*a^7*b^4*d^13 + 4*B^2*b^11*c^3*d^10 + 28*B^2*b^11*c^5*d^8 + 45*B^2*b^11*c^7*d^6 + 24*B^2*b^11*c^9*d^4 - B^2*b^11*c^11*d^2 + C^2*b^11*c^7*d^6 + 3*C^2*b^11*c^11*d^2 + 36*A^2*a*b^10*d^13 + 36*A^2*b^11*c*d^12 + 168*A^2*a^2*b^9*c^3*d^10 + 393*A^2*a^2*b^9*c^5*d^8 + 43*A^2*a^2*b^9*c^7*d^6 + 5*A^2*a^2*b^9*c^9*d^4 + 7*A^2*a^2*b^9*c^11*d^2 + 8*A^2*a^3*b^8*c^2*d^11 - 417*A^2*a^3*b^8*c^4*d^9 - 411*A^2*a^3*b^8*c^6*d^7 - 7*A^2*a^3*b^8*c^8*d^5 - 17*A^2*a^3*b^8*c^10*d^3 + 87*A^2*a^4*b^7*c^3*d^10 + 359*A^2*a^4*b^7*c^5*d^8 - 75*A^2*a^4*b^7*c^7*d^6 + 9*A^2*a^4*b^7*c^9*d^4 + 17*A^2*a^5*b^6*c^2*d^11 - 205*A^2*a^5*b^6*c^4*d^9 - 13*A^2*a^5*b^6*c^6*d^7 + 37*A^2*a^5*b^6*c^8*d^5 + A^2*a^6*b^5*c^3*d^10 + 13*A^2*a^6*b^5*c^5*d^8 - 89*A^2*a^6*b^5*c^7*d^6 + 23*A^2*a^7*b^4*c^2*d^11 + 23*A^2*a^7*b^4*c^4*d^9 + 93*A^2*a^7*b^4*c^6*d^7 - 53*A^2*a^8*b^3*c^5*d^8 - 8*A^2*a^9*b^2*c^2*d^11 + 16*A^2*a^9*b^2*c^4*d^9 + 48*B^2 *a^2*b^9*c^3*d^10 - 47*B^2*a^2*b^9*c^5*d^8 + 131*B^2*a^2*b^9*c^7*d^6 + 9*B^2*a^2*b^9*c^9*d^4 - 5*B^2*a^2*b^9*c^11*d^2 + 36*B^2*a^3*b^8*c^2*d^11 + 163*B^2*a^3*b^8*c^4*d^9 - 31*B^2*a^3*b^8*c^6*d^7 - 199*B^2*a^3*b^8*c^8*d^5 + 7*B^2*a^3*b^8*c^10*d^3 - 49*B^2*a^4*b^7*c^3*d^10 - 209*B^2*a^4*b^7*c^5*d^8 + 149*B^2*a^4*b^7*c^7*d^6 - 19*B^2*a^4*b^7*c^9*d^4 - 11*B^2*a^5*b^6*c^2*d^11 + 91*B^2*a^5*b^6*c^4*d^9 - 185*B^2*a^5*b^6*c^6*d^7 - 127*B^2*a^5*b^6*c^8*d^5 - 39*B^2*a^6*b^5*c^3*d^10 + 13*B^2*a^6*b^5*c^5*d^8 + 119*B^2*a^6*b^5*c^7*d^6 - 13*B^2*a^7*b^4*c^2*d^11 + 3*B^2*a^7*b^4*c^4*d^9 - 79*B^2*a^7*b^4*c^6*d^7 - 20*B^2*a^8*b^3*c^3*d^10 + 43*B^2*a^8*b^3*c^5*d^8 + 12*B^2*a^9*b^2*c^2*d^11 - 14*B^2*a^9*b^2*c^4*d^9 + 36*C^2*a^2*b^9*c^3*d^10 + 141*C^2*a^2*b^9*c^5*d^8 - 65*C^2*a^2*b^9*c^7*d^6 + 17*C^2*a^2*b^9*c^9*d^4 + 7*C^2*a^2*b^9*c^11*d^2 + 20*C^2*a^3*b^8*c^2*d^11 - 69*C^2*a^3*b^8*c^4*d^9 + 57*C^2*a^3*b^8*c^6*d^7 + 113*C^2*a^3*b^8*c^8*d^5 - 65*C^2*a^3*b^8*c^10*d^3 + 99*C^2*a^4*b^7*c^3*d^10 + 179*C^2*a^4*b^7*c^5*d^8 - 231*C^2*a^4*b^7*c^7*d^6 - 15*C^2*a^4*b^7*c^9*d^4 + 41*C^2*a^5*b^6*c^2*d^11 - 97*C^2*a^5*b^6*c^4*d^9 + 143*C^2*a^5*b^6*c^6*d^7 + 61*C^2*a^5*b^6*c^8*d^5 - 36*C^2*a^5*b^6*c^10*d^3 - 11*C^2*a^6*b^5*c^3*d^10 - 119*C^2*a^6*b^5*c^5*d^8 - 221*C^2*a^6*b^5*c^7*d^6 - 36*C^2*a^6*b^5*c^9*d^4 + 11*C^2*a^7*b^4*c^2*d^11 - 37*C^2*a^7*b^4*c^4*d^9 + 57*C^2*a^7*b^4*c^6*d^7 - 53*C^2*a^8*b^3*c^5*d^8 - 8*C^2*a^9*b^2*c^2*d^11 + 16*C^2*a^9*b^2*c^4*d^9 - 48*A*B*a^2*b^9*d^13 - 48*A*B*a^4*b^7*d^13 - A*B*a^8*b^3*d^13 + 36*A*C*a^3*b^8*d^13 + 32*A*C*a^5*b^6*d^13 - 6*A*C*a^7*b^4*d^13 - 24*A*B*b^11*c^2*d^11 - 136*A*B*b^11*c^4*d^9 - 200*A*B*b^11*c^6*d^7 - 89*A*B*b^11*c^8*d^5 + 6*A*B*b^11*c^10*d^3 - 24*B*C*a^4*b^7*d^13 - 24*B*C*a^6*b^5*d^13 + B*C*a^8*b^3*d^13 - 12*A*C*b^11*c^3*d^10 + 12*A*C*b^11*c^5*d^8 + 58*A*C*b^11*c^7*d^6 + 36*A*C*b^11*c^9*d^4 - 6*A*C*b^11*c^11*d^2 + 4*B*C*b^11*c^4*d^9 - 4*B*C*b^11*c^6*d^7 - 19*B*C*b^11*c^8*d^5 - 18*B*C*b^11*c^10*d^3 - A^2*a*b^10*c^12*d + 2*A^2*a^10*b*c*d^12 + B^2*a*b^10*c^12*d - 2*B^2*a^10*b*c*d^12 - C^2*a*b^10*c^12*d + 2*C^2*a^10*b*c*d^12 + 24*A^2*a*b^10*c^2*d^11 - 188*A^2*a*b^10*c^4*d^9 - 277*A^2*a*b^10*c^6*d^7 - 27*A^2*a*b^10*c^8*d^5 - 15*A^2*a*b^10*c^10*d^3 - 44*A^2*a^4*b^7*c*d^12 - 29*A^2*a^6*b^5*c*d^12 + A^2*a^8*b^3*c*d^12 - 2*A^2*a^10*b*c^3*d^10 + 20*B^2*a*b^10*c^2*d^11 + 72*B^2*a*b^10*c^4*d^9 + 47*B^2*a*b^10*c^6*d^7 - 89*B^2*a*b^10*c^8*d^5 + 5*B^2*a*b^10*c^10*d^3 + 32*B^2*a^2*b^9*c*d^12 + 16*B^2*a^4*b^7*c*d^12 - 5*B^2*a^6*b^5*c*d^12 - 11*B^2*a^8*b^3*c*d^12 + 2*B^2*a^10*b*c^3*d^10 - 8*C^2*a*b^10*c^4*d^9 - C^2*a*b^10*c^6*d^7 + 69*C^2*a*b^10*c^8*d^5 - 27*C^2*a*b^10*c^10*d^3 + 16*C^2*a^4*b^7*c*d^12 - 5*C^2*a^6*b^5*c*d^12 + C^2*a^8*b^3*c*d^12 - 2*C^2*a^10*b*c^3*d^10 + A*B*a^10*b*d^13 - A*B*b^11*c^12*d - B*C*a^10*b*d^13 + B*C*b^11*c^12*d - 72*A*B*a*b^10*c*d^12 + 2*A*C*a*b^10*c^12*d - 4*A*C*a^10*b*c*d^12 - 160*A*B*a*b^10*c^3*d^10 + 56*A*B*a*b^10*c^5*d^8 + 312*A*B*a*b^10*c^7*d^6 - 8*A*B*a*b^10*c^9*d^4 + A*B*a^2*b^9*c^12*d - 24*A*B*a^3*b^8*c*d^12 + 40*A*B*a^5*b^6*c*d^12 + 32*A*B*a^7*b^4*c*d^12 - 6*A*B*a^10*b*c^2*d^11 + A*B*a^10*b*c^4*d^9 + 84*A*C*a*b^10*c^2*d^11 + 268*A*C*a*b^10*c^4*d^9 + 206*A*C*a*b^10*c^6*d^7 - 150*A*C*a*b^10*c^8*d^5 + 6*A*C*a*b^10*c^10*d^3 + 36*A*C*a^2*b^9*c*d^12 - 8*A*C*a^4*b^7*c*d^12 - 2*A*C*a^6*b^5*c*d^12 - 2*A*C*a^8*b^3*c*d^12 + 4*A*C*a^10*b*c^3*d^10 - 20*B*C*a*b^10*c^3*d^10 - 116*B*C*a*b^10*c

^5*d^8 - 180*B*C*a*b^10*c^7*d^6 + 92*B*C*a*b^10*c^9*d^4 - B*C*a^2*b^9*c^12*
d - 36*B*C*a^3*b^8*c*d^12 + 8*B*C*a^5*b^6*c*d^12 + 4*B*C*a^7*b^4*c*d^12 + 6
*B*C*a^10*b*c^2*d^11 - B*C*a^10*b*c^4*d^9 - 64*A*B*a^2*b^9*c^2*d^11 - 112*A
*B*a^2*b^9*c^4*d^9 - 508*A*B*a^2*b^9*c^6*d^7 - 23*A*B*a^2*b^9*c^8*d^5 + 30*
A*B*a^2*b^9*c^10*d^3 - 112*A*B*a^3*b^8*c^3*d^10 + 480*A*B*a^3*b^8*c^5*d^8 +
584*A*B*a^3*b^8*c^7*d^6 - 56*A*B*a^3*b^8*c^9*d^4 - 8*A*B*a^3*b^8*c^11*d^2
+ 40*A*B*a^4*b^7*c^2*d^11 + 114*A*B*a^4*b^7*c^4*d^9 - 456*A*B*a^4*b^7*c^6*d
^7 + 170*A*B*a^4*b^7*c^8*d^5 + 28*A*B*a^4*b^7*c^10*d^3 - 104*A*B*a^5*b^6*c^
3*d^10 + 368*A*B*a^5*b^6*c^5*d^8 + 104*A*B*a^5*b^6*c^7*d^6 - 56*A*B*a^5*b^6
*c^9*d^4 + 52*A*B*a^6*b^5*c^2*d^11 - 50*A*B*a^6*b^5*c^4*d^9 - 176*A*B*a^6*b
^5*c^6*d^7 + 70*A*B*a^6*b^5*c^8*d^5 + 40*A*B*a^7*b^4*c^3*d^10 + 144*A*B*a^7
*b^4*c^5*d^8 - 56*A*B*a^7*b^4*c^7*d^6 - 30*A*B*a^8*b^3*c^2*d^11 - 105*A*B*a
^8*b^3*c^4*d^9 + 28*A*B*a^8*b^3*c^6*d^7 + 40*A*B*a^9*b^2*c^3*d^10 - 8*A*B*a
^9*b^2*c^5*d^8 - 60*A*C*a^2*b^9*c^3*d^10 - 318*A*C*a^2*b^9*c^5*d^8 + 166*A*
C*a^2*b^9*c^7*d^6 + 14*A*C*a^2*b^9*c^9*d^4 - 14*A*C*a^2*b^9*c^11*d^2 + 188*
A*C*a^3*b^8*c^2*d^11 + 630*A*C*a^3*b^8*c^4*d^9 + 210*A*C*a^3*b^8*c^6*d^7 -
322*A*C*a^3*b^8*c^8*d^5 + 10*A*C*a^3*b^8*c^10*d^3 - 330*A*C*a^4*b^7*c^3*d^1
0 - 754*A*C*a^4*b^7*c^5*d^8 + 162*A*C*a^4*b^7*c^7*d^6 - 30*A*C*a^4*b^7*c^9*
d^4 + 50*A*C*a^5*b^6*c^2*d^11 + 374*A*C*a^5*b^6*c^4*d^9 - 202*A*C*a^5*b^6*c
^6*d^7 - 206*A*C*a^5*b^6*c^8*d^5 - 134*A*C*a^6*b^5*c^3*d^10 - 110*A*C*a^6*b
^5*c^5*d^8 + 166*A*C*a^6*b^5*c^7*d^6 - 34*A*C*a^7*b^4*c^2*d^11 + 14*A*C*a^7
*b^4*c^4*d^9 - 150*A*C*a^7*b^4*c^6*d^7 + 106*A*C*a^8*b^3*c^5*d^8 + 16*A*C*a
^9*b^2*c^2*d^11 - 32*A*C*a^9*b^2*c^4*d^9 - 68*B*C*a^2*b^9*c^2*d^11 - 140*B*
C*a^2*b^9*c^4*d^9 + 208*B*C*a^2*b^9*c^6*d^7 - 109*B*C*a^2*b^9*c^8*d^5 - 30*
B*C*a^2*b^9*c^10*d^3 + 4*B*C*a^3*b^8*c^3*d^10 - 300*B*C*a^3*b^8*c^5*d^8 - 1
40*B*C*a^3*b^8*c^7*d^6 + 272*B*C*a^3*b^8*c^9*d^4 + 8*B*C*a^3*b^8*c^11*d^2 -
160*B*C*a^4*b^7*c^2*d^11 - 174*B*C*a^4*b^7*c^4*d^9 + 420*B*C*a^4*b^7*c^6*d
^7 - 182*B*C*a^4*b^7*c^8*d^5 - 16*B*C*a^4*b^7*c^10*d^3 + 236*B*C*a^5*b^6*c^
3*d^10 - 116*B*C*a^5*b^6*c^5*d^8 + 196*B*C*a^5*b^6*c^7*d^6 + 188*B*C*a^5*b^
6*c^9*d^4 - 64*B*C*a^6*b^5*c^2*d^11 + 110*B*C*a^6*b^5*c^4*d^9 + 236*B*C*a^6
*b^5*c^6*d^7 - 58*B*C*a^6*b^5*c^8*d^5 + 20*B*C*a^7*b^4*c^3*d^10 - 132*B*C*a
^7*b^4*c^5*d^8 + 44*B*C*a^7*b^4*c^7*d^6 + 30*B*C*a^8*b^3*c^2*d^11 + 105*B*C
*a^8*b^3*c^4*d^9 - 28*B*C*a^8*b^3*c^6*d^7 - 40*B*C*a^9*b^2*c^3*d^10 + 8*B*C
*a^9*b^2*c^5*d^8)/(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 +
a^6*b^4*d^14 + 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*
c^6*d^8 + a^10*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 +
4*b^10*c^12*d^2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 2
4*a*b^9*c^11*d^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d
- 12*a^7*b^3*c*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d
^7 - 6*a^9*b*c^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^
8*c^8*d^6 + 72*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11
- 92*a^3*b^7*c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*
b^7*c^11*d^3 + 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*
d^8 + 244*a^4*b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64
*a^5*b^5*c^3*d^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5

*c^9*d^5 - 64*a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10
 + 244*a^6*b^4*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6
*b^4*c^12*d^2 - 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7
*d^7 - 92*a^7*b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*
a^8*b^2*c^4*d^10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^1
0*d^4 - 6*a*b^9*c^13*d - 6*a^9*b*c*d^13) - (tan(e + f*x)*(10*A^2*a^4*b^7*d^
13 - 6*A^2*a^2*b^9*d^13 - 18*A^2*b^11*d^13 + 12*A^2*a^6*b^5*d^13 - 3*A^2*a^
8*b^3*d^13 - 8*B^2*a^2*b^9*d^13 - 8*B^2*a^4*b^7*d^13 - 18*B^2*a^6*b^5*d^13
 - 2*B^2*a^8*b^3*d^13 - 54*A^2*b^11*c^2*d^11 - 18*A^2*b^11*c^4*d^9 + 20*A^2*
b^11*c^6*d^7 - 65*A^2*b^11*c^8*d^5 - 2*C^2*a^4*b^7*d^13 + 6*C^2*a^6*b^5*d^1
3 - 9*C^2*a^8*b^3*d^13 - 2*B^2*b^11*c^2*d^11 - 6*B^2*b^11*c^4*d^9 + 12*B^2*
b^11*c^6*d^7 + 66*B^2*b^11*c^8*d^5 - 18*B^2*b^11*c^10*d^3 + 2*C^2*b^11*c^6*
d^7 - 29*C^2*b^11*c^8*d^5 + 36*C^2*b^11*c^10*d^3 - B^2*a^10*b*d^13 - A^2*b^
11*c^12*d - C^2*b^11*c^12*d - 158*A^2*a^2*b^9*c^2*d^11 - 224*A^2*a^2*b^9*c^
4*d^9 - 252*A^2*a^2*b^9*c^6*d^7 - 194*A^2*a^2*b^9*c^8*d^5 - 2*A^2*a^2*b^9*c
^10*d^3 + 504*A^2*a^3*b^8*c^3*d^10 + 580*A^2*a^3*b^8*c^5*d^8 + 464*A^2*a^3*
b^8*c^7*d^6 + 28*A^2*a^3*b^8*c^9*d^4 - 232*A^2*a^4*b^7*c^2*d^11 - 446*A^2*a
^4*b^7*c^4*d^9 - 452*A^2*a^4*b^7*c^6*d^7 - 128*A^2*a^4*b^7*c^8*d^5 + 248*A^
2*a^5*b^6*c^3*d^10 + 332*A^2*a^5*b^6*c^5*d^8 + 152*A^2*a^5*b^6*c^7*d^6 - 96
*A^2*a^6*b^5*c^2*d^11 - 244*A^2*a^6*b^5*c^4*d^9 - 144*A^2*a^6*b^5*c^6*d^7 +
 120*A^2*a^7*b^4*c^3*d^10 + 132*A^2*a^7*b^4*c^5*d^8 - 34*A^2*a^8*b^3*c^2*d^
11 - 83*A^2*a^8*b^3*c^4*d^9 + 28*A^2*a^9*b^2*c^3*d^10 + 18*B^2*a^2*b^9*c^2*
d^11 + 36*B^2*a^2*b^9*c^4*d^9 + 208*B^2*a^2*b^9*c^6*d^7 + 179*B^2*a^2*b^9*c
^8*d^5 - 32*B^2*a^2*b^9*c^10*d^3 + 128*B^2*a^3*b^8*c^3*d^10 + 180*B^2*a^3*b
^8*c^5*d^8 - 96*B^2*a^3*b^8*c^7*d^6 + 36*B^2*a^3*b^8*c^9*d^4 + 8*B^2*a^3*b^
8*c^11*d^2 + 4*B^2*a^4*b^7*c^2*d^11 - 36*B^2*a^4*b^7*c^4*d^9 + 164*B^2*a^4*
b^7*c^6*d^7 + 76*B^2*a^4*b^7*c^8*d^5 - 16*B^2*a^4*b^7*c^10*d^3 + 208*B^2*a^
5*b^6*c^3*d^10 + 148*B^2*a^5*b^6*c^5*d^8 + 16*B^2*a^5*b^6*c^7*d^6 - 84*B^2*
a^6*b^5*c^2*d^11 - 134*B^2*a^6*b^5*c^4*d^9 - 96*B^2*a^6*b^5*c^6*d^7 - 36*B^
2*a^6*b^5*c^8*d^5 + 40*B^2*a^7*b^4*c^3*d^10 + 20*B^2*a^7*b^4*c^5*d^8 + 48*B
^2*a^7*b^4*c^7*d^6 - 4*B^2*a^8*b^3*c^2*d^11 + 22*B^2*a^8*b^3*c^4*d^9 - 28*B
^2*a^8*b^3*c^6*d^7 - 12*B^2*a^9*b^2*c^3*d^10 + 8*B^2*a^9*b^2*c^5*d^8 - 8*C^
2*a^2*b^9*c^2*d^11 + 16*C^2*a^2*b^9*c^4*d^9 - 132*C^2*a^2*b^9*c^6*d^7 - 104
*C^2*a^2*b^9*c^8*d^5 + 64*C^2*a^2*b^9*c^10*d^3 + 64*C^2*a^3*b^8*c^5*d^8 + 3
56*C^2*a^3*b^8*c^7*d^6 + 64*C^2*a^3*b^8*c^9*d^4 - 12*C^2*a^3*b^8*c^11*d^2 +
 44*C^2*a^4*b^7*c^2*d^11 + 178*C^2*a^4*b^7*c^4*d^9 - 68*C^2*a^4*b^7*c^6*d^7
 - 68*C^2*a^4*b^7*c^8*d^5 + 12*C^2*a^4*b^7*c^10*d^3 - 4*C^2*a^5*b^6*c^3*d^1
0 + 80*C^2*a^5*b^6*c^5*d^8 + 164*C^2*a^5*b^6*c^7*d^6 + 72*C^2*a^5*b^6*c^9*d
^4 + 90*C^2*a^6*b^5*c^2*d^11 + 188*C^2*a^6*b^5*c^4*d^9 + 120*C^2*a^6*b^5*c^
6*d^7 + 6*C^2*a^6*b^5*c^8*d^5 - 18*C^2*a^6*b^5*c^10*d^3 + 36*C^2*a^7*b^4*c^
3*d^10 - 60*C^2*a^7*b^4*c^7*d^6 - 28*C^2*a^8*b^3*c^2*d^11 - 53*C^2*a^8*b^3*
c^4*d^9 + 18*C^2*a^8*b^3*c^6*d^7 + 28*C^2*a^9*b^2*c^3*d^10 + 16*A*B*a^3*b^8
*d^13 + 16*A*B*a^5*b^6*d^13 - 8*A*B*a^7*b^4*d^13 + 2*A*B*a^9*b^2*d^13 - 12*
A*C*a^2*b^9*d^13 + 10*A*C*a^4*b^7*d^13 + 12*A*C*a^8*b^3*d^13 + 36*A*B*b^11*
c^3*d^10 - 36*A*B*b^11*c^5*d^8 - 132*A*B*b^11*c^7*d^6 + 60*A*B*b^11*c^9*d^4

- 4*A*B*b^11*c^11*d^2 + 8*B*C*a^3*b^8*d^13 - 4*B*C*a^5*b^6*d^13 + 20*B*C*a
^7*b^4*d^13 - 2*B*C*a^9*b^2*d^13 - 18*A*C*b^11*c^4*d^9 + 14*A*C*b^11*c^6*d^
7 + 148*A*C*b^11*c^8*d^5 - 18*A*C*b^11*c^10*d^3 + 6*B*C*b^11*c^5*d^8 + 18*B
*C*b^11*c^7*d^6 - 114*B*C*b^11*c^9*d^4 + 10*B*C*b^11*c^11*d^2 + 96*A^2*a*b^
10*c*d^12 - 8*B^2*a*b^10*c*d^12 + 336*A^2*a*b^10*c^3*d^10 + 372*A^2*a*b^10*
c^5*d^8 + 320*A^2*a*b^10*c^7*d^6 + 40*A^2*a*b^10*c^9*d^4 + 4*A^2*a*b^10*c^1
1*d^2 + 136*A^2*a^3*b^8*c*d^12 + 52*A^2*a^5*b^6*c*d^12 + 20*A^2*a^7*b^4*c*d
^12 + 4*A^2*a^9*b^2*c*d^12 - 4*A^2*a^10*b*c^2*d^11 - 16*B^2*a*b^10*c^3*d^10
 + 52*B^2*a*b^10*c^5*d^8 - 72*B^2*a*b^10*c^7*d^6 + 24*B^2*a*b^10*c^9*d^4 +
4*B^2*a*b^10*c^11*d^2 - B^2*a^2*b^9*c^12*d + 48*B^2*a^3*b^8*c*d^12 + 92*B^2
*a^5*b^6*c*d^12 + 36*B^2*a^7*b^4*c*d^12 + 4*B^2*a^9*b^2*c*d^12 + 2*B^2*a^10
*b*c^2*d^11 - B^2*a^10*b*c^4*d^9 - 24*C^2*a*b^10*c^5*d^8 + 140*C^2*a*b^10*c
^7*d^6 + 4*C^2*a*b^10*c^9*d^4 - 8*C^2*a*b^10*c^11*d^2 - 8*C^2*a^3*b^8*c*d^1
2 - 8*C^2*a^5*b^6*c*d^12 + 8*C^2*a^7*b^4*c*d^12 + 4*C^2*a^9*b^2*c*d^12 - 4*
C^2*a^10*b*c^2*d^11 + 24*A*B*a*b^10*d^13 + 12*A*B*b^11*c*d^12 + 2*A*C*b^11*
c^12*d + 2*A*B*a*b^10*c^12*d - 4*A*B*a^10*b*c*d^12 - 24*A*C*a*b^10*c*d^12 -
 2*B*C*a*b^10*c^12*d + 4*B*C*a^10*b*c*d^12 + 16*A*B*a*b^10*c^2*d^11 - 136*A
*B*a*b^10*c^4*d^9 + 8*A*B*a*b^10*c^6*d^7 - 174*A*B*a*b^10*c^8*d^5 - 4*A*B*a
*b^10*c^10*d^3 - 140*A*B*a^2*b^9*c*d^12 - 220*A*B*a^4*b^7*c*d^12 - 68*A*B*a
^6*b^5*c*d^12 - 12*A*B*a^8*b^3*c*d^12 + 4*A*B*a^10*b*c^3*d^10 - 48*A*C*a*b^
10*c^3*d^10 + 84*A*C*a*b^10*c^5*d^8 - 172*A*C*a*b^10*c^7*d^6 + 28*A*C*a*b^1
0*c^9*d^4 + 4*A*C*a*b^10*c^11*d^2 + 16*A*C*a^3*b^8*c*d^12 + 28*A*C*a^5*b^6*
c*d^12 - 28*A*C*a^7*b^4*c*d^12 - 8*A*C*a^9*b^2*c*d^12 + 8*A*C*a^10*b*c^2*d^
11 + 8*B*C*a*b^10*c^2*d^11 + 28*B*C*a*b^10*c^4*d^9 - 188*B*C*a*b^10*c^6*d^7
 + 114*B*C*a*b^10*c^8*d^5 + 16*B*C*a*b^10*c^10*d^3 + 20*B*C*a^2*b^9*c*d^12
- 14*B*C*a^4*b^7*c*d^12 - 52*B*C*a^6*b^5*c*d^12 - 6*B*C*a^8*b^3*c*d^12 - 4*
B*C*a^10*b*c^3*d^10 - 300*A*B*a^2*b^9*c^3*d^10 - 580*A*B*a^2*b^9*c^5*d^8 -
340*A*B*a^2*b^9*c^7*d^6 + 92*A*B*a^2*b^9*c^9*d^4 - 12*A*B*a^2*b^9*c^11*d^2
+ 64*A*B*a^3*b^8*c^2*d^11 + 8*A*B*a^3*b^8*c^4*d^9 + 208*A*B*a^3*b^8*c^6*d^7
 - 200*A*B*a^3*b^8*c^8*d^5 - 420*A*B*a^4*b^7*c^3*d^10 - 596*A*B*a^4*b^7*c^5
*d^8 - 100*A*B*a^4*b^7*c^7*d^6 + 56*A*B*a^4*b^7*c^9*d^4 + 184*A*B*a^5*b^6*c
^2*d^11 + 292*A*B*a^5*b^6*c^4*d^9 + 128*A*B*a^5*b^6*c^6*d^7 - 28*A*B*a^5*b^
6*c^8*d^5 - 84*A*B*a^6*b^5*c^3*d^10 + 60*A*B*a^6*b^5*c^5*d^8 + 92*A*B*a^6*b
^5*c^7*d^6 + 32*A*B*a^7*b^4*c^2*d^11 - 40*A*B*a^7*b^4*c^4*d^9 - 144*A*B*a^7
*b^4*c^6*d^7 - 20*A*B*a^8*b^3*c^3*d^10 + 96*A*B*a^8*b^3*c^5*d^8 + 20*A*B*a^
9*b^2*c^2*d^11 - 30*A*B*a^9*b^2*c^4*d^9 + 112*A*C*a^2*b^9*c^2*d^11 + 172*A*
C*a^2*b^9*c^4*d^9 + 420*A*C*a^2*b^9*c^6*d^7 + 352*A*C*a^2*b^9*c^8*d^5 - 44*
A*C*a^2*b^9*c^10*d^3 + 72*A*C*a^3*b^8*c^3*d^10 + 220*A*C*a^3*b^8*c^5*d^8 -
244*A*C*a^3*b^8*c^7*d^6 + 52*A*C*a^3*b^8*c^9*d^4 + 12*A*C*a^3*b^8*c^11*d^2
+ 242*A*C*a^4*b^7*c^2*d^11 + 304*A*C*a^4*b^7*c^4*d^9 + 484*A*C*a^4*b^7*c^6*
d^7 + 142*A*C*a^4*b^7*c^8*d^5 - 30*A*C*a^4*b^7*c^10*d^3 + 44*A*C*a^5*b^6*c^
3*d^10 + 20*A*C*a^5*b^6*c^5*d^8 - 28*A*C*a^5*b^6*c^7*d^6 + 60*A*C*a^6*b^5*c
^2*d^11 + 92*A*C*a^6*b^5*c^4*d^9 - 12*A*C*a^6*b^5*c^6*d^7 - 60*A*C*a^6*b^5*
c^8*d^5 - 156*A*C*a^7*b^4*c^3*d^10 - 132*A*C*a^7*b^4*c^5*d^8 + 60*A*C*a^7*b
^4*c^7*d^6 + 62*A*C*a^8*b^3*c^2*d^11 + 136*A*C*a^8*b^3*c^4*d^9 - 18*A*C*a^8

*b^3*c^6*d^7 - 56*A*C*a^9*b^2*c^3*d^10 + 160*B*C*a^2*b^9*c^5*d^8 - 80*B*C*a
^2*b^9*c^7*d^6 - 272*B*C*a^2*b^9*c^9*d^4 + 12*B*C*a^2*b^9*c^11*d^2 - 88*B*C
*a^3*b^8*c^2*d^11 - 332*B*C*a^3*b^8*c^4*d^9 - 652*B*C*a^3*b^8*c^6*d^7 + 68*
B*C*a^3*b^8*c^8*d^5 + 36*B*C*a^3*b^8*c^10*d^3 - 66*B*C*a^4*b^7*c^3*d^10 + 2
48*B*C*a^4*b^7*c^5*d^8 - 80*B*C*a^4*b^7*c^7*d^6 - 146*B*C*a^4*b^7*c^9*d^4 -
 6*B*C*a^4*b^7*c^11*d^2 - 172*B*C*a^5*b^6*c^2*d^11 - 448*B*C*a^5*b^6*c^4*d^
9 - 404*B*C*a^5*b^6*c^6*d^7 - 68*B*C*a^5*b^6*c^8*d^5 + 24*B*C*a^5*b^6*c^10*
d^3 - 96*B*C*a^6*b^5*c^3*d^10 - 24*B*C*a^6*b^5*c^5*d^8 + 40*B*C*a^6*b^5*c^7
*d^6 + 36*B*C*a^6*b^5*c^9*d^4 + 28*B*C*a^7*b^4*c^2*d^11 + 100*B*C*a^7*b^4*c
^4*d^9 + 132*B*C*a^7*b^4*c^6*d^7 - 24*B*C*a^7*b^4*c^8*d^5 - 10*B*C*a^8*b^3*
c^3*d^10 - 102*B*C*a^8*b^3*c^5*d^8 + 6*B*C*a^8*b^3*c^7*d^6 - 20*B*C*a^9*b^2
*c^2*d^11 + 30*B*C*a^9*b^2*c^4*d^9))/(a^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^1
4 + a^4*b^6*c^14 + a^6*b^4*d^14 + 2*a^8*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10
*c^4*d^10 + 4*a^10*c^6*d^8 + a^10*c^8*d^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 +
 6*b^10*c^10*d^4 + 4*b^10*c^12*d^2 - 6*a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 3
6*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d^3 - 12*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13
 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c*d^13 - 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d
^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c^9*d^5 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*
c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2*b^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 -
20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*c^5*d^9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^
7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 + 15*a^4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10
 + 211*a^4*b^6*c^6*d^8 + 244*a^4*b^6*c^8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^
4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d^11 - 202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^
7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*b^5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 1
41*a^6*b^4*c^4*d^10 + 244*a^6*b^4*c^6*d^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^
4*c^10*d^4 + 15*a^6*b^4*c^12*d^2 - 68*a^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^
9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*b^3*c^9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8
*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^10 + 98*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d
^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^9*c^13*d - 6*a^9*b*c*d^13)) + (tan(e + f*x
)*(4*B^3*a^5*b^4*d^10 - 12*A^3*a^2*b^7*d^10 - A^3*a^4*b^5*d^10 - 9*A^3*b^9*
d^10 - 27*A^3*b^9*c^2*d^8 - 24*A^3*b^9*c^4*d^6 + 10*A^3*b^9*c^6*d^4 + C^3*a
^4*b^5*d^10 + B^3*b^9*c^3*d^7 + B^3*b^9*c^5*d^5 - C^3*b^9*c^6*d^4 + 3*C^3*b
^9*c^8*d^2 + 9*A^2*C*b^9*d^10 - 58*A^3*a^2*b^7*c^2*d^8 - 46*A^3*a^2*b^7*c^4
*d^6 + 52*A^3*a^3*b^6*c^3*d^7 - 17*A^3*a^4*b^5*c^2*d^8 + 16*B^3*a^2*b^7*c^3
*d^7 - 26*B^3*a^2*b^7*c^5*d^5 - 6*B^3*a^2*b^7*c^7*d^3 - 8*B^3*a^3*b^6*c^2*d
^8 + 20*B^3*a^3*b^6*c^4*d^6 + 28*B^3*a^3*b^6*c^6*d^4 + 17*B^3*a^4*b^5*c^3*d
^7 - 17*B^3*a^4*b^5*c^5*d^5 - 8*B^3*a^5*b^4*c^2*d^8 + 4*B^3*a^5*b^4*c^4*d^6
 + 4*C^3*a^2*b^7*c^2*d^8 - 2*C^3*a^2*b^7*c^4*d^6 + 6*C^3*a^2*b^7*c^6*d^4 +
20*C^3*a^3*b^6*c^3*d^7 - 10*C^3*a^4*b^5*c^2*d^8 - 6*C^3*a^4*b^5*c^4*d^6 + 9
*C^3*a^4*b^5*c^6*d^4 + 36*C^3*a^5*b^4*c^3*d^7 - 12*C^3*a^6*b^3*c^2*d^8 + 12
*A^2*B*a*b^8*d^10 + 15*A^2*B*b^9*c*d^9 + 12*A^3*a*b^8*c*d^9 - 4*A*B^2*a^2*b
^7*d^10 - 14*A*B^2*a^4*b^5*d^10 + 20*A^2*B*a^3*b^6*d^10 + 6*A*C^2*a^2*b^7*d
^10 + 6*A*C^2*a^4*b^5*d^10 + 6*A^2*C*a^2*b^7*d^10 - 6*A^2*C*a^4*b^5*d^10 -
7*A*B^2*b^9*c^2*d^8 - 15*A*B^2*b^9*c^4*d^6 - 24*A*B^2*b^9*c^6*d^4 - 4*B*C^2
*a^3*b^6*d^10 - 6*B*C^2*a^5*b^4*d^10 + 45*A^2*B*b^9*c^3*d^7 + 56*A^2*B*b^9*

c^5*d^5 - 6*A^2*B*b^9*c^7*d^3 + 4*B^2*C*a^2*b^7*d^10 + 8*B^2*C*a^4*b^5*d^10
 - 3*B^2*C*a^6*b^3*d^10 + 3*A*C^2*b^9*c^4*d^6 + 21*A*C^2*b^9*c^6*d^4 - 6*A*
C^2*b^9*c^8*d^2 + 27*A^2*C*b^9*c^2*d^8 + 21*A^2*C*b^9*c^4*d^6 - 30*A^2*C*b^
9*c^6*d^4 + 3*A^2*C*b^9*c^8*d^2 - B*C^2*b^9*c^5*d^5 - 9*B*C^2*b^9*c^7*d^3 +
 B^2*C*b^9*c^2*d^8 + 3*B^2*C*b^9*c^4*d^6 + 6*B^2*C*b^9*c^6*d^4 + 36*A^3*a*b
^8*c^3*d^7 - 8*A^3*a*b^8*c^5*d^5 + 20*A^3*a^3*b^6*c*d^9 + 4*B^3*a*b^8*c^2*d
^8 + 12*B^3*a*b^8*c^4*d^6 + 24*B^3*a*b^8*c^6*d^4 + 4*B^3*a^2*b^7*c*d^9 + 2*
B^3*a^4*b^5*c*d^9 + 8*C^3*a*b^8*c^5*d^5 + 4*C^3*a^3*b^6*c*d^9 + 12*C^3*a^5*
b^4*c*d^9 - 12*A*B*C*a*b^8*d^10 - 6*A*B*C*b^9*c*d^9 + 8*A*B^2*a^2*b^7*c^2*d
^8 + 54*A*B^2*a^2*b^7*c^4*d^6 - 22*A*B^2*a^2*b^7*c^6*d^4 - 92*A*B^2*a^3*b^6
*c^3*d^7 - 56*A*B^2*a^3*b^6*c^5*d^5 - 7*A*B^2*a^4*b^5*c^2*d^8 + 55*A*B^2*a^
4*b^5*c^4*d^6 - 16*A*B^2*a^5*b^4*c^3*d^7 + 46*A^2*B*a^2*b^7*c^3*d^7 + 82*A^
2*B*a^2*b^7*c^5*d^5 + 68*A^2*B*a^3*b^6*c^2*d^8 - 16*A^2*B*a^3*b^6*c^4*d^6 -
 33*A^2*B*a^4*b^5*c^3*d^7 + 16*A^2*B*a^5*b^4*c^2*d^8 - 12*A*C^2*a^2*b^7*c^2
*d^8 + 12*A*C^2*a^2*b^7*c^4*d^6 + 6*A*C^2*a^2*b^7*c^6*d^4 + 12*A*C^2*a^3*b^
6*c^3*d^7 + 30*A*C^2*a^4*b^5*c^2*d^8 + 39*A*C^2*a^4*b^5*c^4*d^6 - 9*A*C^2*a
^4*b^5*c^6*d^4 - 72*A*C^2*a^5*b^4*c^3*d^7 + 24*A*C^2*a^6*b^3*c^2*d^8 + 66*A
^2*C*a^2*b^7*c^2*d^8 + 36*A^2*C*a^2*b^7*c^4*d^6 - 12*A^2*C*a^2*b^7*c^6*d^4
 - 84*A^2*C*a^3*b^6*c^3*d^7 - 3*A^2*C*a^4*b^5*c^2*d^8 - 33*A^2*C*a^4*b^5*c^4
*d^6 + 36*A^2*C*a^5*b^4*c^3*d^7 - 12*A^2*C*a^6*b^3*c^2*d^8 - 20*B*C^2*a^2*b
^7*c^3*d^7 + 4*B*C^2*a^2*b^7*c^5*d^5 + 6*B*C^2*a^2*b^7*c^7*d^3 + 8*B*C^2*a^
3*b^6*c^2*d^8 + 32*B*C^2*a^3*b^6*c^4*d^6 - 12*B*C^2*a^3*b^6*c^6*d^4 - 66*B*
C^2*a^4*b^5*c^3*d^7 - 21*B*C^2*a^4*b^5*c^5*d^5 + 9*B*C^2*a^4*b^5*c^7*d^3 +
4*B*C^2*a^5*b^4*c^2*d^8 + 42*B*C^2*a^5*b^4*c^4*d^6 - 12*B*C^2*a^6*b^3*c^3*d
^7 - 2*B^2*C*a^2*b^7*c^2*d^8 - 63*B^2*C*a^2*b^7*c^4*d^6 - 2*B^2*C*a^2*b^7*c
^6*d^4 + 3*B^2*C*a^2*b^7*c^8*d^2 + 32*B^2*C*a^3*b^6*c^3*d^7 + 44*B^2*C*a^3*
b^6*c^5*d^5 - 12*B^2*C*a^3*b^6*c^7*d^3 + 13*B^2*C*a^4*b^5*c^2*d^8 - 73*B^2*
C*a^4*b^5*c^4*d^6 - 18*B^2*C*a^4*b^5*c^6*d^4 + 4*B^2*C*a^5*b^4*c^3*d^7 + 12
*B^2*C*a^5*b^4*c^5*d^5 + 6*B^2*C*a^6*b^3*c^2*d^8 - 3*B^2*C*a^6*b^3*c^4*d^6
 - 16*A*B*C*a^3*b^6*d^10 + 6*A*B*C*a^5*b^4*d^10 - 18*A*B*C*b^9*c^3*d^7 - 28*
A*B*C*b^9*c^5*d^5 + 24*A*B*C*b^9*c^7*d^3 - 16*A*B^2*a*b^8*c*d^9 + 12*A*C^2*
a*b^8*c*d^9 - 24*A^2*C*a*b^8*c*d^9 + 4*B^2*C*a*b^8*c*d^9 - 56*A*B^2*a*b^8*c
^3*d^7 - 28*A*B^2*a*b^8*c^5*d^5 + 12*A*B^2*a*b^8*c^7*d^3 - 4*A*B^2*a^3*b^6*
c*d^9 + 16*A*B^2*a^5*b^4*c*d^9 + 20*A^2*B*a*b^8*c^2*d^8 - 56*A^2*B*a*b^8*c^
4*d^6 - 16*A^2*B*a*b^8*c^6*d^4 - 4*A^2*B*a^2*b^7*c*d^9 - 33*A^2*B*a^4*b^5*c
*d^9 + 36*A*C^2*a*b^8*c^3*d^7 - 24*A*C^2*a*b^8*c^5*d^5 + 12*A*C^2*a^3*b^6*c
*d^9 - 24*A*C^2*a^5*b^4*c*d^9 - 72*A^2*C*a*b^8*c^3*d^7 + 24*A^2*C*a*b^8*c^5
*d^5 - 36*A^2*C*a^3*b^6*c*d^9 + 12*A^2*C*a^5*b^4*c*d^9 - 4*B*C^2*a*b^8*c^2*
d^8 - 14*B*C^2*a*b^8*c^4*d^6 - 4*B*C^2*a*b^8*c^6*d^4 + 6*B*C^2*a*b^8*c^8*d^
2 - 10*B*C^2*a^2*b^7*c*d^9 - 12*B*C^2*a^4*b^5*c*d^9 + 12*B*C^2*a^6*b^3*c*d^
9 + 8*B^2*C*a*b^8*c^3*d^7 + 4*B^2*C*a*b^8*c^5*d^5 - 24*B^2*C*a*b^8*c^7*d^3
 - 8*B^2*C*a^3*b^6*c*d^9 - 16*B^2*C*a^5*b^4*c*d^9 + 28*A*B*C*a^2*b^7*c^3*d^7
 - 32*A*B*C*a^2*b^7*c^5*d^5 + 12*A*B*C*a^2*b^7*c^7*d^3 - 76*A*B*C*a^3*b^6*c
^2*d^8 - 16*A*B*C*a^3*b^6*c^4*d^6 + 12*A*B*C*a^3*b^6*c^6*d^4 + 126*A*B*C*a^
4*b^5*c^3*d^7 + 48*A*B*C*a^4*b^5*c^5*d^5 - 20*A*B*C*a^5*b^4*c^2*d^8 - 42*A*

B*C*a^5*b^4*c^4*d^6 + 12*A*B*C*a^6*b^3*c^3*d^7 - 16*A*B*C*a*b^8*c^2*d^8 + 7
0*A*B*C*a*b^8*c^4*d^6 + 20*A*B*C*a*b^8*c^6*d^4 - 6*A*B*C*a*b^8*c^8*d^2 + 32
*A*B*C*a^2*b^7*c*d^9 + 54*A*B*C*a^4*b^5*c*d^9 - 12*A*B*C*a^6*b^3*c*d^9))/(a
^10*d^14 + b^10*c^14 + 2*a^2*b^8*c^14 + a^4*b^6*c^14 + a^6*b^4*d^14 + 2*a^8
*b^2*d^14 + 4*a^10*c^2*d^12 + 6*a^10*c^4*d^10 + 4*a^10*c^6*d^8 + a^10*c^8*d
^6 + b^10*c^6*d^8 + 4*b^10*c^8*d^6 + 6*b^10*c^10*d^4 + 4*b^10*c^12*d^2 - 6*
a*b^9*c^5*d^9 - 24*a*b^9*c^7*d^7 - 36*a*b^9*c^9*d^5 - 24*a*b^9*c^11*d^3 - 1
2*a^3*b^7*c^13*d - 6*a^5*b^5*c*d^13 - 6*a^5*b^5*c^13*d - 12*a^7*b^3*c*d^13
- 24*a^9*b*c^3*d^11 - 36*a^9*b*c^5*d^9 - 24*a^9*b*c^7*d^7 - 6*a^9*b*c^9*d^5
 + 15*a^2*b^8*c^4*d^10 + 62*a^2*b^8*c^6*d^8 + 98*a^2*b^8*c^8*d^6 + 72*a^2*b
^8*c^10*d^4 + 23*a^2*b^8*c^12*d^2 - 20*a^3*b^7*c^3*d^11 - 92*a^3*b^7*c^5*d^
9 - 168*a^3*b^7*c^7*d^7 - 152*a^3*b^7*c^9*d^5 - 68*a^3*b^7*c^11*d^3 + 15*a^
4*b^6*c^2*d^12 + 90*a^4*b^6*c^4*d^10 + 211*a^4*b^6*c^6*d^8 + 244*a^4*b^6*c^
8*d^6 + 141*a^4*b^6*c^10*d^4 + 34*a^4*b^6*c^12*d^2 - 64*a^5*b^5*c^3*d^11 -
202*a^5*b^5*c^5*d^9 - 288*a^5*b^5*c^7*d^7 - 202*a^5*b^5*c^9*d^5 - 64*a^5*b^
5*c^11*d^3 + 34*a^6*b^4*c^2*d^12 + 141*a^6*b^4*c^4*d^10 + 244*a^6*b^4*c^6*d
^8 + 211*a^6*b^4*c^8*d^6 + 90*a^6*b^4*c^10*d^4 + 15*a^6*b^4*c^12*d^2 - 68*a
^7*b^3*c^3*d^11 - 152*a^7*b^3*c^5*d^9 - 168*a^7*b^3*c^7*d^7 - 92*a^7*b^3*c^
9*d^5 - 20*a^7*b^3*c^11*d^3 + 23*a^8*b^2*c^2*d^12 + 72*a^8*b^2*c^4*d^10 + 9
8*a^8*b^2*c^6*d^8 + 62*a^8*b^2*c^8*d^6 + 15*a^8*b^2*c^10*d^4 - 6*a*b^9*c^13
*d - 6*a^9*b*c*d^13))*root(640*a^15*b*c^7*d^13*f^4 + 640*a*b^15*c^13*d^7*f^
4 + 480*a^15*b*c^9*d^11*f^4 + 480*a^15*b*c^5*d^15*f^4 + 480*a*b^15*c^15*d^5
*f^4 + 480*a*b^15*c^11*d^9*f^4 + 192*a^15*b*c^11*d^9*f^4 + 192*a^15*b*c^3*d
^17*f^4 + 192*a^11*b^5*c*d^19*f^4 + 192*a^5*b^11*c^19*d*f^4 + 192*a*b^15*c^
17*d^3*f^4 + 192*a*b^15*c^9*d^11*f^4 + 128*a^13*b^3*c*d^19*f^4 + 128*a^9*b^
7*c*d^19*f^4 + 128*a^7*b^9*c^19*d*f^4 + 128*a^3*b^13*c^19*d*f^4 + 32*a^15*b
*c^13*d^7*f^4 + 32*a^9*b^7*c^19*d*f^4 + 32*a^7*b^9*c*d^19*f^4 + 32*a*b^15*c
^7*d^13*f^4 + 32*a^15*b*c*d^19*f^4 + 32*a*b^15*c^19*d*f^4 - 47088*a^8*b^8*c
^10*d^10*f^4 + 42432*a^9*b^7*c^9*d^11*f^4 + 42432*a^7*b^9*c^11*d^9*f^4 + 39
328*a^9*b^7*c^11*d^9*f^4 + 39328*a^7*b^9*c^9*d^11*f^4 - 36912*a^8*b^8*c^12*
d^8*f^4 - 36912*a^8*b^8*c^8*d^12*f^4 - 34256*a^10*b^6*c^10*d^10*f^4 - 34256
*a^6*b^10*c^10*d^10*f^4 - 31152*a^10*b^6*c^8*d^12*f^4 - 31152*a^6*b^10*c^12
*d^8*f^4 + 28128*a^9*b^7*c^7*d^13*f^4 + 28128*a^7*b^9*c^13*d^7*f^4 + 24160*
a^11*b^5*c^9*d^11*f^4 + 24160*a^5*b^11*c^11*d^9*f^4 - 23088*a^10*b^6*c^12*d
^8*f^4 - 23088*a^6*b^10*c^8*d^12*f^4 + 22272*a^9*b^7*c^13*d^7*f^4 + 22272*a
^7*b^9*c^7*d^13*f^4 + 19072*a^11*b^5*c^11*d^9*f^4 + 19072*a^5*b^11*c^9*d^11
*f^4 + 18624*a^11*b^5*c^7*d^13*f^4 + 18624*a^5*b^11*c^13*d^7*f^4 - 17328*a^
8*b^8*c^14*d^6*f^4 - 17328*a^8*b^8*c^6*d^14*f^4 - 17232*a^10*b^6*c^6*d^14*f
^4 - 17232*a^6*b^10*c^14*d^6*f^4 - 13520*a^12*b^4*c^8*d^12*f^4 - 13520*a^4*
b^12*c^12*d^8*f^4 - 12464*a^12*b^4*c^10*d^10*f^4 - 12464*a^4*b^12*c^10*d^10
*f^4 + 10880*a^9*b^7*c^5*d^15*f^4 + 10880*a^7*b^9*c^15*d^5*f^4 - 9072*a^10*
b^6*c^14*d^6*f^4 - 9072*a^6*b^10*c^6*d^14*f^4 + 8928*a^11*b^5*c^13*d^7*f^4
+ 8928*a^5*b^11*c^7*d^13*f^4 - 8880*a^12*b^4*c^6*d^14*f^4 - 8880*a^4*b^12*c
^14*d^6*f^4 + 8480*a^11*b^5*c^5*d^15*f^4 + 8480*a^5*b^11*c^15*d^5*f^4 + 720
0*a^9*b^7*c^15*d^5*f^4 + 7200*a^7*b^9*c^5*d^15*f^4 - 6912*a^12*b^4*c^12*d^8

*f^4 - 6912*a^4*b^12*c^8*d^12*f^4 + 6400*a^13*b^3*c^9*d^11*f^4 + 6400*a^3*b ^13*c^11*d^9*f^4 + 5920*a^13*b^3*c^7*d^13*f^4 + 5920*a^3*b^13*c^13*d^7*f^4 - 5392*a^10*b^6*c^4*d^16*f^4 - 5392*a^6*b^10*c^16*d^4*f^4 - 4428*a^8*b^8*c^ 16*d^4*f^4 - 4428*a^8*b^8*c^4*d^16*f^4 + 4128*a^13*b^3*c^11*d^9*f^4 + 4128* a^3*b^13*c^9*d^11*f^4 - 3328*a^12*b^4*c^4*d^16*f^4 - 3328*a^4*b^12*c^16*d^4 *f^4 + 3264*a^13*b^3*c^5*d^15*f^4 + 3264*a^3*b^13*c^15*d^5*f^4 - 2480*a^14* b^2*c^8*d^12*f^4 - 2480*a^2*b^14*c^12*d^8*f^4 + 2240*a^11*b^5*c^15*d^5*f^4 + 2240*a^5*b^11*c^5*d^15*f^4 - 2128*a^12*b^4*c^14*d^6*f^4 - 2128*a^4*b^12*c ^6*d^14*f^4 + 2112*a^9*b^7*c^3*d^17*f^4 + 2112*a^7*b^9*c^17*d^3*f^4 + 2048* a^11*b^5*c^3*d^17*f^4 + 2048*a^5*b^11*c^17*d^3*f^4 - 2000*a^14*b^2*c^6*d^14 *f^4 - 2000*a^2*b^14*c^14*d^6*f^4 - 1792*a^10*b^6*c^16*d^4*f^4 - 1792*a^6*b ^10*c^4*d^16*f^4 - 1776*a^14*b^2*c^10*d^10*f^4 - 1776*a^2*b^14*c^10*d^10*f^ 4 + 1472*a^13*b^3*c^13*d^7*f^4 + 1472*a^3*b^13*c^7*d^13*f^4 + 1088*a^9*b^7* c^17*d^3*f^4 + 1088*a^7*b^9*c^3*d^17*f^4 + 992*a^13*b^3*c^3*d^17*f^4 + 992* a^3*b^13*c^17*d^3*f^4 - 912*a^14*b^2*c^4*d^16*f^4 - 912*a^2*b^14*c^16*d^4*f ^4 - 768*a^10*b^6*c^2*d^18*f^4 - 768*a^6*b^10*c^18*d^2*f^4 - 688*a^14*b^2*c ^12*d^8*f^4 - 688*a^2*b^14*c^8*d^12*f^4 - 592*a^12*b^4*c^2*d^18*f^4 - 592*a ^4*b^12*c^18*d^2*f^4 - 472*a^8*b^8*c^18*d^2*f^4 - 472*a^8*b^8*c^2*d^18*f^4 - 280*a^12*b^4*c^16*d^4*f^4 - 280*a^4*b^12*c^4*d^16*f^4 + 224*a^13*b^3*c^15 *d^5*f^4 + 224*a^11*b^5*c^17*d^3*f^4 + 224*a^5*b^11*c^3*d^17*f^4 + 224*a^3* b^13*c^5*d^15*f^4 - 208*a^14*b^2*c^2*d^18*f^4 - 208*a^2*b^14*c^18*d^2*f^4 - 112*a^14*b^2*c^14*d^6*f^4 - 112*a^10*b^6*c^18*d^2*f^4 - 112*a^6*b^10*c^2*d ^18*f^4 - 112*a^2*b^14*c^6*d^14*f^4 - 80*b^16*c^14*d^6*f^4 - 60*b^16*c^16*d ^4*f^4 - 60*b^16*c^12*d^8*f^4 - 24*b^16*c^18*d^2*f^4 - 24*b^16*c^10*d^10*f^ 4 - 4*b^16*c^8*d^12*f^4 - 80*a^16*c^6*d^14*f^4 - 60*a^16*c^8*d^12*f^4 - 60* a^16*c^4*d^16*f^4 - 24*a^16*c^10*d^10*f^4 - 24*a^16*c^2*d^18*f^4 - 4*a^16*c ^12*d^8*f^4 - 24*a^12*b^4*d^20*f^4 - 16*a^14*b^2*d^20*f^4 - 16*a^10*b^6*d^2 0*f^4 - 4*a^8*b^8*d^20*f^4 - 24*a^4*b^12*c^20*f^4 - 16*a^6*b^10*c^20*f^4 - 16*a^2*b^14*c^20*f^4 - 4*a^8*b^8*c^20*f^4 - 4*b^16*c^20*f^4 - 4*a^16*d^20*f ^4 + 56*A*C*a*b^11*c^13*d*f^2 - 48*A*C*a^11*b*c*d^13*f^2 + 48*A*C*a*b^11*c* d^13*f^2 + 5904*B*C*a^6*b^6*c^7*d^7*f^2 - 5016*B*C*a^5*b^7*c^8*d^6*f^2 - 46 08*B*C*a^7*b^5*c^6*d^8*f^2 - 4512*B*C*a^5*b^7*c^6*d^8*f^2 - 4384*B*C*a^7*b^ 5*c^8*d^6*f^2 + 3056*B*C*a^8*b^4*c^7*d^7*f^2 + 2256*B*C*a^4*b^8*c^7*d^7*f^2 - 1824*B*C*a^3*b^9*c^8*d^6*f^2 + 1632*B*C*a^9*b^3*c^4*d^10*f^2 - 1400*B*C* a^8*b^4*c^3*d^11*f^2 - 1320*B*C*a^4*b^8*c^11*d^3*f^2 - 1248*B*C*a^3*b^9*c^6 *d^8*f^2 + 1152*B*C*a^3*b^9*c^10*d^4*f^2 - 1072*B*C*a^9*b^3*c^6*d^8*f^2 + 1 068*B*C*a^6*b^6*c^9*d^5*f^2 - 1004*B*C*a^4*b^8*c^5*d^9*f^2 - 968*B*C*a^6*b^ 6*c^3*d^11*f^2 - 864*B*C*a^8*b^4*c^5*d^9*f^2 - 828*B*C*a^4*b^8*c^9*d^5*f^2 - 792*B*C*a^4*b^8*c^3*d^11*f^2 - 792*B*C*a^2*b^10*c^11*d^3*f^2 - 776*B*C*a^ 9*b^3*c^8*d^6*f^2 + 688*B*C*a^7*b^5*c^4*d^10*f^2 - 672*B*C*a^10*b^2*c^3*d^1 1*f^2 - 592*B*C*a^2*b^10*c^9*d^5*f^2 + 544*B*C*a^10*b^2*c^7*d^7*f^2 - 492*B *C*a^2*b^10*c^5*d^9*f^2 + 480*B*C*a^5*b^7*c^10*d^4*f^2 - 392*B*C*a^10*b^2*c ^5*d^9*f^2 + 332*B*C*a^8*b^4*c^9*d^5*f^2 - 328*B*C*a^6*b^6*c^11*d^3*f^2 + 3 20*B*C*a^9*b^3*c^2*d^12*f^2 + 272*B*C*a^3*b^9*c^12*d^2*f^2 - 248*B*C*a^5*b^ 7*c^4*d^10*f^2 - 248*B*C*a^2*b^10*c^3*d^11*f^2 - 208*B*C*a^7*b^5*c^10*d^4*f

^2 - 192*B*C*a^5*b^7*c^2*d^12*f^2 + 144*B*C*a^2*b^10*c^7*d^7*f^2 - 96*B*C*a
^3*b^9*c^4*d^10*f^2 + 88*B*C*a^5*b^7*c^12*d^2*f^2 - 72*B*C*a^8*b^4*c^11*d^3
*f^2 + 48*B*C*a^9*b^3*c^10*d^4*f^2 - 48*B*C*a^7*b^5*c^12*d^2*f^2 - 48*B*C*a
^7*b^5*c^2*d^12*f^2 - 48*B*C*a^3*b^9*c^2*d^12*f^2 - 12*B*C*a^10*b^2*c^9*d^5
*f^2 + 4*B*C*a^6*b^6*c^5*d^9*f^2 + 5824*A*C*a^7*b^5*c^5*d^9*f^2 - 4378*A*C*
a^8*b^4*c^6*d^8*f^2 + 4296*A*C*a^5*b^7*c^5*d^9*f^2 - 3912*A*C*a^6*b^6*c^6*d
^8*f^2 - 3672*A*C*a^5*b^7*c^9*d^5*f^2 + 3594*A*C*a^4*b^8*c^8*d^6*f^2 + 3236
*A*C*a^6*b^6*c^8*d^6*f^2 + 2816*A*C*a^9*b^3*c^5*d^9*f^2 + 2624*A*C*a^3*b^9*
c^5*d^9*f^2 + 2432*A*C*a^7*b^5*c^7*d^7*f^2 - 2366*A*C*a^8*b^4*c^4*d^10*f^2
+ 2298*A*C*a^4*b^8*c^10*d^4*f^2 + 1872*A*C*a^3*b^9*c^7*d^7*f^2 + 1848*A*C*a
^6*b^6*c^10*d^4*f^2 - 1644*A*C*a^6*b^6*c^4*d^10*f^2 - 1488*A*C*a^7*b^5*c^9*
d^5*f^2 - 1408*A*C*a^3*b^9*c^9*d^5*f^2 - 1308*A*C*a^4*b^8*c^6*d^8*f^2 + 124
8*A*C*a^5*b^7*c^7*d^7*f^2 - 1012*A*C*a^10*b^2*c^6*d^8*f^2 + 1008*A*C*a^7*b^
5*c^3*d^11*f^2 + 992*A*C*a^5*b^7*c^3*d^11*f^2 + 928*A*C*a^3*b^9*c^3*d^11*f^
2 + 848*A*C*a^9*b^3*c^7*d^7*f^2 + 636*A*C*a^2*b^10*c^8*d^6*f^2 - 628*A*C*a^
10*b^2*c^4*d^10*f^2 - 600*A*C*a^2*b^10*c^6*d^8*f^2 - 576*A*C*a^5*b^7*c^11*d
^3*f^2 + 572*A*C*a^2*b^10*c^10*d^4*f^2 + 464*A*C*a^8*b^4*c^8*d^6*f^2 + 304*
A*C*a^6*b^6*c^2*d^12*f^2 - 304*A*C*a^4*b^8*c^4*d^10*f^2 + 296*A*C*a^4*b^8*c
^2*d^12*f^2 + 260*A*C*a^8*b^4*c^10*d^4*f^2 - 232*A*C*a^9*b^3*c^9*d^5*f^2 -
232*A*C*a^2*b^10*c^12*d^2*f^2 + 228*A*C*a^10*b^2*c^2*d^12*f^2 - 188*A*C*a^2
*b^10*c^4*d^10*f^2 + 144*A*C*a^3*b^9*c^11*d^3*f^2 + 116*A*C*a^6*b^6*c^12*d^
2*f^2 + 112*A*C*a^9*b^3*c^3*d^11*f^2 - 112*A*C*a^7*b^5*c^11*d^3*f^2 + 92*A*
C*a^10*b^2*c^8*d^6*f^2 + 74*A*C*a^4*b^8*c^12*d^2*f^2 + 62*A*C*a^8*b^4*c^2*d
^12*f^2 + 40*A*C*a^2*b^10*c^2*d^12*f^2 - 7008*A*B*a^6*b^6*c^7*d^7*f^2 - 403
2*A*B*a^4*b^8*c^7*d^7*f^2 + 3952*A*B*a^7*b^5*c^8*d^6*f^2 + 3648*A*B*a^5*b^7
*c^8*d^6*f^2 - 3392*A*B*a^8*b^4*c^7*d^7*f^2 + 3264*A*B*a^7*b^5*c^6*d^8*f^2
- 2992*A*B*a^5*b^7*c^4*d^10*f^2 - 2368*A*B*a^7*b^5*c^4*d^10*f^2 - 2304*A*B*
a^3*b^9*c^4*d^10*f^2 - 1968*A*B*a^6*b^6*c^9*d^5*f^2 - 1872*A*B*a^9*b^3*c^4*
d^10*f^2 - 1728*A*B*a^2*b^10*c^7*d^7*f^2 + 1712*A*B*a^8*b^4*c^3*d^11*f^2 +
1536*A*B*a^5*b^7*c^6*d^8*f^2 - 1536*A*B*a^3*b^9*c^10*d^4*f^2 - 1392*A*B*a^5
*b^7*c^2*d^12*f^2 + 1328*A*B*a^6*b^6*c^3*d^11*f^2 - 1104*A*B*a^3*b^9*c^2*d^
12*f^2 - 1056*A*B*a^3*b^9*c^6*d^8*f^2 + 976*A*B*a^9*b^3*c^6*d^8*f^2 + 960*A
*B*a^4*b^8*c^11*d^3*f^2 + 936*A*B*a^8*b^4*c^5*d^9*f^2 - 912*A*B*a^5*b^7*c^1
0*d^4*f^2 + 848*A*B*a^9*b^3*c^8*d^6*f^2 - 816*A*B*a^7*b^5*c^2*d^12*f^2 + 81
6*A*B*a^4*b^8*c^3*d^11*f^2 + 768*A*B*a^10*b^2*c^3*d^11*f^2 + 672*A*B*a^3*b^
9*c^8*d^6*f^2 - 632*A*B*a^8*b^4*c^9*d^5*f^2 - 608*A*B*a^2*b^10*c^9*d^5*f^2
- 552*A*B*a^4*b^8*c^9*d^5*f^2 - 544*A*B*a^10*b^2*c^7*d^7*f^2 - 480*A*B*a^2*
b^10*c^5*d^9*f^2 + 464*A*B*a^10*b^2*c^5*d^9*f^2 - 464*A*B*a^9*b^3*c^2*d^12*
f^2 + 432*A*B*a^2*b^10*c^11*d^3*f^2 - 368*A*B*a^3*b^9*c^12*d^2*f^2 - 256*A*
B*a^6*b^6*c^5*d^9*f^2 - 208*A*B*a^5*b^7*c^12*d^2*f^2 + 176*A*B*a^4*b^8*c^5*
d^9*f^2 + 112*A*B*a^7*b^5*c^10*d^4*f^2 + 112*A*B*a^6*b^6*c^11*d^3*f^2 - 16*
A*B*a^2*b^10*c^3*d^11*f^2 - 576*B*C*a*b^11*c^8*d^6*f^2 + 400*B*C*a^11*b*c^4
*d^10*f^2 - 288*B*C*a*b^11*c^6*d^8*f^2 - 176*B*C*a^11*b*c^6*d^8*f^2 + 128*B
*C*a*b^11*c^10*d^4*f^2 - 108*B*C*a^4*b^8*c*d^13*f^2 - 104*B*C*a*b^11*c^4*d^
10*f^2 - 92*B*C*a^4*b^8*c^13*d*f^2 - 60*B*C*a^8*b^4*c*d^13*f^2 - 60*B*C*a^6

*b^6*c*d^13*f^2 + 48*B*C*a^11*b*c^2*d^12*f^2 - 40*B*C*a^2*b^10*c*d^13*f^2 -
28*B*C*a^2*b^10*c^13*d*f^2 - 24*B*C*a*b^11*c^12*d^2*f^2 + 20*B*C*a^10*b^2*
c*d^13*f^2 - 16*B*C*a*b^11*c^2*d^12*f^2 + 12*B*C*a^6*b^6*c^13*d*f^2 + 912*A
*C*a*b^11*c^7*d^7*f^2 + 808*A*C*a*b^11*c^5*d^9*f^2 + 432*A*C*a^11*b*c^5*d^9
*f^2 + 336*A*C*a*b^11*c^3*d^11*f^2 + 224*A*C*a*b^11*c^11*d^3*f^2 - 112*A*C*
a^11*b*c^3*d^11*f^2 + 112*A*C*a^3*b^9*c*d^13*f^2 - 88*A*C*a^9*b^3*c*d^13*f^
2 + 80*A*C*a^3*b^9*c^13*d*f^2 + 56*A*C*a^5*b^7*c*d^13*f^2 + 48*A*C*a*b^11*c
^9*d^5*f^2 - 40*A*C*a^5*b^7*c^13*d*f^2 - 16*A*C*a^11*b*c^7*d^7*f^2 + 16*A*C
*a^7*b^5*c*d^13*f^2 - 496*A*B*a*b^11*c^4*d^10*f^2 - 400*A*B*a^11*b*c^4*d^10
*f^2 + 288*A*B*a*b^11*c^8*d^6*f^2 - 288*A*B*a*b^11*c^6*d^8*f^2 - 272*A*B*a*
b^11*c^2*d^12*f^2 + 240*A*B*a^6*b^6*c*d^13*f^2 - 224*A*B*a*b^11*c^10*d^4*f^
2 + 192*A*B*a^8*b^4*c*d^13*f^2 + 192*A*B*a^4*b^8*c*d^13*f^2 + 176*A*B*a^11*
b*c^6*d^8*f^2 + 104*A*B*a^4*b^8*c^13*d*f^2 - 48*A*B*a^11*b*c^2*d^12*f^2 + 1
6*A*B*a^10*b^2*c*d^13*f^2 + 16*A*B*a^2*b^10*c^13*d*f^2 + 16*A*B*a^2*b^10*c*
d^13*f^2 - 112*B*C*b^12*c^11*d^3*f^2 + 4*B*C*b^12*c^5*d^9*f^2 + 150*A*C*b^1
2*c^10*d^4*f^2 - 80*B*C*a^12*c^3*d^11*f^2 + 66*A*C*b^12*c^8*d^6*f^2 - 30*A*
C*b^12*c^12*d^2*f^2 + 24*B*C*a^12*c^5*d^9*f^2 - 12*A*C*b^12*c^4*d^10*f^2 -
576*A*B*b^12*c^7*d^7*f^2 - 432*A*B*b^12*c^9*d^5*f^2 - 400*A*B*b^12*c^5*d^9*
f^2 - 144*A*B*b^12*c^3*d^11*f^2 - 96*B*C*a^7*b^5*d^14*f^2 - 72*B*C*a^5*b^7*
d^14*f^2 - 66*A*C*a^12*c^4*d^10*f^2 + 54*A*C*a^12*c^2*d^12*f^2 - 32*A*B*b^1
2*c^11*d^3*f^2 - 24*B*C*a^9*b^3*d^14*f^2 - 16*B*C*a^3*b^9*d^14*f^2 + 2*A*C*
a^12*c^6*d^8*f^2 + 116*A*C*a^6*b^6*d^14*f^2 + 100*A*C*a^4*b^8*d^14*f^2 + 80
*A*B*a^12*c^3*d^11*f^2 + 24*A*C*a^2*b^10*d^14*f^2 - 24*A*B*a^12*c^5*d^9*f^2
+ 22*A*C*a^8*b^4*d^14*f^2 + 16*B*C*a^3*b^9*c^14*f^2 + 8*A*C*a^10*b^2*d^14*
f^2 - 192*A*B*a^5*b^7*d^14*f^2 - 176*A*B*a^3*b^9*d^14*f^2 - 48*A*B*a^7*b^5*
d^14*f^2 - 28*A*C*a^2*b^10*c^14*f^2 + 2*A*C*a^4*b^8*c^14*f^2 - 16*A*B*a^3*b
^9*c^14*f^2 + 2508*C^2*a^6*b^6*c^6*d^8*f^2 + 2376*C^2*a^5*b^7*c^9*d^5*f^2 +
2357*C^2*a^8*b^4*c^6*d^8*f^2 - 2048*C^2*a^7*b^5*c^5*d^9*f^2 + 1304*C^2*a^3
*b^9*c^9*d^5*f^2 + 1303*C^2*a^8*b^4*c^4*d^10*f^2 + 1212*C^2*a^6*b^6*c^4*d^1
0*f^2 - 1203*C^2*a^4*b^8*c^8*d^6*f^2 - 1192*C^2*a^9*b^3*c^5*d^9*f^2 + 1062*
C^2*a^4*b^8*c^6*d^8*f^2 + 984*C^2*a^7*b^5*c^9*d^5*f^2 - 952*C^2*a^6*b^6*c^8
*d^6*f^2 + 768*C^2*a^5*b^7*c^7*d^7*f^2 - 681*C^2*a^4*b^8*c^10*d^4*f^2 - 672
*C^2*a^5*b^7*c^5*d^9*f^2 - 480*C^2*a^6*b^6*c^10*d^4*f^2 + 458*C^2*a^10*b^2*
c^6*d^8*f^2 - 448*C^2*a^7*b^5*c^7*d^7*f^2 + 422*C^2*a^4*b^8*c^4*d^10*f^2 +
372*C^2*a^2*b^10*c^6*d^8*f^2 + 360*C^2*a^5*b^7*c^11*d^3*f^2 + 312*C^2*a^3*b
^9*c^7*d^7*f^2 + 278*C^2*a^10*b^2*c^4*d^10*f^2 - 232*C^2*a^9*b^3*c^7*d^7*f^
2 + 194*C^2*a^2*b^10*c^12*d^2*f^2 + 176*C^2*a^9*b^3*c^9*d^5*f^2 + 152*C^2*a
^5*b^7*c^3*d^11*f^2 + 124*C^2*a^2*b^10*c^4*d^10*f^2 - 120*C^2*a^7*b^5*c^3*d
^11*f^2 - 114*C^2*a^10*b^2*c^2*d^12*f^2 - 102*C^2*a^2*b^10*c^8*d^6*f^2 + 10
1*C^2*a^4*b^8*c^12*d^2*f^2 + 100*C^2*a^6*b^6*c^2*d^12*f^2 - 88*C^2*a^3*b^9*
c^5*d^9*f^2 + 77*C^2*a^8*b^4*c^2*d^12*f^2 + 72*C^2*a^3*b^9*c^11*d^3*f^2 - 6
4*C^2*a^10*b^2*c^8*d^6*f^2 + 64*C^2*a^3*b^9*c^3*d^11*f^2 - 58*C^2*a^2*b^10*
c^10*d^4*f^2 + 56*C^2*a^7*b^5*c^11*d^3*f^2 + 56*C^2*a^6*b^6*c^12*d^2*f^2 +
40*C^2*a^9*b^3*c^3*d^11*f^2 + 36*C^2*a^8*b^4*c^12*d^2*f^2 + 32*C^2*a^4*b^8*
c^2*d^12*f^2 + 26*C^2*a^8*b^4*c^10*d^4*f^2 + 16*C^2*a^2*b^10*c^2*d^12*f^2 +

2*C^2*a^8*b^4*c^8*d^6*f^2 + 2277*B^2*a^4*b^8*c^8*d^6*f^2 + 2144*B^2*a^7*b^5*c^5*d^9*f^2 - 2112*B^2*a^5*b^7*c^9*d^5*f^2 + 2028*B^2*a^6*b^6*c^8*d^6*f^2 - 1671*B^2*a^8*b^4*c^6*d^8*f^2 + 1275*B^2*a^4*b^8*c^10*d^4*f^2 + 1176*B^2*a^5*b^7*c^5*d^9*f^2 + 1096*B^2*a^9*b^3*c^5*d^9*f^2 - 1044*B^2*a^6*b^6*c^6*d^8*f^2 + 984*B^2*a^6*b^6*c^10*d^4*f^2 - 968*B^2*a^3*b^9*c^9*d^5*f^2 - 888*B^2*a^7*b^5*c^9*d^5*f^2 + 672*B^2*a^7*b^5*c^7*d^7*f^2 + 664*B^2*a^3*b^9*c^5*d^9*f^2 - 649*B^2*a^8*b^4*c^4*d^10*f^2 + 618*B^2*a^2*b^10*c^8*d^6*f^2 + 514*B^2*a^4*b^8*c^4*d^10*f^2 + 460*B^2*a^6*b^6*c^2*d^12*f^2 + 422*B^2*a^8*b^4*c^8*d^6*f^2 + 406*B^2*a^2*b^10*c^10*d^4*f^2 - 382*B^2*a^10*b^2*c^6*d^8*f^2 + 368*B^2*a^4*b^8*c^2*d^12*f^2 - 312*B^2*a^5*b^7*c^11*d^3*f^2 + 312*B^2*a^3*b^9*c^7*d^7*f^2 + 248*B^2*a^9*b^3*c^7*d^7*f^2 + 245*B^2*a^8*b^4*c^2*d^12*f^2 - 192*B^2*a^5*b^7*c^7*d^7*f^2 - 184*B^2*a^9*b^3*c^3*d^11*f^2 + 182*B^2*a^10*b^2*c^2*d^12*f^2 + 176*B^2*a^3*b^9*c^3*d^11*f^2 + 174*B^2*a^4*b^8*c^6*d^8*f^2 - 170*B^2*a^10*b^2*c^4*d^10*f^2 - 152*B^2*a^9*b^3*c^9*d^5*f^2 + 152*B^2*a^2*b^10*c^4*d^10*f^2 + 142*B^2*a^8*b^4*c^10*d^4*f^2 - 90*B^2*a^2*b^10*c^12*d^2*f^2 + 88*B^2*a^2*b^10*c^2*d^12*f^2 + 84*B^2*a^10*b^2*c^8*d^6*f^2 + 84*B^2*a^2*b^10*c^6*d^8*f^2 + 60*B^2*a^6*b^6*c^12*d^2*f^2 - 56*B^2*a^7*b^5*c^11*d^3*f^2 + 53*B^2*a^4*b^8*c^12*d^2*f^2 + 24*B^2*a^7*b^5*c^3*d^11*f^2 + 24*B^2*a^6*b^6*c^4*d^10*f^2 + 24*B^2*a^3*b^9*c^11*d^3*f^2 - 8*B^2*a^5*b^7*c^3*d^11*f^2 + 4566*A^2*a^4*b^8*c^6*d^8*f^2 + 4284*A^2*a^6*b^6*c^6*d^8*f^2 - 3776*A^2*a^7*b^5*c^5*d^9*f^2 - 3624*A^2*a^5*b^7*c^5*d^9*f^2 + 3122*A^2*a^4*b^8*c^4*d^10*f^2 + 3108*A^2*a^2*b^10*c^6*d^8*f^2 + 2741*A^2*a^8*b^4*c^6*d^8*f^2 + 2592*A^2*a^6*b^6*c^4*d^10*f^2 - 2536*A^2*a^3*b^9*c^5*d^9*f^2 + 2224*A^2*a^2*b^10*c^4*d^10*f^2 - 2184*A^2*a^3*b^9*c^7*d^7*f^2 - 2016*A^2*a^5*b^7*c^7*d^7*f^2 - 1984*A^2*a^7*b^5*c^7*d^7*f^2 + 1626*A^2*a^2*b^10*c^8*d^6*f^2 - 1624*A^2*a^9*b^3*c^5*d^9*f^2 + 1603*A^2*a^8*b^4*c^4*d^10*f^2 + 1296*A^2*a^5*b^7*c^9*d^5*f^2 - 1144*A^2*a^5*b^7*c^3*d^11*f^2 - 992*A^2*a^3*b^9*c^3*d^11*f^2 + 968*A^2*a^4*b^8*c^2*d^12*f^2 - 888*A^2*a^7*b^5*c^3*d^11*f^2 + 849*A^2*a^4*b^8*c^8*d^6*f^2 + 808*A^2*a^2*b^10*c^2*d^12*f^2 - 616*A^2*a^9*b^3*c^7*d^7*f^2 + 554*A^2*a^10*b^2*c^6*d^8*f^2 + 504*A^2*a^7*b^5*c^9*d^5*f^2 - 504*A^2*a^6*b^6*c^10*d^4*f^2 + 460*A^2*a^6*b^6*c^2*d^12*f^2 + 350*A^2*a^10*b^2*c^4*d^10*f^2 + 350*A^2*a^2*b^10*c^10*d^4*f^2 - 321*A^2*a^4*b^8*c^10*d^4*f^2 + 216*A^2*a^5*b^7*c^11*d^3*f^2 - 216*A^2*a^3*b^9*c^11*d^3*f^2 + 182*A^2*a^2*b^10*c^12*d^2*f^2 - 152*A^2*a^9*b^3*c^3*d^11*f^2 - 124*A^2*a^6*b^6*c^8*d^6*f^2 - 114*A^2*a^10*b^2*c^2*d^12*f^2 + 104*A^2*a^3*b^9*c^9*d^5*f^2 + 77*A^2*a^8*b^4*c^2*d^12*f^2 + 74*A^2*a^8*b^4*c^8*d^6*f^2 - 70*A^2*a^8*b^4*c^10*d^4*f^2 + 56*A^2*a^9*b^3*c^9*d^5*f^2 + 56*A^2*a^7*b^5*c^11*d^3*f^2 + 41*A^2*a^4*b^8*c^12*d^2*f^2 - 28*A^2*a^10*b^2*c^8*d^6*f^2 - 28*A^2*a^6*b^6*c^12*d^2*f^2 + 12*B*C*b^12*c^13*d*f^2 + 24*B*C*a^12*c*d^13*f^2 - 24*A*B*b^12*c^13*d*f^2 - 24*A*B*b^12*c*d^13*f^2 - 16*B*C*a^11*b*d^14*f^2 - 24*A*B*a^12*c*d^13*f^2 - 16*B*C*a*b^11*c^14*f^2 - 48*A*B*a*b^11*d^14*f^2 + 16*A*B*a^11*b*d^14*f^2 + 16*A*B*a*b^11*c^14*f^2 - 216*C^2*a^11*b*c^5*d^9*f^2 + 216*C^2*a*b^11*c^9*d^5*f^2 + 56*C^2*a^11*b*c^3*d^11*f^2 + 56*C^2*a^9*b^3*c*d^13*f^2 + 56*C^2*a^5*b^7*c*d^13*f^2 + 40*C^2*a^7*b^5*c*d^13*f^2 - 40*C^2*a*b^11*c^11*d^3*f^2 + 32*C^2*a^5*b^7*c^13*d*f^2 - 24*C^2*a*b^11*c^7*d^7*f^2 - 16*C^

2*a^3*b^9*c^13*d*f^2 + 16*C^2*a^3*b^9*c*d^13*f^2 + 8*C^2*a^11*b*c^7*d^7*f^2
 - 8*C^2*a*b^11*c^5*d^9*f^2 + 264*B^2*a*b^11*c^7*d^7*f^2 + 224*B^2*a*b^11*c
^5*d^9*f^2 + 168*B^2*a^11*b*c^5*d^9*f^2 - 112*B^2*a^9*b^3*c*d^13*f^2 - 104*
B^2*a^11*b*c^3*d^11*f^2 - 104*B^2*a^7*b^5*c*d^13*f^2 + 96*B^2*a*b^11*c^3*d^
11*f^2 + 88*B^2*a*b^11*c^11*d^3*f^2 - 72*B^2*a*b^11*c^9*d^5*f^2 - 64*B^2*a^
5*b^7*c*d^13*f^2 + 32*B^2*a^3*b^9*c^13*d*f^2 - 24*B^2*a^11*b*c^7*d^7*f^2 -
24*B^2*a^5*b^7*c^13*d*f^2 + 16*B^2*a^3*b^9*c*d^13*f^2 - 888*A^2*a*b^11*c^7*
d^7*f^2 - 800*A^2*a*b^11*c^5*d^9*f^2 - 336*A^2*a*b^11*c^3*d^11*f^2 - 264*A^
2*a*b^11*c^9*d^5*f^2 - 216*A^2*a^11*b*c^5*d^9*f^2 - 184*A^2*a*b^11*c^11*d^3
*f^2 - 128*A^2*a^3*b^9*c*d^13*f^2 - 112*A^2*a^5*b^7*c*d^13*f^2 - 64*A^2*a^3
*b^9*c^13*d*f^2 + 56*A^2*a^11*b*c^3*d^11*f^2 - 56*A^2*a^7*b^5*c*d^13*f^2 +
32*A^2*a^9*b^3*c*d^13*f^2 + 8*A^2*a^11*b*c^7*d^7*f^2 + 8*A^2*a^5*b^7*c^13*d
*f^2 + 24*C^2*a^11*b*c*d^13*f^2 - 16*C^2*a*b^11*c^13*d*f^2 - 40*B^2*a^11*b*
c*d^13*f^2 + 24*B^2*a*b^11*c^13*d*f^2 + 16*B^2*a*b^11*c*d^13*f^2 - 48*A^2*a
*b^11*c*d^13*f^2 - 40*A^2*a*b^11*c^13*d*f^2 + 24*A^2*a^11*b*c*d^13*f^2 - 6*
A*C*a^12*d^14*f^2 + 2*A*C*b^12*c^14*f^2 + 33*C^2*b^12*c^12*d^2*f^2 - 27*C^2
*b^12*c^10*d^4*f^2 + 3*C^2*b^12*c^8*d^6*f^2 + 117*B^2*b^12*c^10*d^4*f^2 + 1
11*B^2*b^12*c^8*d^6*f^2 + 72*B^2*b^12*c^6*d^8*f^2 + 33*C^2*a^12*c^4*d^10*f^
2 - 27*C^2*a^12*c^2*d^12*f^2 + 24*B^2*b^12*c^4*d^10*f^2 + 4*B^2*b^12*c^2*d^
12*f^2 - 3*B^2*b^12*c^12*d^2*f^2 - C^2*a^12*c^6*d^8*f^2 + 720*A^2*b^12*c^6*
d^8*f^2 + 552*A^2*b^12*c^4*d^10*f^2 + 471*A^2*b^12*c^8*d^6*f^2 + 216*A^2*b^
12*c^2*d^12*f^2 + 93*A^2*b^12*c^10*d^4*f^2 + 33*B^2*a^12*c^2*d^12*f^2 + 33*
A^2*b^12*c^12*d^2*f^2 + 31*C^2*a^8*b^4*d^14*f^2 - 27*B^2*a^12*c^4*d^10*f^2
+ 20*C^2*a^6*b^6*d^14*f^2 + 4*C^2*a^4*b^8*d^14*f^2 + 3*B^2*a^12*c^6*d^8*f^2
 + 2*C^2*a^10*b^2*d^14*f^2 + 80*B^2*a^6*b^6*d^14*f^2 + 64*B^2*a^4*b^8*d^14*
f^2 + 33*A^2*a^12*c^4*d^10*f^2 + 31*B^2*a^8*b^4*d^14*f^2 - 27*A^2*a^12*c^2*
d^12*f^2 + 16*B^2*a^2*b^10*d^14*f^2 + 14*C^2*a^2*b^10*c^14*f^2 + 14*B^2*a^1
0*b^2*d^14*f^2 - C^2*a^4*b^8*c^14*f^2 - A^2*a^12*c^6*d^8*f^2 + 120*A^2*a^2*
b^10*d^14*f^2 + 112*A^2*a^4*b^8*d^14*f^2 - 17*A^2*a^8*b^4*d^14*f^2 - 10*B^2
*a^2*b^10*c^14*f^2 - 10*A^2*a^10*b^2*d^14*f^2 + 8*A^2*a^6*b^6*d^14*f^2 + 3*
B^2*a^4*b^8*c^14*f^2 + 14*A^2*a^2*b^10*c^14*f^2 - A^2*a^4*b^8*c^14*f^2 + 3*
C^2*a^12*d^14*f^2 - C^2*b^12*c^14*f^2 + 36*A^2*b^12*d^14*f^2 + 3*B^2*b^12*c
^14*f^2 - B^2*a^12*d^14*f^2 + 3*A^2*a^12*d^14*f^2 - A^2*b^12*c^14*f^2 - 44*
A*B*C*a*b^9*c^10*d*f + 3816*A*B*C*a^5*b^5*c^4*d^7*f + 2920*A*B*C*a^2*b^8*c^
5*d^6*f - 2736*A*B*C*a^3*b^7*c^6*d^5*f - 2672*A*B*C*a^4*b^6*c^3*d^8*f + 199
6*A*B*C*a^4*b^6*c^7*d^4*f - 1412*A*B*C*a^6*b^4*c^5*d^6*f + 1120*A*B*C*a^3*b
^7*c^2*d^9*f + 1080*A*B*C*a^2*b^8*c^7*d^4*f + 1040*A*B*C*a^5*b^5*c^2*d^9*f
+ 684*A*B*C*a^4*b^6*c^5*d^6*f + 592*A*B*C*a^3*b^7*c^4*d^7*f - 560*A*B*C*a^7
*b^3*c^2*d^9*f - 448*A*B*C*a^2*b^8*c^3*d^8*f - 400*A*B*C*a^5*b^5*c^8*d^3*f
- 398*A*B*C*a^2*b^8*c^9*d^2*f - 312*A*B*C*a^6*b^4*c^3*d^8*f + 166*A*B*C*a^8
*b^2*c^3*d^8*f + 136*A*B*C*a^5*b^5*c^6*d^5*f + 128*A*B*C*a^7*b^3*c^6*d^5*f
- 100*A*B*C*a^6*b^4*c^7*d^4*f + 64*A*B*C*a^7*b^3*c^4*d^7*f - 64*A*B*C*a^4*b
^6*c^9*d^2*f - 32*A*B*C*a^3*b^7*c^8*d^3*f - 16*A*B*C*a^8*b^2*c^5*d^6*f - 13
12*A*B*C*a*b^9*c^4*d^7*f + 996*A*B*C*a*b^9*c^8*d^3*f + 728*A*B*C*a^6*b^4*c*
d^10*f - 624*A*B*C*a*b^9*c^6*d^5*f - 584*A*B*C*a^2*b^8*c*d^10*f - 512*A*B*C

*a^4*b^6*c*d^10*f - 320*A*B*C*a*b^9*c^2*d^9*f - 98*A*B*C*a^8*b^2*c*d^10*f +
 36*A*B*C*a^9*b*c^2*d^9*f + 32*A*B*C*a^3*b^7*c^10*d*f - 16*A*B*C*a^9*b*c^4*
d^7*f + 46*B*C^2*a*b^9*c^10*d*f - 16*B^2*C*a*b^9*c*d^10*f - 2*B^2*C*a^9*b*c
*d^10*f + 312*A^2*C*a*b^9*c*d^10*f - 48*A*C^2*a*b^9*c*d^10*f - 6*A^2*C*a^9*
b*c*d^10*f + 6*A*C^2*a^9*b*c*d^10*f + 208*A*B^2*a*b^9*c*d^10*f - 2*A^2*B*a*
b^9*c^10*d*f + 2*A*B^2*a^9*b*c*d^10*f - 480*A*B*C*b^10*c^7*d^4*f + 78*A*B*C
*b^10*c^9*d^2*f - 64*A*B*C*b^10*c^5*d^6*f + 2*A*B*C*a^10*c^3*d^8*f - 224*A*
B*C*a^5*b^5*d^11*f + 80*A*B*C*a^7*b^3*d^11*f - 32*A*B*C*a^3*b^7*d^11*f + 2*
A*B*C*a^2*b^8*c^11*f - 1692*B*C^2*a^5*b^5*c^4*d^7*f - 1500*B^2*C*a^5*b^5*c^
5*d^6*f - 1464*B^2*C*a^3*b^7*c^5*d^6*f + 1426*B*C^2*a^6*b^4*c^5*d^6*f - 115
8*B^2*C*a^6*b^4*c^4*d^7*f + 1152*B*C^2*a^3*b^7*c^6*d^5*f + 1026*B^2*C*a^4*b
^6*c^6*d^5*f - 974*B*C^2*a^4*b^6*c^7*d^4*f + 960*B^2*C*a^5*b^5*c^3*d^8*f -
884*B*C^2*a^2*b^8*c^5*d^6*f - 764*B^2*C*a^5*b^5*c^7*d^4*f + 752*B^2*C*a^2*b
^8*c^4*d^7*f - 752*B*C^2*a^3*b^7*c^4*d^7*f + 738*B^2*C*a^4*b^6*c^4*d^7*f -
688*B^2*C*a^6*b^4*c^2*d^9*f - 675*B^2*C*a^2*b^8*c^8*d^3*f + 560*B*C^2*a^5*b
^5*c^8*d^3*f + 496*B*C^2*a^7*b^3*c^2*d^9*f + 496*B*C^2*a^4*b^6*c^3*d^8*f -
468*B*C^2*a^2*b^8*c^7*d^4*f + 456*B^2*C*a^7*b^3*c^3*d^8*f - 452*B^2*C*a^4*b
^6*c^8*d^3*f - 416*B*C^2*a^3*b^7*c^2*d^9*f + 378*B*C^2*a^4*b^6*c^5*d^6*f +
376*B*C^2*a^3*b^7*c^8*d^3*f - 360*B^2*C*a^2*b^8*c^6*d^5*f + 355*B*C^2*a^2*b
^8*c^9*d^2*f + 346*B^2*C*a^6*b^4*c^6*d^5*f - 320*B^2*C*a^4*b^6*c^2*d^9*f +
268*B^2*C*a^2*b^8*c^2*d^9*f + 216*B^2*C*a^3*b^7*c^7*d^4*f - 203*B*C^2*a^8*b
^2*c^3*d^8*f - 184*B*C^2*a^7*b^3*c^6*d^5*f + 170*B*C^2*a^6*b^4*c^7*d^4*f +
160*B^2*C*a^7*b^3*c^5*d^6*f - 160*B*C^2*a^5*b^5*c^2*d^9*f - 140*B^2*C*a^8*b
^2*c^4*d^7*f - 136*B*C^2*a^2*b^8*c^3*d^8*f + 112*B^2*C*a^3*b^7*c^9*d^2*f +
91*B^2*C*a^8*b^2*c^2*d^9*f + 88*B*C^2*a^7*b^3*c^4*d^7*f + 72*B^2*C*a^6*b^4*
c^8*d^3*f - 64*B^2*C*a^3*b^7*c^3*d^8*f - 60*B*C^2*a^6*b^4*c^3*d^8*f + 56*B*
C^2*a^4*b^6*c^9*d^2*f + 52*B*C^2*a^5*b^5*c^6*d^5*f - 48*B^2*C*a^7*b^3*c^7*d
^4*f + 48*B^2*C*a^5*b^5*c^9*d^2*f + 44*B*C^2*a^8*b^2*c^5*d^6*f - 36*B*C^2*a
^6*b^4*c^9*d^2*f + 12*B^2*C*a^8*b^2*c^6*d^5*f - 2958*A^2*C*a^4*b^6*c^4*d^7*
f - 1932*A^2*C*a^2*b^8*c^4*d^7*f + 1848*A^2*C*a^3*b^7*c^5*d^6*f + 1728*A^2*
C*a^3*b^7*c^3*d^8*f + 1524*A^2*C*a^5*b^5*c^5*d^6*f + 1374*A*C^2*a^4*b^6*c^4
*d^7*f - 1272*A*C^2*a^3*b^7*c^5*d^6*f - 1236*A*C^2*a^5*b^5*c^5*d^6*f + 1116
*A*C^2*a^2*b^8*c^4*d^7*f - 1110*A^2*C*a^4*b^6*c^6*d^5*f + 1038*A*C^2*a^4*b^
6*c^6*d^5*f - 768*A^2*C*a^2*b^8*c^2*d^9*f - 696*A^2*C*a^3*b^7*c^7*d^4*f - 6
66*A*C^2*a^6*b^4*c^4*d^7*f + 564*A^2*C*a^2*b^8*c^6*d^5*f - 564*A*C^2*a^5*b^
5*c^7*d^4*f - 555*A*C^2*a^2*b^8*c^8*d^3*f + 519*A^2*C*a^2*b^8*c^8*d^3*f - 4
80*A*C^2*a^3*b^7*c^3*d^8*f + 456*A*C^2*a^5*b^5*c^3*d^8*f - 420*A*C^2*a^6*b^
4*c^2*d^9*f + 408*A*C^2*a^3*b^7*c^7*d^4*f + 408*A*C^2*a^2*b^8*c^2*d^9*f + 3
48*A^2*C*a^6*b^4*c^2*d^9*f - 348*A*C^2*a^2*b^8*c^6*d^5*f + 342*A*C^2*a^6*b^
4*c^6*d^5*f - 336*A*C^2*a^4*b^6*c^8*d^3*f + 324*A^2*C*a^5*b^5*c^7*d^4*f - 3
12*A^2*C*a^4*b^6*c^2*d^9*f + 264*A^2*C*a^4*b^6*c^8*d^3*f + 240*A*C^2*a^7*b^
3*c^5*d^6*f + 195*A*C^2*a^8*b^2*c^2*d^9*f - 174*A^2*C*a^6*b^4*c^6*d^5*f + 1
44*A*C^2*a^3*b^7*c^9*d^2*f - 123*A^2*C*a^8*b^2*c^2*d^9*f + 120*A*C^2*a^7*b^
3*c^3*d^8*f + 108*A*C^2*a^6*b^4*c^8*d^3*f - 102*A^2*C*a^6*b^4*c^4*d^7*f - 9
6*A^2*C*a^8*b^2*c^4*d^7*f + 72*A^2*C*a^7*b^3*c^3*d^8*f + 72*A*C^2*a^5*b^5*c

^9*d^2*f + 48*A^2*C*a^7*b^3*c^5*d^6*f - 48*A^2*C*a^3*b^7*c^9*d^2*f - 48*A*C
^2*a^4*b^6*c^2*d^9*f - 24*A^2*C*a^5*b^5*c^3*d^8*f - 12*A*C^2*a^8*b^2*c^4*d^
7*f + 2736*A^2*B*a^3*b^7*c^6*d^5*f + 2464*A^2*B*a^4*b^6*c^3*d^8*f - 2298*A*
B^2*a^4*b^6*c^4*d^7*f - 2252*A^2*B*a^2*b^8*c^5*d^6*f - 1692*A^2*B*a^5*b^5*c
^4*d^7*f - 1592*A*B^2*a^2*b^8*c^4*d^7*f - 1338*A*B^2*a^4*b^6*c^6*d^5*f + 13
20*A*B^2*a^3*b^7*c^5*d^6*f + 1212*A*B^2*a^5*b^5*c^5*d^6*f - 1056*A*B^2*a^5*
b^5*c^3*d^8*f + 1024*A^2*B*a^3*b^7*c^4*d^7*f - 1022*A^2*B*a^4*b^6*c^7*d^4*f
 - 880*A^2*B*a^5*b^5*c^2*d^9*f - 846*A^2*B*a^4*b^6*c^5*d^6*f - 840*A*B^2*a^
3*b^7*c^7*d^4*f + 760*A*B^2*a^6*b^4*c^2*d^9*f - 704*A^2*B*a^3*b^7*c^2*d^9*f
 + 688*A*B^2*a^3*b^7*c^3*d^8*f + 660*A^2*B*a^6*b^4*c^3*d^8*f - 612*A^2*B*a^
2*b^8*c^7*d^4*f + 462*A*B^2*a^6*b^4*c^4*d^7*f + 459*A*B^2*a^2*b^8*c^8*d^3*f
 - 412*A*B^2*a^2*b^8*c^2*d^9*f - 408*A*B^2*a^7*b^3*c^3*d^8*f + 388*A^2*B*a^
5*b^5*c^6*d^5*f + 296*A^2*B*a^2*b^8*c^3*d^8*f + 288*A*B^2*a^2*b^8*c^6*d^5*f
 + 284*A*B^2*a^5*b^5*c^7*d^4*f + 236*A*B^2*a^4*b^6*c^8*d^3*f - 226*A*B^2*a^
6*b^4*c^6*d^5*f + 212*A*B^2*a^4*b^6*c^2*d^9*f + 202*A^2*B*a^6*b^4*c^5*d^6*f
 - 152*A^2*B*a^7*b^3*c^4*d^7*f + 88*A^2*B*a^3*b^7*c^8*d^3*f + 79*A^2*B*a^2*
b^8*c^9*d^2*f - 70*A^2*B*a^6*b^4*c^7*d^4*f + 68*A*B^2*a^8*b^2*c^4*d^7*f + 6
4*A^2*B*a^7*b^3*c^2*d^9*f - 64*A*B^2*a^3*b^7*c^9*d^2*f + 56*A^2*B*a^7*b^3*c
^6*d^5*f + 56*A^2*B*a^5*b^5*c^8*d^3*f + 37*A^2*B*a^8*b^2*c^3*d^8*f - 28*A^2
*B*a^8*b^2*c^5*d^6*f - 28*A^2*B*a^4*b^6*c^9*d^2*f + 17*A*B^2*a^8*b^2*c^2*d^
9*f - 16*A*B^2*a^7*b^3*c^5*d^6*f + 24*A*B*C*b^10*c*d^10*f - 6*A*B*C*a^10*c*
d^10*f + 48*A*B*C*a*b^9*d^11*f + 4*A*B*C*a^9*b*d^11*f + 432*B^2*C*a*b^9*c^7
*d^4*f - 376*B*C^2*a^6*b^4*c*d^10*f - 354*B*C^2*a*b^9*c^8*d^3*f + 352*B^2*C
*a^5*b^5*c*d^10*f + 320*B^2*C*a*b^9*c^5*d^6*f + 256*B^2*C*a^3*b^7*c*d^10*f
- 232*B^2*C*a^7*b^3*c*d^10*f - 210*B^2*C*a*b^9*c^9*d^2*f - 152*B*C^2*a^4*b^
6*c*d^10*f + 85*B*C^2*a^8*b^2*c*d^10*f + 72*B^2*C*a*b^9*c^3*d^8*f - 48*B*C^
2*a*b^9*c^6*d^5*f - 40*B*C^2*a^3*b^7*c^10*d*f + 40*B*C^2*a^2*b^8*c*d^10*f +
 37*B^2*C*a^2*b^8*c^10*d*f + 22*B^2*C*a^9*b*c^3*d^8*f - 18*B*C^2*a^9*b*c^2*
d^9*f + 16*B*C^2*a*b^9*c^2*d^9*f - 12*B^2*C*a^4*b^6*c^10*d*f + 8*B*C^2*a^9*
b*c^4*d^7*f + 8*B*C^2*a*b^9*c^4*d^7*f - 984*A^2*C*a*b^9*c^7*d^4*f + 672*A^2
*C*a*b^9*c^3*d^8*f + 552*A*C^2*a*b^9*c^7*d^4*f - 504*A^2*C*a^5*b^5*c*d^10*f
 - 408*A^2*C*a*b^9*c^5*d^6*f + 408*A*C^2*a*b^9*c^5*d^6*f + 336*A*C^2*a^5*b^
5*c*d^10*f - 216*A*C^2*a^7*b^3*c*d^10*f + 192*A*C^2*a^3*b^7*c*d^10*f - 162*
A*C^2*a*b^9*c^9*d^2*f + 120*A^2*C*a^7*b^3*c*d^10*f + 96*A^2*C*a^3*b^7*c*d^1
0*f + 90*A^2*C*a*b^9*c^9*d^2*f + 66*A^2*C*a^9*b*c^3*d^8*f - 66*A*C^2*a^9*b*
c^3*d^8*f + 57*A*C^2*a^2*b^8*c^10*d*f - 48*A*C^2*a*b^9*c^3*d^8*f - 9*A^2*C*
a^2*b^8*c^10*d*f + 1736*A^2*B*a*b^9*c^4*d^7*f + 1248*A^2*B*a*b^9*c^6*d^5*f
- 1008*A*B^2*a*b^9*c^7*d^4*f + 772*A^2*B*a^4*b^6*c*d^10*f - 688*A*B^2*a^5*b
^5*c*d^10*f - 608*A*B^2*a*b^9*c^5*d^6*f + 436*A^2*B*a^2*b^8*c*d^10*f - 426*
A^2*B*a*b^9*c^8*d^3*f + 312*A*B^2*a*b^9*c^3*d^8*f + 304*A^2*B*a*b^9*c^2*d^9
*f - 244*A^2*B*a^6*b^4*c*d^10*f - 160*A*B^2*a^3*b^7*c*d^10*f + 114*A*B^2*a*
b^9*c^9*d^2*f + 88*A*B^2*a^7*b^3*c*d^10*f - 22*A*B^2*a^9*b*c^3*d^8*f - 18*A
^2*B*a^9*b*c^2*d^9*f + 13*A^2*B*a^8*b^2*c*d^10*f - 13*A*B^2*a^2*b^8*c^10*d*
f + 8*A^2*B*a^9*b*c^4*d^7*f + 8*A^2*B*a^3*b^7*c^10*d*f + 111*B^2*C*b^10*c^8
*d^3*f - 39*B*C^2*b^10*c^9*d^2*f + 24*B*C^2*b^10*c^7*d^4*f - 4*B^2*C*b^10*c

^2*d^9*f - 4*B*C^2*b^10*c^5*d^6*f + 432*A^2*C*b^10*c^6*d^5*f + 192*A^2*C*b^
10*c^4*d^7*f - 111*A^2*C*b^10*c^8*d^3*f + 111*A*C^2*b^10*c^8*d^3*f - 72*A*C
^2*b^10*c^6*d^5*f + 12*A*C^2*b^10*c^4*d^7*f - 3*B^2*C*a^10*c^2*d^9*f - B*C^
2*a^10*c^3*d^8*f + 456*A^2*B*b^10*c^7*d^4*f - 288*A^2*B*b^10*c^3*d^8*f + 25
2*A*B^2*b^10*c^6*d^5*f + 192*A*B^2*b^10*c^4*d^7*f - 183*A*B^2*b^10*c^8*d^3*
f - 148*A^2*B*b^10*c^5*d^6*f + 112*B^2*C*a^6*b^4*d^11*f + 76*A*B^2*b^10*c^2
*d^9*f - 64*B*C^2*a^7*b^3*d^11*f + 16*B^2*C*a^4*b^6*d^11*f - 16*B^2*C*a^2*b
^8*d^11*f + 16*B*C^2*a^5*b^5*d^11*f + 16*B*C^2*a^3*b^7*d^11*f - 9*A^2*C*a^1
0*c^2*d^9*f + 9*A*C^2*a^10*c^2*d^9*f - 3*A^2*B*b^10*c^9*d^2*f - B^2*C*a^8*b
^2*d^11*f + 96*A^2*C*a^4*b^6*d^11*f - 84*A^2*C*a^6*b^4*d^11*f + 72*A*C^2*a^
6*b^4*d^11*f - 24*A*C^2*a^4*b^6*d^11*f - 24*A*C^2*a^2*b^8*d^11*f - 21*A*C^2
*a^8*b^2*d^11*f + 12*A^2*C*a^2*b^8*d^11*f + 9*A^2*C*a^8*b^2*d^11*f + 3*A*B^
2*a^10*c^2*d^9*f - A^2*B*a^10*c^3*d^8*f - B*C^2*a^2*b^8*c^11*f + 176*A*B^2*
a^4*b^6*d^11*f + 136*A^2*B*a^5*b^5*d^11*f - 128*A^2*B*a^3*b^7*d^11*f + 112*
A*B^2*a^2*b^8*d^11*f - 64*A*B^2*a^6*b^4*d^11*f - 16*A^2*B*a^7*b^3*d^11*f -
A^2*B*a^2*b^8*c^11*f - 2*C^3*a^9*b*c*d^10*f - 2*B^3*a*b^9*c^10*d*f - 264*A^
3*a*b^9*c*d^10*f + 2*A^3*a^9*b*c*d^10*f - 9*B^2*C*b^10*c^10*d*f + 9*A^2*C*b
^10*c^10*d*f - 9*A*C^2*b^10*c^10*d*f + 3*B*C^2*a^10*c*d^10*f - 132*A^2*B*b^
10*c*d^10*f - 3*A*B^2*b^10*c^10*d*f - 2*B*C^2*a^9*b*d^11*f + 3*A^2*B*a^10*c
*d^10*f - 2*B^2*C*a*b^9*c^11*f - 120*A^2*B*a*b^9*d^11*f - 6*A^2*C*a*b^9*c^1
1*f + 6*A*C^2*a*b^9*c^11*f - 2*A^2*B*a^9*b*d^11*f + 2*A*B^2*a*b^9*c^11*f +
520*C^3*a^3*b^7*c^5*d^6*f + 460*C^3*a^5*b^5*c^5*d^6*f - 418*C^3*a^4*b^6*c^6
*d^5*f + 406*C^3*a^6*b^4*c^4*d^7*f + 268*C^3*a^5*b^5*c^7*d^4*f - 266*C^3*a^
6*b^4*c^6*d^5*f + 233*C^3*a^2*b^8*c^8*d^3*f - 176*C^3*a^7*b^3*c^5*d^6*f + 1
64*C^3*a^6*b^4*c^2*d^9*f + 140*C^3*a^2*b^8*c^6*d^5*f + 136*C^3*a^4*b^6*c^2*
d^9*f - 128*C^3*a^3*b^7*c^9*d^2*f + 128*C^3*a^3*b^7*c^3*d^8*f - 108*C^3*a^6
*b^4*c^8*d^3*f - 104*C^3*a^7*b^3*c^3*d^8*f - 104*C^3*a^5*b^5*c^3*d^8*f + 10
0*C^3*a^4*b^6*c^8*d^3*f - 89*C^3*a^8*b^2*c^2*d^9*f - 72*C^3*a^5*b^5*c^9*d^2
*f + 40*C^3*a^8*b^2*c^4*d^7*f - 40*C^3*a^3*b^7*c^7*d^4*f - 28*C^3*a^2*b^8*c
^4*d^7*f - 16*C^3*a^2*b^8*c^2*d^9*f - 2*C^3*a^4*b^6*c^4*d^7*f + 828*B^3*a^5
*b^5*c^4*d^7*f + 408*B^3*a^2*b^8*c^5*d^6*f + 390*B^3*a^4*b^6*c^7*d^4*f - 37
2*B^3*a^4*b^6*c^3*d^8*f - 336*B^3*a^3*b^7*c^6*d^5*f - 314*B^3*a^6*b^4*c^5*d
^6*f + 288*B^3*a^3*b^7*c^4*d^7*f + 216*B^3*a^2*b^8*c^7*d^4*f - 176*B^3*a^7*
b^3*c^2*d^9*f + 128*B^3*a^3*b^7*c^2*d^9*f + 108*B^3*a^5*b^5*c^6*d^5*f + 88*
B^3*a^7*b^3*c^4*d^7*f + 72*B^3*a^5*b^5*c^2*d^9*f - 68*B^3*a^2*b^8*c^3*d^8*f
 - 65*B^3*a^2*b^8*c^9*d^2*f - 56*B^3*a^5*b^5*c^8*d^3*f + 40*B^3*a^7*b^3*c^6
*d^5*f + 37*B^3*a^8*b^2*c^3*d^8*f + 30*B^3*a^4*b^6*c^5*d^6*f - 28*B^3*a^8*b
^2*c^5*d^6*f + 24*B^3*a^3*b^7*c^8*d^3*f - 4*B^3*a^4*b^6*c^9*d^2*f - 2*B^3*a
^6*b^4*c^7*d^4*f + 1586*A^3*a^4*b^6*c^4*d^7*f - 1376*A^3*a^3*b^7*c^3*d^8*f
 - 1096*A^3*a^3*b^7*c^5*d^6*f + 844*A^3*a^2*b^8*c^4*d^7*f - 748*A^3*a^5*b^5*
c^5*d^6*f + 490*A^3*a^4*b^6*c^6*d^5*f + 376*A^3*a^2*b^8*c^2*d^9*f + 362*A^3
*a^6*b^4*c^4*d^7*f - 356*A^3*a^2*b^8*c^6*d^5*f - 328*A^3*a^5*b^5*c^3*d^8*f
 + 328*A^3*a^3*b^7*c^7*d^4*f + 224*A^3*a^4*b^6*c^2*d^9*f - 197*A^3*a^2*b^8*c
^8*d^3*f - 112*A^3*a^7*b^3*c^5*d^6*f + 98*A^3*a^6*b^4*c^6*d^5*f - 92*A^3*a^
6*b^4*c^2*d^9*f - 88*A^3*a^7*b^3*c^3*d^8*f + 68*A^3*a^8*b^2*c^4*d^7*f + 32*

A^3*a^3*b^7*c^9*d^2*f - 28*A^3*a^5*b^5*c^7*d^4*f - 28*A^3*a^4*b^6*c^8*d^3*f
 + 17*A^3*a^8*b^2*c^2*d^9*f + 104*C^3*a^7*b^3*c*d^10*f + 54*C^3*a*b^9*c^9*d
^2*f - 40*C^3*a*b^9*c^7*d^4*f - 35*C^3*a^2*b^8*c^10*d*f + 22*C^3*a^9*b*c^3*
d^8*f + 16*C^3*a^5*b^5*c*d^10*f - 16*C^3*a^3*b^7*c*d^10*f + 8*C^3*a*b^9*c^5
*d^6*f - 2*A*B*C*b^10*c^11*f + 198*B^3*a*b^9*c^8*d^3*f + 192*B^3*a^6*b^4*c*
d^10*f - 128*B^3*a*b^9*c^4*d^7*f - 80*B^3*a^2*b^8*c*d^10*f - 56*B^3*a*b^9*c
^2*d^9*f - 24*B^3*a*b^9*c^6*d^5*f - 18*B^3*a^9*b*c^2*d^9*f - 16*B^3*a^4*b^6
*c*d^10*f + 13*B^3*a^8*b^2*c*d^10*f + 8*B^3*a^9*b*c^4*d^7*f + 8*B^3*a^3*b^7
*c^10*d*f - 624*A^3*a*b^9*c^3*d^8*f + 472*A^3*a*b^9*c^7*d^4*f - 272*A^3*a^3
*b^7*c*d^10*f + 152*A^3*a^5*b^5*c*d^10*f - 22*A^3*a^9*b*c^3*d^8*f + 18*A^3*
a*b^9*c^9*d^2*f - 13*A^3*a^2*b^8*c^10*d*f - 8*A^3*a^7*b^3*c*d^10*f - 8*A^3*
a*b^9*c^5*d^6*f + A*B^2*a^8*b^2*d^11*f - C^3*b^10*c^8*d^3*f - 60*B^3*b^10*c
^7*d^4*f - 32*B^3*b^10*c^5*d^6*f + 21*B^3*b^10*c^9*d^2*f - 12*B^3*b^10*c^3*
d^8*f - 3*C^3*a^10*c^2*d^9*f - 360*A^3*b^10*c^6*d^5*f - 204*A^3*b^10*c^4*d^
7*f + 11*C^3*a^8*b^2*d^11*f - 8*C^3*a^6*b^4*d^11*f - 4*C^3*a^4*b^6*d^11*f -
 B^3*a^10*c^3*d^8*f - 64*B^3*a^5*b^5*d^11*f - 32*B^3*a^3*b^7*d^11*f + 3*A^3
*a^10*c^2*d^9*f - 68*A^3*a^4*b^6*d^11*f + 20*A^3*a^6*b^4*d^11*f + 12*A^3*a^
2*b^8*d^11*f - B^3*a^2*b^8*c^11*f + 3*C^3*b^10*c^10*d*f + 3*B^3*a^10*c*d^10
*f - 3*A^3*b^10*c^10*d*f - 2*C^3*a*b^9*c^11*f - 2*B^3*a^9*b*d^11*f + 2*A^3*
a*b^9*c^11*f - 36*A^2*C*b^10*d^11*f + 3*A^2*C*a^10*d^11*f - 3*A*C^2*a^10*d^
11*f - A*B^2*a^10*d^11*f + 36*A^3*b^10*d^11*f - A^3*a^10*d^11*f + A^3*b^10*
c^8*d^3*f + A^3*a^8*b^2*d^11*f + B^2*C*a^10*d^11*f + B*C^2*b^10*c^11*f + A^
2*B*b^10*c^11*f + C^3*a^10*d^11*f + B^3*b^10*c^11*f - 6*A*B^2*C*a*b^7*c^7*d
 + 4*A*B^2*C*a*b^7*c*d^7 + 168*A^2*B*C*a^3*b^5*c^2*d^6 + 144*A*B*C^2*a^4*b^
4*c^3*d^5 - 129*A^2*B*C*a^4*b^4*c^3*d^5 - 96*A*B*C^2*a^3*b^5*c^2*d^6 + 84*A
*B*C^2*a^2*b^6*c^3*d^5 + 72*A^2*B*C*a^3*b^5*c^4*d^4 - 72*A^2*B*C*a^2*b^6*c^
3*d^5 + 64*A*B^2*C*a^4*b^4*c^4*d^4 - 60*A*B*C^2*a^3*b^5*c^4*d^4 + 57*A^2*B*
C*a^2*b^6*c^5*d^3 - 56*A*B^2*C*a^3*b^5*c^5*d^3 - 39*A*B^2*C*a^4*b^4*c^2*d^6
 - 38*A*B^2*C*a^5*b^3*c^3*d^5 + 36*A*B^2*C*a^3*b^5*c^3*d^5 + 36*A*B*C^2*a^4
*b^4*c^5*d^3 - 30*A*B*C^2*a^2*b^6*c^5*d^3 + 27*A*B^2*C*a^2*b^6*c^6*d^2 - 24
*A*B^2*C*a^2*b^6*c^2*d^6 - 24*A*B*C^2*a^5*b^3*c^4*d^4 + 24*A*B*C^2*a^3*b^5*
c^6*d^2 + 18*A^2*B*C*a^5*b^3*c^2*d^6 - 18*A^2*B*C*a^4*b^4*c^5*d^3 - 15*A*B^
2*C*a^2*b^6*c^4*d^4 + 12*A^2*B*C*a^5*b^3*c^4*d^4 - 12*A^2*B*C*a^3*b^5*c^6*d
^2 + 9*A*B^2*C*a^6*b^2*c^2*d^6 + 6*A*B*C^2*a^6*b^2*c^3*d^5 - 3*A^2*B*C*a^6*
b^2*c^3*d^5 + 60*A^2*B*C*a*b^7*c^2*d^6 - 51*A^2*B*C*a^4*b^4*c*d^7 + 48*A*B*
C^2*a*b^7*c^6*d^2 - 42*A^2*B*C*a^2*b^6*c*d^7 - 42*A^2*B*C*a*b^7*c^6*d^2 + 3
6*A*B*C^2*a^4*b^4*c*d^7 + 36*A*B*C^2*a^2*b^6*c*d^7 + 36*A*B*C^2*a*b^7*c^4*d
^4 - 30*A^2*B*C*a*b^7*c^4*d^4 + 24*A*B^2*C*a*b^7*c^3*d^5 - 24*A*B*C^2*a*b^7
*c^2*d^6 + 18*A*B^2*C*a^5*b^3*c*d^7 - 18*A*B*C^2*a^6*b^2*c*d^7 + 12*A*B^2*C
*a^3*b^5*c*d^7 + 9*A^2*B*C*a^6*b^2*c*d^7 + 6*A*B^2*C*a*b^7*c^5*d^3 - 6*A*B*
C^2*a^2*b^6*c^7*d + 3*A^2*B*C*a^2*b^6*c^7*d - 18*B^3*C*a*b^7*c^6*d^2 - 18*B
*C^3*a*b^7*c^6*d^2 - 14*B^3*C*a*b^7*c^4*d^4 - 14*B*C^3*a*b^7*c^4*d^4 - 10*B
^3*C*a^2*b^6*c*d^7 - 10*B*C^3*a^2*b^6*c*d^7 + 9*B^3*C*a^6*b^2*c*d^7 + 9*B*C
^3*a^6*b^2*c*d^7 - 7*B^3*C*a^4*b^4*c*d^7 - 7*B*C^3*a^4*b^4*c*d^7 + 6*B^2*C^
2*a*b^7*c^7*d - 4*B^3*C*a*b^7*c^2*d^6 + 4*B^2*C^2*a*b^7*c*d^7 - 4*B*C^3*a*b

^7*c^2*d^6 + 3*B^3*C*a^2*b^6*c^7*d + 3*B*C^3*a^2*b^6*c^7*d + 144*A^3*C*a*b^7*c^3*d^5 + 62*A^3*C*a*b^7*c^5*d^3 + 48*A*C^3*a*b^7*c^3*d^5 - 36*A^2*C^2*a*b^7*c*d^7 + 26*A*C^3*a*b^7*c^5*d^3 + 20*A^3*C*a^3*b^5*c*d^7 + 18*A^2*C^2*a*b^7*c^7*d - 18*A*C^3*a^5*b^3*c*d^7 - 6*A^3*C*a^5*b^3*c*d^7 - 4*A*C^3*a^3*b^5*c*d^7 - 32*A^3*B*a*b^7*c^2*d^6 - 32*A*B^3*a*b^7*c^2*d^6 + 22*A^3*B*a^4*b^4*c*d^7 + 22*A*B^3*a^4*b^4*c*d^7 + 16*A^3*B*a^2*b^6*c*d^7 + 16*A*B^3*a^2*b^6*c*d^7 + 12*A^3*B*a*b^7*c^6*d^2 + 12*A*B^3*a*b^7*c^6*d^2 + 8*A^3*B*a*b^7*c^4*d^4 - 8*A^2*B^2*a*b^7*c*d^7 + 8*A*B^3*a*b^7*c^4*d^4 + 57*A^2*B*C*b^8*c^5*d^3 + 36*A^2*B*C*b^8*c^3*d^5 - 30*A*B*C^2*b^8*c^5*d^3 - 18*A*B*C^2*b^8*c^3*d^5 - 9*A*B^2*C*b^8*c^4*d^4 - 3*A*B^2*C*b^8*c^6*d^2 - 2*A*B^2*C*b^8*c^2*d^6 + 36*A^2*B*C*a^3*b^5*d^8 + 24*A*B*C^2*a^5*b^3*d^8 - 18*A^2*B*C*a^5*b^3*d^8 - 12*A*B*C^2*a^3*b^5*d^8 - 3*A*B^2*C*a^6*b^2*d^8 - 3*A*B^2*C*a^4*b^4*d^8 - 2*A*B^2*C*a^2*b^6*d^8 + 34*B^2*C^2*a^5*b^3*c^3*d^5 + 28*B^2*C^2*a^3*b^5*c^5*d^3 + 24*B^2*C^2*a^4*b^4*c^2*d^6 - 20*B^2*C^2*a^4*b^4*c^4*d^4 + 12*B^2*C^2*a^3*b^5*c^3*d^5 + 12*B^2*C^2*a^2*b^6*c^2*d^6 - 9*B^2*C^2*a^6*b^2*c^2*d^6 + 9*B^2*C^2*a^4*b^4*c^6*d^2 + 9*B^2*C^2*a^2*b^6*c^4*d^4 - 3*B^2*C^2*a^2*b^6*c^6*d^2 + 159*A^2*C^2*a^2*b^6*c^4*d^4 - 156*A^2*C^2*a^3*b^5*c^3*d^5 + 90*A^2*C^2*a^5*b^3*c^3*d^5 + 78*A^2*C^2*a^2*b^6*c^2*d^6 - 63*A^2*C^2*a^4*b^4*c^4*d^4 - 27*A^2*C^2*a^6*b^2*c^2*d^6 - 27*A^2*C^2*a^2*b^6*c^6*d^2 - 18*A^2*C^2*a^4*b^4*c^2*d^6 + 9*A^2*C^2*a^4*b^4*c^6*d^2 + 66*A^2*B^2*a^2*b^6*c^2*d^6 + 60*A^2*B^2*a^2*b^6*c^4*d^4 - 48*A^2*B^2*a^3*b^5*c^3*d^5 + 42*A^2*B^2*a^4*b^4*c^2*d^6 + 28*A^2*B^2*a^3*b^5*c^5*d^3 - 17*A^2*B^2*a^4*b^4*c^4*d^4 - 6*A^2*B^2*a^2*b^6*c^6*d^2 + 4*A^2*B^2*a^5*b^3*c^3*d^5 + 36*A^3*C*a*b^7*c*d^7 - 18*A*C^3*a*b^7*c^7*d + 12*A*C^3*a*b^7*c*d^7 - 6*A^3*C*a*b^7*c^7*d + 12*A^2*B*C*b^8*c*d^7 + 6*A*B*C^2*b^8*c^7*d - 6*A*B*C^2*b^8*c*d^7 - 3*A^2*B*C*b^8*c^7*d + 24*A^2*B*C*a*b^7*d^8 - 12*A*B*C^2*a*b^7*d^8 - 53*B^3*C*a^4*b^4*c^3*d^5 - 53*B*C^3*a^4*b^4*c^3*d^5 - 32*B^3*C*a^2*b^6*c^3*d^5 - 32*B*C^3*a^2*b^6*c^3*d^5 - 18*B^3*C*a^4*b^4*c^5*d^3 - 18*B*C^3*a^4*b^4*c^5*d^3 + 16*B^3*C*a^3*b^5*c^4*d^4 + 16*B*C^3*a^3*b^5*c^4*d^4 + 12*B^3*C*a^5*b^3*c^4*d^4 - 12*B^3*C*a^3*b^5*c^6*d^2 + 12*B^2*C^2*a*b^7*c^3*d^5 + 12*B*C^3*a^5*b^3*c^4*d^4 - 12*B*C^3*a^3*b^5*c^6*d^2 + 8*B^3*C*a^3*b^5*c^2*d^6 + 8*B*C^3*a^3*b^5*c^2*d^6 - 6*B^3*C*a^5*b^3*c^2*d^6 - 6*B^2*C^2*a^5*b^3*c*d^7 + 6*B^2*C^2*a*b^7*c^5*d^3 - 6*B*C^3*a^5*b^3*c^2*d^6 - 3*B^3*C*a^6*b^2*c^3*d^5 - 3*B*C^3*a^6*b^2*c^3*d^5 - 175*A^3*C*a^2*b^6*c^4*d^4 + 164*A^3*C*a^3*b^5*c^3*d^5 - 144*A^2*C^2*a*b^7*c^3*d^5 - 124*A^3*C*a^2*b^6*c^2*d^6 - 90*A*C^3*a^5*b^3*c^3*d^5 - 73*A*C^3*a^2*b^6*c^4*d^4 - 66*A^2*C^2*a*b^7*c^5*d^3 + 44*A*C^3*a^3*b^5*c^3*d^5 + 36*A*C^3*a^4*b^4*c^4*d^4 - 30*A^3*C*a^5*b^3*c^3*d^5 + 30*A^3*C*a^4*b^4*c^4*d^4 + 27*A*C^3*a^6*b^2*c^2*d^6 + 21*A*C^3*a^4*b^4*c^2*d^6 + 18*A^2*C^2*a^5*b^3*c*d^7 - 18*A*C^3*a^4*b^4*c^6*d^2 - 16*A*C^3*a^2*b^6*c^2*d^6 - 15*A^3*C*a^4*b^4*c^2*d^6 + 15*A^3*C*a^2*b^6*c^6*d^2 - 12*A^2*C^2*a^3*b^5*c*d^7 + 9*A^3*C*a^6*b^2*c^2*d^6 + 9*A*C^3*a^2*b^6*c^6*d^2 - 80*A^3*B*a^3*b^5*c^2*d^6 - 80*A*B^3*a^3*b^5*c^2*d^6 + 38*A^3*B*a^4*b^4*c^3*d^5 + 38*A*B^3*a^4*b^4*c^3*d^5 - 36*A^2*B^2*a*b^7*c^3*d^5 - 28*A^3*B*a^3*b^5*c^4*d^4 - 28*A^3*B*a^2*b^6*c^5*d^3 - 28*A*B^3*a^3*b^5*c^4*d^4 - 28*A*B^3*a^2*b^6*c^5*d^3 + 20*A^3*B*a^2*b^6*c^3*d^5 + 20*A*B^3*a^2*b^6*c^3*d^5 - 12*A^3*B*a^5*b^3*c^2

```
*d^6 - 12*A^2*B^2*a^5*b^3*c*d^7 - 12*A^2*B^2*a^3*b^5*c*d^7 - 12*A^2*B^2*a*b
^7*c^5*d^3 - 12*A*B^3*a^5*b^3*c^2*d^6 + 6*B^2*C^2*b^8*c^6*d^2 + 3*B^2*C^2*b
^8*c^4*d^4 + 36*A^2*C^2*b^8*c^4*d^4 + 27*A^2*C^2*b^8*c^2*d^6 - 18*A^2*C^2*b
^8*c^6*d^2 + 33*A^2*B^2*b^8*c^4*d^4 + 28*A^2*B^2*b^8*c^2*d^6 + 9*B^2*C^2*a^
4*b^4*d^8 + 6*A^2*B^2*b^8*c^6*d^2 + 4*B^2*C^2*a^2*b^6*d^8 + 3*B^2*C^2*a^6*b
^2*d^8 - 30*A^2*C^2*a^4*b^4*d^8 + 9*A^2*C^2*a^6*b^2*d^8 + 16*A^2*B^2*a^2*b^
6*d^8 + 3*A^2*B^2*a^4*b^4*d^8 + 6*C^4*a^5*b^3*c*d^7 + 4*C^4*a^3*b^5*c*d^7 -
 2*C^4*a*b^7*c^5*d^3 - 12*B^4*a^5*b^3*c*d^7 + 12*B^4*a*b^7*c^3*d^5 + 8*B^4*
a*b^7*c^5*d^3 - 4*B^4*a^3*b^5*c*d^7 - 48*A^4*a*b^7*c^3*d^5 - 20*A^4*a*b^7*c
^5*d^3 - 8*A^4*a^3*b^5*c*d^7 - 63*A^3*C*b^8*c^4*d^4 - 54*A^3*C*b^8*c^2*d^6
+ 9*A^3*C*b^8*c^6*d^2 + 9*A*C^3*b^8*c^6*d^2 - 3*A*C^3*b^8*c^4*d^4 - 28*A^3*
B*b^8*c^5*d^3 - 28*A*B^3*b^8*c^5*d^3 - 18*A^3*B*b^8*c^3*d^5 - 18*A*B^3*b^8*
c^3*d^5 - 10*B^3*C*a^5*b^3*d^8 - 10*B*C^3*a^5*b^3*d^8 - 4*B^3*C*a^3*b^5*d^8
 - 4*B*C^3*a^3*b^5*d^8 + 23*A^3*C*a^4*b^4*d^8 - 18*A^3*C*a^2*b^6*d^8 + 11*A
*C^3*a^4*b^4*d^8 - 9*A*C^3*a^6*b^2*d^8 + 6*A*C^3*a^2*b^6*d^8 - 3*A^3*C*a^6*
b^2*d^8 - 20*A^3*B*a^3*b^5*d^8 - 20*A*B^3*a^3*b^5*d^8 + 4*A^3*B*a^5*b^3*d^8
 + 4*A*B^3*a^5*b^3*d^8 + B^3*C*a^2*b^6*c^5*d^3 + B*C^3*a^2*b^6*c^5*d^3 + 6*
C^4*a*b^7*c^7*d + 4*B^4*a*b^7*c*d^7 - 12*A^4*a*b^7*c*d^7 - 3*B^3*C*b^8*c^7*
d - 3*B*C^3*b^8*c^7*d - 6*A^3*B*b^8*c*d^7 - 6*A*B^3*b^8*c*d^7 - 12*A^3*B*a*
b^7*d^8 - 12*A*B^3*a*b^7*d^8 + 30*C^4*a^5*b^3*c^3*d^5 + 19*C^4*a^2*b^6*c^4*
d^4 - 9*C^4*a^6*b^2*c^2*d^6 + 9*C^4*a^4*b^4*c^6*d^2 + 4*C^4*a^3*b^5*c^3*d^5
 + 4*C^4*a^2*b^6*c^2*d^6 - 3*C^4*a^4*b^4*c^4*d^4 - 3*C^4*a^4*b^4*c^2*d^6 +
3*C^4*a^2*b^6*c^6*d^2 + 28*B^4*a^3*b^5*c^5*d^3 + 27*B^4*a^4*b^4*c^2*d^6 - 1
7*B^4*a^4*b^4*c^4*d^4 - 10*B^4*a^2*b^6*c^4*d^4 + 8*B^4*a^3*b^5*c^3*d^5 + 8*
B^4*a^2*b^6*c^2*d^6 - 6*B^4*a^2*b^6*c^6*d^2 + 4*B^4*a^5*b^3*c^3*d^5 + 70*A^
4*a^2*b^6*c^4*d^4 + 58*A^4*a^2*b^6*c^2*d^6 - 56*A^4*a^3*b^5*c^3*d^5 + 15*A^
4*a^4*b^4*c^2*d^6 + B^2*C^2*b^8*c^2*d^6 - 18*A^3*C*b^8*d^8 + B^3*C*b^8*c^5*
d^3 + B*C^3*b^8*c^5*d^3 + 6*B^4*b^8*c^6*d^2 + 3*B^4*b^8*c^4*d^4 + 30*A^4*b^
8*c^4*d^4 + 27*A^4*b^8*c^2*d^6 + 3*C^4*a^6*b^2*d^8 + 8*B^4*a^4*b^4*d^8 + 4*
B^4*a^2*b^6*d^8 + 12*A^4*a^2*b^6*d^8 - 5*A^4*a^4*b^4*d^8 + 9*A^2*C^2*b^8*d^
8 + 9*A^2*B^2*b^8*d^8 + 9*A^4*b^8*d^8 + B^4*b^8*c^2*d^6 + C^4*a^4*b^4*d^8,
f, k), k, 1, 4))/f
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: NotImplementedError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**2/(c+d*tan(f*x
+e))**3,x)
```

```
[Out] Exception raised: NotImplementedError
```

3.90 $\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\,\left(A+B\tan(e+\right.$

Optimal. Leaf size=464

$$\frac{2(c+d\tan(e+fx))^{3/2}\left(40a^3Cd^3-6a^2bd^2(16cC-45Bd)+9ab^2d\left(35d^2(A-C)-14Bcd+8c^2C\right)-\left(b^3\left(42cd^2(\right.\right.\right.}{315d^4f}$$

```
[Out] -(a-I*b)^3*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d
)^(1/2)/f+(a+I*b)^3*(I*A-B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))*(c+I*d)^(1/2)/f+2*(a^3*B-3*a*b^2*B+3*a^2*b*(A-C)-b^3*(A-C))*(c+d*tan(f*x
+e))^(1/2)/f+2/315*(40*a^3*C*d^3-6*a^2*b*d^2*(-45*B*d+16*C*c)+9*a*b^2*d*(8*
c^2*C-14*B*c*d+35*(A-C)*d^2)-b^3*(16*c^3*C-24*B*c^2*d+42*c*(A-C)*d^2+105*B*
d^3))*(c+d*tan(f*x+e))^(3/2)/d^4/f+2/105*b*(21*b*(A*b+B*a-C*b)*d^2+4*(-a*d+
b*c)*(-3*B*b*d-2*C*a*d+2*C*b*c))*tan(f*x+e)*(c+d*tan(f*x+e))^(3/2)/d^3/f-2/
21*(-3*B*b*d-2*C*a*d+2*C*b*c)*(a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3/2)/d^2
/f+2/9*C*(a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^(3/2)/d/f
```

Rubi [A] time = 2.09, antiderivative size = 464, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 8, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3647, 3637, 3630, 3528, 3539, 3537, 63, 208}

$$\frac{2(c+d\tan(e+fx))^{3/2}\left(-6a^2bd^2(16cC-45Bd)+40a^3Cd^3+9ab^2d\left(35d^2(A-C)-14Bcd+8c^2C\right)+b^3\left(-\left(42c\right.\right.\right.}{315d^4f}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^3*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C
*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^3*(I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x
]]/Sqrt[c - I*d]])/f) + ((a + I*b)^3*(I*A - B - I*C)*Sqrt[c + I*d]*ArcTanh[
Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(a^3*B - 3*a*b^2*B + 3*a^2*
b*(A - C) - b^3*(A - C))*Sqrt[c + d*Tan[e + f*x]])/f + (2*(40*a^3*C*d^3 - 6
*a^2*b*d^2*(16*c*C - 45*B*d) + 9*a*b^2*d*(8*c^2*C - 14*B*c*d + 35*(A - C)*d
^2) - b^3*(16*c^3*C - 24*B*c^2*d + 42*c*(A - C)*d^2 + 105*B*d^3))*(c + d*Ta
n[e + f*x])^(3/2))/(315*d^4*f) + (2*b*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c
- a*d)*(2*b*c*C - 3*b*B*d - 2*a*C*d))*Tan[e + f*x]*(c + d*Tan[e + f*x])^(3/
2))/(105*d^3*f) - (2*(2*b*c*C - 3*b*B*d - 2*a*C*d)*(a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^(3/2))/(21*d^2*f) + (2*C*(a + b*Tan[e + f*x])^3*(c + d*
Tan[e + f*x])^(3/2))/(9*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
```

```
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
```

```
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int (a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}\,\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{2C(a+b\tan(e+fx))^3(c+\cdots}{9df}$$

$$= -\frac{2(2bcC-3bBd-2aCd)(\cdots}{\cdots}$$

$$= \frac{2b\left(21b(Ab+aB-bC)d^2\cdots\right.}{\cdots}$$

$$= \frac{2\left(40a^3Cd^3-6a^2bd^2(16cC\cdots\right.}{\cdots}$$

$$= \frac{2\left(a^3B-3ab^2B+3a^2b(A\cdots\right.}{\cdots}$$

$$= \frac{2\left(a^3B-3ab^2B+3a^2b(A\cdots\right.}{\cdots}$$

$$= \frac{2\left(a^3B-3ab^2B+3a^2b(A\cdots\right.}{\cdots}$$

$$= \frac{2\left(a^3B-3ab^2B+3a^2b(A\cdots\right.}{\cdots}$$

$$= -\frac{(a-ib)^3(iA+B-iC)\sqrt{c\cdots}}{\cdots}$$

Mathematica [B] time = 6.39, size = 1232, normalized size = 2.66

$$\frac{2C(c+d\tan(e+fx))^{3/2}(a+b\tan(e+fx))^3}{9df}+\frac{2\left(\frac{2\left(2\left(\frac{3b\left(21b(Ab-Cb+aB)d^2+4(bc-ad)(2bcC-2adC-3bBd)\right)\tan(e+fx)(c+d\tan(e+fx))^{3/2}}{10df}-\frac{2\left(\frac{2\left(b\left(\frac{3}{4}c(\cdots\right.\right.}{\cdots}\right.\right.}{\cdots}\right.}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^3*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*
x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^(3/2))/(9*d*f) + (2*((-3*(
2*b*c*C - 3*b*B*d - 2*a*C*d)*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(3
/2))/(7*d*f) + (2*((3*b*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*
C - 3*b*B*d - 2*a*C*d))*Tan[e + f*x]*(c + d*Tan[e + f*x])^(3/2))/(10*d*f) -
 (2*((2*((-15*a*d*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*
b*B*d - 2*a*C*d)))/8 + b*((-315*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (3
*c*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d - 2*a*C*d
)))/4))*(c + d*Tan[e + f*x])^(3/2))/(3*d*f) + ((I/2)*((-15*a*d*(a^2*(21*A -
 13*C)*d^2 + 4*b^2*c*(2*c*C - 3*B*d) - a*b*d*(16*c*C + 9*B*d)))/8 + (3*b*c*
(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d - 2*a*C*d)))
/4 + (15*a*d*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d
 - 2*a*C*d)))/8 + ((5*I)/2)*d*((63*a*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2)/4
 + (3*b*(a^2*(21*A - 13*C)*d^2 + 4*b^2*c*(2*c*C - 3*B*d) - a*b*d*(16*c*C +
```

```
9*B*d)))/4 - (3*b*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*
b*B*d - 2*a*C*d)))/4) - b*((-315*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (
3*c*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d - 2*a*C*
d)))/4))*((2*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]
])/(-c + I*d) + 2*Sqrt[c + d*Tan[e + f*x]]))/f - ((I/2)*((-15*a*d*(a^2*(21*
A - 13*C)*d^2 + 4*b^2*c*(2*c*C - 3*B*d) - a*b*d*(16*c*C + 9*B*d)))/8 + (3*b
*c*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d - 2*a*C*d
)))/4 + (15*a*d*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*
B*d - 2*a*C*d)))/8 - ((5*I)/2)*d*((63*a*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2
)/4 + (3*b*(a^2*(21*A - 13*C)*d^2 + 4*b^2*c*(2*c*C - 3*B*d) - a*b*d*(16*c*C
 + 9*B*d)))/4 - (3*b*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C -
 3*b*B*d - 2*a*C*d)))/4) - b*((-315*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8
+ (3*c*(21*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - 3*b*B*d - 2*a
*C*d)))/4))*((2*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I
*d]])/(-c - I*d) + 2*Sqrt[c + d*Tan[e + f*x]]))/f))/(5*d)))/(7*d)))/(9*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.62, size = 6661, normalized size = 14.36

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x)
```

```
[Out] result too large to display
```

maxima [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [**F(-1)**] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^3*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) +
 C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

sympy [**F**] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a + b\tan\left(e + fx\right)\right)^3 \sqrt{c + d\tan\left(e + fx\right)} \left(A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)\right) dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan
(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**3*sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*
x) + C*tan(e + f*x)**2), x)
```

3.91 $\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\,\big(A+B\tan(e+f$

Optimal. Leaf size=325

$$\frac{2(c+d\tan(e+fx))^{3/2}\left(20a^2Cd^2-14abd(2cC-5Bd)+b^2\left(35d^2(A-C)-14Bcd+8c^2C\right)\right)}{105d^3f}+\frac{2\left(a^2B+2ab(A-C\right.}{}$$

```
[Out] -(a-I*b)^2*(B+I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d
)^(1/2)/f-(a+I*b)^2*(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))*(c+I*d)^(1/2)/f+2*(a^2*B-b^2*B+2*a*b*(A-C))*(c+d*tan(f*x+e))^(1/2)/f+2/1
05*(20*a^2*C*d^2-14*a*b*d*(-5*B*d+2*C*c)+b^2*(8*c^2*C-14*B*c*d+35*(A-C)*d^2
))*(c+d*tan(f*x+e))^(3/2)/d^3/f-2/35*b*(-7*B*b*d-4*C*a*d+4*C*b*c)*tan(f*x+e
)*(c+d*tan(f*x+e))^(3/2)/d^2/f+2/7*C*(a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3
/2)/d/f
```

Rubi [A] time = 1.31, antiderivative size = 325, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 8, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3647, 3637, 3630, 3528, 3539, 3537, 63, 208}

$$\frac{2(c+d\tan(e+fx))^{3/2}\left(20a^2Cd^2-14abd(2cC-5Bd)+b^2\left(35d^2(A-C)-14Bcd+8c^2C\right)\right)}{105d^3f}+\frac{2\left(a^2B+2ab(A-C\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C
*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^2*(B + I*(A - C))*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x
]]/Sqrt[c - I*d]])/f) - ((a + I*b)^2*(B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[
Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(a^2*B - b^2*B + 2*a*b*(A -
 C))*Sqrt[c + d*Tan[e + f*x]])/f + (2*(20*a^2*C*d^2 - 14*a*b*d*(2*c*C - 5*B
*d) + b^2*(8*c^2*C - 14*B*c*d + 35*(A - C)*d^2))*(c + d*Tan[e + f*x])^(3/2)
)/(105*d^3*f) - (2*b*(4*b*c*C - 7*b*B*d - 4*a*C*d)*Tan[e + f*x]*(c + d*Tan[
e + f*x])^(3/2))/(35*d^2*f) + (2*C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*
x])^(3/2))/(7*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
```

```
!LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int (a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}\,\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{2C(a+b\tan(e+fx))^2(c}{7df}$$

$$= -\frac{2b(4bcC-7bBd-4aC}{}$$

$$= \frac{2\left(20a^2Cd^2-14abd(2cC\right.}{}$$

$$= \frac{2\left(a^2B-b^2B+2ab(A-\right.}{f}$$

$$= \frac{2\left(a^2B-b^2B+2ab(A-\right.}{f}$$

$$= \frac{2\left(a^2B-b^2B+2ab(A-\right.}{f}$$

$$= \frac{2\left(a^2B-b^2B+2ab(A-\right.}{f}$$

$$= -\frac{(a-ib)^2(B+i(A-C))}{}$$

Mathematica [A] time = 4.82, size = 314, normalized size = 0.97

$$\frac{2\left((c+d\tan(e+fx))^{3/2}\left(20a^2Cd^2+14abd(5Bd-2cC)+b^2\left(35d^2(A-C)-14Bcd+8c^2C\right)\right)+\frac{105}{2}d^3(a-ib)^2(i\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*
x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*((20*a^2*C*d^2 + 14*a*b*d*(-2*c*C + 5*B*d) + b^2*(8*c^2*C - 14*B*c*d + 3
5*(A - C)*d^2))*(c + d*Tan[e + f*x])^(3/2) + 3*b*d*(-4*b*c*C + 7*b*B*d + 4*
a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^(3/2) + 15*C*d^2*(a + b*Tan[e + f*
x])^2*(c + d*Tan[e + f*x])^(3/2) + (105*(a - I*b)^2*(I*A + B - I*C)*d^3*(-(
```

```
Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]]) + Sqrt[c + d
*Tan[e + f*x]]))/2 + (105*(a + I*b)^2*((-I)*A + B + I*C)*d^3*(-(Sqrt[c + I*
d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]) + Sqrt[c + d*Tan[e + f*
x]]))/2))/(105*d^3*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.46, size = 4775, normalized size = 14.69

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x)
```

```
[Out] -1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(
1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^2+2
/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d
^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*(c^2+d^2)^(1/2)*a*b+
2/f*B*a^2*(c+d*tan(f*x+e))^(1/2)-2/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((
2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-
2*c)^(1/2))*C*a*b*c-2/f*B*b^2*(c+d*tan(f*x+e))^(1/2)-1/2/f/d*ln(d*tan(f*x+e
)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B
*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*b*c+1/2/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x
+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1
/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a*b+1/2/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^
```

2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)*a*b*c-1/2/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c
)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^
2+d^2)^(1/2)*a*b+2/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(
1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*
a*b+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f
*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b^
2+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x
+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^2*c-1/4/f/d*ln((c+
d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^
(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b^2*c-2/f*d/(2*(c^2+d^2)^(1/2)-2*c)^
(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c
^2+d^2)^(1/2)-2*c)^(1/2))*B*a*b-4/3/f/d^2*C*(c+d*tan(f*x+e))^(3/2)*a*b*c+1/
4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2
)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b^2+1/4/
f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+
(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^2*c-2/f/(2*(c^2+d^2)^(1/
2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2
))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a*b*c-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)
(2(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^
2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^2-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(
2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)*a^2*c+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2
)*b^2*c-2/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(
1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*(c^2+d^2)^(
1/2)*a*b+2/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^
(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*(c^2+d^2)^
(1/2)*a*b+2/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)
^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a*b*c-2/f
/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2
)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*(c^2+d^2)^(1/2)*a*b+1/
4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-
c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^2-1/4/
f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-
(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b^2+2/f/(2
*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(
1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a*b*c-1/4/f/d*ln(d*tan(f*
x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2)
)*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b^2*c+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(
f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a^2-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*
x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(
1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b^2-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+
e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/

```
2)+2*c)^(1/2)*a^2*c+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^
2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b^
2*c-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2
*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a^2+1/f*d/(2*
(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1
/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a^2-1/f*d/(2*(c^2+d^2)^(1/
2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2
))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*b^2+2/3/f/d^3*C*(c+d*tan(f*x+e))^(3/2)*
b^2*c^2-4/5/f/d^3*C*(c+d*tan(f*x+e))^(5/2)*b^2*c+4/3/f/d*B*(c+d*tan(f*x+e))
^(3/2)*a*b-2/3/f/d^2*B*(c+d*tan(f*x+e))^(3/2)*b^2*c+4/5/f/d^2*C*(c+d*tan(f*
x+e))^(5/2)*a*b+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+
e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*b
^2-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2
)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a^2+1/2/f*ln((
c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2
)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*b-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1
/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2
+d^2)^(1/2)-2*c)^(1/2))*B*a^2*c+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((
2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2
*c)^(1/2))*B*b^2*c+1/2/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d
^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*b-1
/2/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c
-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*b+1/f/(2*(c^2+d^2)^(1/2
)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2)
)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)*a^2-1/f/(2*(c^2+d^2)^(1/
2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2
))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)*b^2+1/f/(2*(c^2+d^2)^(1
/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/
2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a^2*c-1/2/f*ln(d*tan(f*x+e)+c+(c+d*tan
(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2
)^(1/2)+2*c)^(1/2)*a*b-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan
(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)
)*B*(c^2+d^2)^(1/2)*a^2+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*ta
n(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2
))*B*(c^2+d^2)^(1/2)*b^2-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*t
an(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/
2))*B*b^2*c+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+
2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*b^2+1
/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*
(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a^2-1/f*d/(2*(c^2+
d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+
e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*b^2+2/7/f/d^3*b^2*C*(c+d*tan(f*
x+e))^(7/2)+2/3/f/d*C*(c+d*tan(f*x+e))^(3/2)*a^2-2/3/f/d*C*(c+d*tan(f*x+e))
^(3/2)*b^2+2/5/f/d^2*B*(c+d*tan(f*x+e))^(5/2)*b^2+2/3/f/d*A*(c+d*tan(f*x+e)
)^(3/2)*b^2+4/f*A*a*b*(c+d*tan(f*x+e))^(1/2)-4/f*C*a*b*(c+d*tan(f*x+e))^(1/
```

```
2)+1/4/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+
e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a^2-1/4/f*ln((c+d*tan
(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2)
)*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b^2-1/4/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+
e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/
2)+2*c)^(1/2)*a^2+1/4/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^
2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b^2
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) +
 C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a + b \tan \left(e + f x\right)\right)^{2} \sqrt{c + d \tan \left(e + f x\right)} \left(A + B \tan \left(e + f x\right) + C \tan^{2} \left(e + f x\right)\right)\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan
(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*
x) + C*tan(e + f*x)**2), x)
```

3.92 $\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\,\big(A+B\tan(e+fx$

Optimal. Leaf size=224

$$\frac{2(aB+Ab-bC)\sqrt{c+d\tan(e+fx)}}{f}-\frac{(b+ia)\sqrt{c-id}\,(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}+\frac{(-b+ia)\sqrt{c+id}\,(A}{}$$

```
[Out] -(I*a+b)*(A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d)^(1
/2)/f+(I*a-b)*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))*(c+I*
d)^(1/2)/f+2*(A*b+B*a-C*b)*(c+d*tan(f*x+e))^(1/2)/f-2/15*(-5*B*b*d-5*C*a*d+
2*C*b*c)*(c+d*tan(f*x+e))^(3/2)/d^2/f+2/5*b*C*tan(f*x+e)*(c+d*tan(f*x+e))^(
3/2)/d/f
```

Rubi [A] time = 0.63, antiderivative size = 224, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 7, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.156, Rules used = {3637, 3630, 3528, 3539, 3537, 63, 208}

$$\frac{2(aB+Ab-bC)\sqrt{c+d\tan(e+fx)}}{f}-\frac{(b+ia)\sqrt{c-id}\,(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}+\frac{(-b+ia)\sqrt{c+id}\,(A}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/S
qrt[c - I*d]])/f) + ((I*a - b)*(A + I*B - C)*Sqrt[c + I*d]*ArcTanh[Sqrt[c +
 d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(A*b + a*B - b*C)*Sqrt[c + d*Tan[e
+ f*x]])/f - (2*(2*b*c*C - 5*b*B*d - 5*a*C*d)*(c + d*Tan[e + f*x])^(3/2))/(
15*d^2*f) + (2*b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(3/2))/(5*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int (a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}\,\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{2bC\tan(e+fx)(c+d\tan(e}{5df}$$

$$= -\frac{2(2bcC-5bBd-5aCd)(c}{15d^2f}$$

$$= \frac{2(Ab+aB-bC)\sqrt{c+d\tan}}{f}$$

$$= \frac{2(Ab+aB-bC)\sqrt{c+d\tan}}{f}$$

$$= \frac{2(Ab+aB-bC)\sqrt{c+d\tan}}{f}$$

$$= \frac{2(Ab+aB-bC)\sqrt{c+d\tan}}{f}$$

$$= -\frac{(ia+b)(A-iB-C)\sqrt{c-}}{f}$$

Mathematica [A] time = 1.99, size = 220, normalized size = 0.98

$$15d(b+ia)(A-iB-C)\left(\sqrt{c+d\tan(e+fx)}-\sqrt{c-id}\,\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)\right)+15d(b-ia)(A+iB-C)\left(\sqrt{c+}\right.$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] ((2*(-2*b*c*C + 5*b*B*d + 5*a*C*d)*(c + d*Tan[e + f*x])^(3/2))/d + 6*b*C*Ta
n[e + f*x]*(c + d*Tan[e + f*x])^(3/2) + 15*(I*a + b)*(A - I*B - C)*d*(-(Sqr
t[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]]) + Sqrt[c + d*Ta
n[e + f*x]]) + 15*((-I)*a + b)*(A + I*B - C)*d*(-(Sqrt[c + I*d]*ArcTanh[Sqr
t[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]) + Sqrt[c + d*Tan[e + f*x]]))/(15*d*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.50, size = 3028, normalized size = 13.52

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x)
```

```
[Out] 1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(
c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)*a+1
/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c
+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*b*c+1/4/f/d*ln(d*tan
(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1
/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*
arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2
)^(1/2)-2*c)^(1/2))*C*b*c-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2
+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1
/2))*B*a*c+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/
2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a*c+1/4/
f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c
^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b+1/4/f*ln((c+d*tan(f*x+e))^
(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c
^2+d^2)^(1/2)+2*c)^(1/2)*a-1/4/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/
2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/
2)*b-1/4/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)
^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b-1/4/f*ln(d*tan(f*
x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2)
)*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a+1/4/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e)
```

)^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)
+2*c)^(1/2)*b+2/3/f/d*B*(c+d*tan(f*x+e))^(3/2)*b+2/5/f/d^2*C*b*(c+d*tan(f*x
+e))^(5/2)+2/3/f/d*C*(c+d*tan(f*x+e))^(3/2)*a-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)
^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(
c^2+d^2)^(1/2)-2*c)^(1/2))*A*a-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*
(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*
c)^(1/2))*B*(c^2+d^2)^(1/2)*a+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(
c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c
)^(1/2))*C*(c^2+d^2)^(1/2)*b+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(
c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)
^(1/2))*A*(c^2+d^2)^(1/2)*b+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^
(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^
(1/2)*b*c+1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/
2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1
/2)*b-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan
(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*
b+1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)
-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b-1/f*d/(2*(c^2
+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+
2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a-2/3/f/d^2*C*(c+d*tan(f*x+e))
^(3/2)*b*c-1/f*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(
1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b+1/f*
d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+
d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a+1/f*d/(2*(c^2+d^2)^
(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(
1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*
arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2
)^(1/2)-2*c)^(1/2))*A*b*c-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2
+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1
/2))*C*(c^2+d^2)^(1/2)*b-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+
d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/
2))*A*b*c-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2
)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*(c^2+d^2)
^(1/2)*b-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b*c+1/4/f/d*l
n(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+
d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a-1/4/f/d*ln((c
+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)
^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a+1/4/f/d*ln((c+d*t
an(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/
2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(
f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)*a*c+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+
2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*
(c^2+d^2)^(1/2)*a-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)

```
^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c-
1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1
/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a+2/f*
A*(c+d*tan(f*x+e))^(1/2)*b+2/f*B*(c+d*tan(f*x+e))^(1/2)*a-2/f*C*b*(c+d*tan(
f*x+e))^(1/2)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)\sqrt{d\tan\left(fx+e\right)+c}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)*sqrt
(d*tan(f*x + e) + c), x)
```

mupad [B] time = 60.11, size = 22955, normalized size = 102.48

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C
*tan(e + f*x)^2),x)
```

```
[Out] ((2*B*a*d - 4*C*a*c)/(d*f) + (4*C*a*c)/(d*f))*(c + d*tan(e + f*x))^(1/2) +
((2*B*b*d - 6*C*b*c)/(3*d^2*f) + (4*C*b*c)/(3*d^2*f))*(c + d*tan(e + f*x))^
(3/2) + (c + d*tan(e + f*x))^(1/2)*(2*c*((2*B*b*d - 6*C*b*c)/(d^2*f) + (4*C
*b*c)/(d^2*f)) + (2*A*b*d^2 + 6*C*b*c^2 - 4*B*b*c*d)/(d^2*f) - (2*C*b*(d^4*
f + c^2*d^2*f))/(d^4*f^2)) - atan(((((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*
A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(
1/2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^
4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 +
 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*
B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b
^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^
2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B
^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)
/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*
d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4
*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^
2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4
 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^
2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B
```

*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*1i - (((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*1i)/((16*(B^3*b^3*d^5 - A^3*b^3*c^3*d^2 + B^3*b^3*c^2*d^3 + C^3*b^3*c^3*d^2 + A^2*B*b^3*d^5 + B*C^2*b^3*d^5 - A^3*b^3*c*d^4 + C^3*b^3*c*d^4 - A*B^2*b^3*c*d^4 - 3*A*C^2*b^3*c*d^4 + 3*A^2*C*b^3*c*d^4 + B^2*C*b^3*c*d^4 - A*B^2*b^3*c^3*d^2 + A^2*B*b^3*c^2*d^3 - 3*A*C^2*b^3*c^3*d^2 + 3*A^2*C*b^3*c^3*d^2 + B*C^2*b^3*c^2*d^3 + B^2*C*b^3*c^3*d^2 - 2*A*B*C*b^3*d^5 - 2*A*B*C*b^3*c^2*d^3))/f^3 + (((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2* d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A ^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^ 3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d *f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f ^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2 *c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^ 2))^(1/2))*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2 *f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^ 4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B *C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C* b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C* b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2* d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4 *b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4* A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2 *C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^ 4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f ^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^ 4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/ (2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) + (((8*(4*A*b*d^ 4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 + 64*c* d^2*(c + d*tan(e + f*x))^(1/2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3 *B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4* c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c *d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^ 2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((A^2*b^2* c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4 *b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4* d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2 *f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4* B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B* C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^ 2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C *b^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2* b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/

f^2)*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^
4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 +
2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*
B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b
^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^
2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B
^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)
/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)))*((A^2*b^2*c)/(4*f^2) - (4*A*C^3*b^4
*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^
4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d
^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^
4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B
^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*
B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2)
- (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*2i
- atan(((((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^
2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*b^4*d^2*f^4
- B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^
4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 -
4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^
3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4
*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*
c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b
^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*
f^2))^(1/2))*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^
4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4
*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^
2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4
*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B
*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f
^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*
C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1
/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^
2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d
^3 - 4*B*C*b^2*c*d^3))/f^2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b
^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4
+ 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2
*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*
b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d
^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (
A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d
)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*1i - (((8*(4*A
*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 +
64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4
- C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*

c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2
*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 +
4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2
*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*
f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A
*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((4*A
*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*
A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C
^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^
4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4
 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 +
 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/
(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) +
 (B*C*b^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 -
 B^2*b^2*d^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^
2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d
^3))/f^2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b
^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^
2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f
^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^
3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^
2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2)
 - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b
^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*1i)/((16*(B^3*b^3*d^5 - A^3*b^3*
c^3*d^2 + B^3*b^3*c^2*d^3 + C^3*b^3*c^3*d^2 + A^2*B*b^3*d^5 + B*C^2*b^3*d^5
 - A^3*b^3*c*d^4 + C^3*b^3*c*d^4 - A*B^2*b^3*c*d^4 - 3*A*C^2*b^3*c*d^4 + 3*
A^2*C*b^3*c*d^4 + B^2*C*b^3*c*d^4 - A*B^2*b^3*c^3*d^2 + A^2*B*b^3*c^2*d^3 -
 3*A*C^2*b^3*c^3*d^2 + 3*A^2*C*b^3*c^3*d^2 + B*C^2*b^3*c^2*d^3 + B^2*C*b^3*
c^3*d^2 - 2*A*B*C*b^3*d^5 - 2*A*B*C*b^3*c^2*d^3))/f^3 + (((8*(4*A*b*d^4*f^2
 - 4*C*b*d^4*f^2 + 4*A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 - 64*c*d^2*(
c + d*tan(e + f*x))^(1/2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4
*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 +
2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B
^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^
4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2
*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^
2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/
(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((4*A*C^3*b^4*d
^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*
d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2
*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4
- 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2
*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*
C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) +
(C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*

d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d
^4 + C^2*b^2*d^4 - A^2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*
A*C*b^2*d^4 + 2*A*C*b^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*
((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4
 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*
A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B
^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*
d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*
f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^
2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f
^2) + (B*C*b^2*d)/(2*f^2))^(1/2) + (((8*(4*A*b*d^4*f^2 - 4*C*b*d^4*f^2 + 4*
A*b*c^2*d^2*f^2 - 4*C*b*c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(
1/2)*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^
2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4
 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 +
4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b
^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4
*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B
^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)
/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2))*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*
f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b
^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*
c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4
 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*
B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/
(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) -
 (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2) + (
16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^4 - B^2*b^2*d^4 + C^2*b^2*d^4 - A^
2*b^2*c^2*d^2 + B^2*b^2*c^2*d^2 - C^2*b^2*c^2*d^2 - 2*A*C*b^2*d^4 + 2*A*C*b
^2*c^2*d^2 + 4*A*B*b^2*c*d^3 - 4*B*C*b^2*c*d^3))/f^2)*((4*A*C^3*b^4*d^2*f^4
 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^
4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 -
 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^
3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 - 4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4
*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*
c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b
^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*
f^2))^(1/2)))*((4*A*C^3*b^4*d^2*f^4 - B^4*b^4*d^2*f^4 - C^4*b^4*d^2*f^4 - A
^4*b^4*d^2*f^4 + 4*A^3*C*b^4*d^2*f^4 - 4*A^2*B^2*b^4*c^2*f^4 + 2*A^2*B^2*b^
4*d^2*f^4 - 6*A^2*C^2*b^4*d^2*f^4 - 4*B^2*C^2*b^4*c^2*f^4 + 2*B^2*C^2*b^4*d
^2*f^4 + 4*A*B^3*b^4*c*d*f^4 - 4*A^3*B*b^4*c*d*f^4 + 4*B*C^3*b^4*c*d*f^4 -
4*B^3*C*b^4*c*d*f^4 + 8*A*B^2*C*b^4*c^2*f^4 - 4*A*B^2*C*b^4*d^2*f^4 - 12*A*
B*C^2*b^4*c*d*f^4 + 12*A^2*B*C*b^4*c*d*f^4)^(1/2)/(4*f^4) + (A^2*b^2*c)/(4*
f^2) - (B^2*b^2*c)/(4*f^2) + (C^2*b^2*c)/(4*f^2) - (A*B*b^2*d)/(2*f^2) - (A
*C*b^2*c)/(2*f^2) + (B*C*b^2*d)/(2*f^2))^(1/2)*2i - atan(((((8*(4*B*a*d^4*f

^2 + 4*B*a*c^2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((B^2*a^
2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4
 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*
c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2
*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 +
4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2
*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*
f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B
*C*a^2*d)/(2*f^2))^(1/2))*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A
*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*
A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C
^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^
4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4
 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 +
 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/
(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan
(e + f*x))^(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2
 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 +
4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)
/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a
^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^
2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f
^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^
3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^
2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2)
 + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)*1
i - (((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e +
 f*x))^(1/2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*
f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2
*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^
4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4
*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*
a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a
^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*
C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2))*((B^2*a^2*c)/(4*f^2) - (A^2*
a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 -
 A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*
a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4
*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4
- 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*
A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(
4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(
1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*
d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 +
 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((B^2*a^2*c)/

```
(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^
4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f
^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4
+ 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C
^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^
4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4)
- (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^
2*d)/(2*f^2))^(1/2)*1i)/((((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))/f^3 - 64
*c*d^2*(c + d*tan(e + f*x))^(1/2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2
) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*
f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 -
 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*
A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4
*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c
*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B
*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2))*((B^2*a
^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^
4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4
*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^
2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 +
 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^
2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4
*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (
B*C*a^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^4 - B
^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*
d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3
))/f^2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 -
 B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4
- 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4
*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*
B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c
^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*
d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2
*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) - (16*(A^3*a^3*d^5 - C^3*a^3*d^5 +
 A^3*a^3*c^2*d^3 + B^3*a^3*c^3*d^2 - C^3*a^3*c^2*d^3 + A*B^2*a^3*d^5 + 3*A*
C^2*a^3*d^5 - 3*A^2*C*a^3*d^5 - B^2*C*a^3*d^5 + B^3*a^3*c*d^4 + A^2*B*a^3*c
*d^4 + B*C^2*a^3*c*d^4 + A*B^2*a^3*c^2*d^3 + A^2*B*a^3*c^3*d^2 + 3*A*C^2*a^
3*c^2*d^3 - 3*A^2*C*a^3*c^2*d^3 + B*C^2*a^3*c^3*d^2 - B^2*C*a^3*c^2*d^3 - 2
*A*B*C*a^3*c*d^4 - 2*A*B*C*a^3*c^3*d^2))/f^3 + (((8*(4*B*a*d^4*f^2 + 4*B*a*
c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((B^2*a^2*c)/(4*f^2
) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*
d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2
*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^
2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4
*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*
```

f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2
*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(
2*f^2))^(1/2))*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^
2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d
^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*
f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 -
4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*
C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C
*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (
A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^
(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*
c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c
*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((B^2*a^2*c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) -
(4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4
+ 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A
^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^
3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d
*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f
^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2
*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)))*((B^2*a^2*
c)/(4*f^2) - (A^2*a^2*c)/(4*f^2) - (4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 -
C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^
2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f
^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*
B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C
*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^
4) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C
*a^2*d)/(2*f^2))^(1/2)*2i - atan(((((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))
/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*
d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B
^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*
a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d
*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 -
4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1
/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^
2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)
)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f
^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 -
6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A
*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*
c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*
d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*
a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2
*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^
2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2

*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*
a^2*c*d^3))/f^2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4
- A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2
*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^
4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4
 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12
*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/
(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) +
 (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)*1i - (((8*(4*B*a*d^4*f^2
+ 4*B*a*c^2*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*a
^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*
a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4
*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*
f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A
*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^
2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2
) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*
a^2*d)/(2*f^2))^(1/2))*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^
2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A
^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*
C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c
*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^
4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a
^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*
f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e
+ f*x))^(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 +
B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A
*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f
^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^
4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c
^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4
+ 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B
^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(
4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) +
(A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)*1i)/
((((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*
x))^(1/2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a
^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^
2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f
^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^
3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^
2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2)
 + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a
^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2))*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4
*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*

B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2
*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*
d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 -
 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(
1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f
^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2
) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^4 - B^2*a^2*d^4 + C^2*a^2*d^4
 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^2 - 2*A*C*a^2*d^4 + 2*
A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))/f^2)*((4*A*C^3*a^4*d^
2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d
^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*
f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 -
 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*
C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C
*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (
C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d
)/(2*f^2))^(1/2) - (16*(A^3*a^3*d^5 - C^3*a^3*d^5 + A^3*a^3*c^2*d^3 + B^3*a
^3*c^3*d^2 - C^3*a^3*c^2*d^3 + A*B^2*a^3*d^5 + 3*A*C^2*a^3*d^5 - 3*A^2*C*a^
3*d^5 - B^2*C*a^3*d^5 + B^3*a^3*c*d^4 + A^2*B*a^3*c*d^4 + B*C^2*a^3*c*d^4 +
 A*B^2*a^3*c^2*d^3 + A^2*B*a^3*c^3*d^2 + 3*A*C^2*a^3*c^2*d^3 - 3*A^2*C*a^3*
c^2*d^3 + B*C^2*a^3*c^3*d^2 - B^2*C*a^3*c^2*d^3 - 2*A*B*C*a^3*c*d^4 - 2*A*B
*C*a^3*c^3*d^2))/f^3 + (((8*(4*B*a*d^4*f^2 + 4*B*a*c^2*d^2*f^2))/f^3 + 64*c
d^2(c + d*tan(e + f*x))^(1/2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C
^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*
f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4
 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*
C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a
^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4)
 - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a
^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2))*((4*A*C^3
*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*
C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a
^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*
d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8
*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*
A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f
^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*
C*a^2*d)/(2*f^2))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^4 - B^2
*a^2*d^4 + C^2*a^2*d^4 - A^2*a^2*c^2*d^2 + B^2*a^2*c^2*d^2 - C^2*a^2*c^2*d^
2 - 2*A*C*a^2*d^4 + 2*A*C*a^2*c^2*d^2 + 4*A*B*a^2*c*d^3 - 4*B*C*a^2*c*d^3))
/f^2)*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d
^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^
4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 +
 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*
a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^

```
4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (
B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2) + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c
)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)))*((4*A*C^3*a^4*d^2*f^4 - B^4*a^4*d^
2*f^4 - C^4*a^4*d^2*f^4 - A^4*a^4*d^2*f^4 + 4*A^3*C*a^4*d^2*f^4 - 4*A^2*B^2
*a^4*c^2*f^4 + 2*A^2*B^2*a^4*d^2*f^4 - 6*A^2*C^2*a^4*d^2*f^4 - 4*B^2*C^2*a^
4*c^2*f^4 + 2*B^2*C^2*a^4*d^2*f^4 + 4*A*B^3*a^4*c*d*f^4 - 4*A^3*B*a^4*c*d*f
^4 + 4*B*C^3*a^4*c*d*f^4 - 4*B^3*C*a^4*c*d*f^4 + 8*A*B^2*C*a^4*c^2*f^4 - 4*
A*B^2*C*a^4*d^2*f^4 - 12*A*B*C^2*a^4*c*d*f^4 + 12*A^2*B*C*a^4*c*d*f^4)^(1/2
)/(4*f^4) - (A^2*a^2*c)/(4*f^2) + (B^2*a^2*c)/(4*f^2) - (C^2*a^2*c)/(4*f^2)
 + (A*B*a^2*d)/(2*f^2) + (A*C*a^2*c)/(2*f^2) - (B*C*a^2*d)/(2*f^2))^(1/2)*2
i + (2*C*a*(c + d*tan(e + f*x))^(3/2))/(3*d*f) + (2*C*b*(c + d*tan(e + f*x)
)^(5/2))/(5*d^2*f)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a+b\tan\left(e+fx\right)\right)\sqrt{c+d\tan\left(e+fx\right)}\,\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*
x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x)
+ C*tan(e + f*x)**2), x)
```

3.93 $\int \sqrt{c+d\tan(e+fx)}\,\left(A+B\tan(e+fx)+C\tan^2(e+\right.$

Optimal. Leaf size=155

$$-\frac{\sqrt{c-id}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}-\frac{\sqrt{c+id}\,(B-i(A-C))\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}+\frac{2B\sqrt{c+d\tan(e}}{f}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d)^(1/2)/f-
(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))*(c+I*d)^(1/2)/f+2
*B*(c+d*tan(f*x+e))^(1/2)/f+2/3*C*(c+d*tan(f*x+e))^(3/2)/d/f
```

Rubi [A] time = 0.31, antiderivative size = 155, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, integrand size = 35, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.171, Rules used = {3630, 3528, 3539, 3537, 63, 208}

$$-\frac{\sqrt{c-id}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}-\frac{\sqrt{c+id}\,(B-i(A-C))\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f}+\frac{2B\sqrt{c+d\tan(e}}{f}$$

Antiderivative was successfully verified.

```
[In] Int[Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c -
I*d]])/f) - ((B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c + I*d]])/f + (2*B*Sqrt[c + d*Tan[e + f*x]])/f + (2*C*(c + d*Tan[e +
 f*x])^(3/2))/(3*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
```

```
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int \sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{2C(c+d\tan(e+fx))^{3/2}}{3df} + \int (A-C+B\tan$$

$$= \frac{2B\sqrt{c+d\tan(e+fx)}}{f} + \frac{2C(c+d\tan(e+f}{3df}$$

$$= \frac{2B\sqrt{c+d\tan(e+fx)}}{f} + \frac{2C(c+d\tan(e+f}{3df}$$

$$= \frac{2B\sqrt{c+d\tan(e+fx)}}{f} + \frac{2C(c+d\tan(e+f}{3df}$$

$$= \frac{2B\sqrt{c+d\tan(e+fx)}}{f} + \frac{2C(c+d\tan(e+f}{3df}$$

$$= -\frac{(B+i(A-C))\sqrt{c-id}\,\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+f}}{\sqrt{c-id}}\right.}{f}$$

Mathematica [A] time = 0.56, size = 150, normalized size = 0.97

$$\frac{-3id\sqrt{c-id}\,(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)+3id\sqrt{c+id}\,(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)+2\sqrt{c+d\,\mathrm{t}}}{3df}$$

Antiderivative was successfully verified.

```
[In] Integrate[Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),
x]
```

```
[Out] ((-3*I)*(A - I*B - C)*Sqrt[c - I*d]*d*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt
[c - I*d]] + (3*I)*(A + I*B - C)*Sqrt[c + I*d]*d*ArcTanh[Sqrt[c + d*Tan[e +
 f*x]]/Sqrt[c + I*d]] + 2*Sqrt[c + d*Tan[e + f*x]]*(c*C + 3*B*d + C*d*Tan[e
 + f*x]))/(3*d*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.44, size = 1472, normalized size = 9.50

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] 2/3*C*(c+d*tan(f*x+e))^(3/2)/d/f+2*B*(c+d*tan(f*x+e))^(1/2)/f+d/f/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*
c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A-d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2
)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2))*C-d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d
^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2
))*A+d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2
)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C+1/4/d/f*ln((c+
d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^
(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)-1/4/d/f*ln((c+d*tan(
f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))
*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c-1/4/d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^
2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)*(c^2+d^2)^(1/2)+1/4/d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)
^(1/2)*c+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^
(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^
(1/2)-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/
2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*c-1/f/(2*(c^2
+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+
2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)+1/f/(2*(c^2+d^
2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c
)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*c-1/4/d/f*ln(d*tan(f*x+e)+c+(c+d*
tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)+1/4/d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*
```

```
x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(
1/2)+2*c)^(1/2)*c+1/4/d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2
+d^2)^(1/2)-1/4/d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(
1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c-1/4/f*ln
(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d
^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+1/4/f*ln((c+d*tan(f*x+e))^(1/2)*
(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2
)^(1/2)+2*c)^(1/2)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\sqrt{d\tan\left(fx+e\right)+c}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(d*tan(f*x + e) + c),
 x)
```

mupad [B] time = 17.40, size = 1199, normalized size = 7.74

$$2\,\mathrm{atanh}\left(\frac{32\,B^2\,d^4\,\sqrt{\frac{B^2\,c}{4\,f^2}-\frac{\sqrt{-B^4\,d^2\,f^4}}{4\,f^4}}\,\sqrt{c+d\,\tan\left(e+f\,x\right)}}{\frac{16\,B\,d^4\,\sqrt{-B^4\,d^2\,f^4}}{f^3}+\frac{16\,B\,c^2\,d^2\,\sqrt{-B^4\,d^2\,f^4}}{f^3}}-\frac{32\,c\,d^2\,\sqrt{\frac{B^2\,c}{4\,f^2}-\frac{\sqrt{-B^4\,d^2\,f^4}}{4\,f^4}}\,\sqrt{c+d\,\tan\left(e+f\,x\right)}\,\sqrt{-}}{\frac{16\,B\,d^4\,\sqrt{-B^4\,d^2\,f^4}}{f}+\frac{16\,B\,c^2\,d^2\,\sqrt{-B^4\,d^2\,f^4}}{f}}\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] 2*atanh((32*B^2*d^4*((B^2*c)/(4*f^2) - (-B^4*d^2*f^4)^(1/2)/(4*f^4))^(1/2)*
(c + d*tan(e + f*x))^(1/2))/((16*B*d^4*(-B^4*d^2*f^4)^(1/2))/f^3 + (16*B*c^
2*d^2*(-B^4*d^2*f^4)^(1/2))/f^3) - (32*c*d^2*((B^2*c)/(4*f^2) - (-B^4*d^2*f
^4)^(1/2)/(4*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(-B^4*d^2*f^4)^(1/2))/(
(16*B*d^4*(-B^4*d^2*f^4)^(1/2))/f + (16*B*c^2*d^2*(-B^4*d^2*f^4)^(1/2))/f))
*(-((-B^4*d^2*f^4)^(1/2) - B^2*c*f^2)/(4*f^4))^(1/2) - 2*atanh((32*B^2*d^4*
((-B^4*d^2*f^4)^(1/2)/(4*f^4) + (B^2*c)/(4*f^2))^(1/2)*(c + d*tan(e + f*x))
^(1/2))/((16*B*d^4*(-B^4*d^2*f^4)^(1/2))/f^3 + (16*B*c^2*d^2*(-B^4*d^2*f^4)
^(1/2))/f^3) + (32*c*d^2*((-B^4*d^2*f^4)^(1/2)/(4*f^4) + (B^2*c)/(4*f^2))^(
1/2)*(c + d*tan(e + f*x))^(1/2)*(-B^4*d^2*f^4)^(1/2))/((16*B*d^4*(-B^4*d^2*
f^4)^(1/2))/f + (16*B*c^2*d^2*(-B^4*d^2*f^4)^(1/2))/f))*(((-B^4*d^2*f^4)^(1
/2) + B^2*c*f^2)/(4*f^4))^(1/2) - atanh((f^3*(-((-A^4*d^2*f^4)^(1/2) + A^2*
```

```
c*f^2)/f^4)^(1/2)*((16*(A^2*d^4 - A^2*c^2*d^2)*(c + d*tan(e + f*x))^(1/2))/
f^2 + (16*c*d^2*((-A^4*d^2*f^4)^(1/2) + A^2*c*f^2)*(c + d*tan(e + f*x))^(1/
2))/f^4))/(16*(A^3*d^5 + A^3*c^2*d^3)))*(-((-A^4*d^2*f^4)^(1/2) + A^2*c*f^2
)/f^4)^(1/2) - atanh((f^3*(((-A^4*d^2*f^4)^(1/2) - A^2*c*f^2)/f^4)^(1/2)*((
16*(A^2*d^4 - A^2*c^2*d^2)*(c + d*tan(e + f*x))^(1/2))/f^2 - (16*c*d^2*((-A
^4*d^2*f^4)^(1/2) - A^2*c*f^2)*(c + d*tan(e + f*x))^(1/2))/f^4))/(16*(A^3*d
^5 + A^3*c^2*d^3)))*(((-A^4*d^2*f^4)^(1/2) - A^2*c*f^2)/f^4)^(1/2) + atanh(
(f^3*(-((-C^4*d^2*f^4)^(1/2) + C^2*c*f^2)/f^4)^(1/2)*((16*(C^2*d^4 - C^2*c^
2*d^2)*(c + d*tan(e + f*x))^(1/2))/f^2 + (16*c*d^2*((-C^4*d^2*f^4)^(1/2) +
C^2*c*f^2)*(c + d*tan(e + f*x))^(1/2))/f^4))/(16*(C^3*d^5 + C^3*c^2*d^3)))*
(-((-C^4*d^2*f^4)^(1/2) + C^2*c*f^2)/f^4)^(1/2) + atanh((f^3*(((-C^4*d^2*f^
4)^(1/2) - C^2*c*f^2)/f^4)^(1/2)*((16*(C^2*d^4 - C^2*c^2*d^2)*(c + d*tan(e
+ f*x))^(1/2))/f^2 - (16*c*d^2*((-C^4*d^2*f^4)^(1/2) - C^2*c*f^2)*(c + d*ta
n(e + f*x))^(1/2))/f^4))/(16*(C^3*d^5 + C^3*c^2*d^3)))*(((-C^4*d^2*f^4)^(1/
2) - C^2*c*f^2)/f^4)^(1/2) + (2*B*(c + d*tan(e + f*x))^(1/2))/f + (2*C*(c +
 d*tan(e + f*x))^(3/2))/(3*d*f)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \sqrt{c + d \tan\left(e + fx\right)} \left(A + B \tan\left(e + fx\right) + C \tan^2\left(e + fx\right)\right) dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2),
 x)
```

3.94 $\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$

Optimal. Leaf size=234

$$-\frac{2\sqrt{bc-ad}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{3/2}f\left(a^2+b^2\right)}-\frac{\sqrt{c-id}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)}+\frac{\sqrt{c}}{}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d)^(1/2)/(a
-I*b)/f+(I*A-B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))*(c+I*d)^(
1/2)/(a+I*b)/f-2*(A*b^2-a*(B*b-C*a))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)
/(-a*d+b*c)^(1/2))*(-a*d+b*c)^(1/2)/b^(3/2)/(a^2+b^2)/f+2*C*(c+d*tan(f*x+e)
)^(1/2)/b/f
```

Rubi [A] time = 1.09, antiderivative size = 234, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3647, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{2\sqrt{bc-ad}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{3/2}f\left(a^2+b^2\right)}-\frac{\sqrt{c-id}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)}+\frac{\sqrt{c}}{}$$

Antiderivative was successfully verified.

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x]),x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c -
I*d]])/((a - I*b)*f)) + ((I*A - B - I*C)*Sqrt[c + I*d]*ArcTanh[Sqrt[c + d*T
an[e + f*x]]/Sqrt[c + I*d]])/((a + I*b)*f) - (2*(A*b^2 - a*(b*B - a*C))*Sqr
t[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(
b^(3/2)*(a^2 + b^2)*f) + (2*C*Sqrt[c + d*Tan[e + f*x]])/(b*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
```

```
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \frac{2C\sqrt{c+d\tan(e+fx)}}{bf} + \frac{2\int \frac{\frac{1}{2}(Abc-aCd)+\frac{1}{2}b(B}{\ }}{\ }$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{bf} + \frac{2\int \frac{\frac{1}{2}b(bBc+b(A-C)d+}{\ }}{\ }$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{bf} + \frac{((A-iB-C)(c-i}{2(a}$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{bf} - \frac{(i(A+iB-C)(c+}{\ }$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{bc-ad}\ \tanh^{-1}\left(\frac{\sqrt{b}}{\ }\right.}{b^{3/2}\left(a^2+b^2\right)f}$$

$$= \frac{(A-iB-C)\sqrt{c-id}\ \tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)f}$$

Mathematica [A] time = 0.69, size = 233, normalized size = 1.00

$$\frac{2\sqrt{b}\,C\left(a^2+b^2\right)\sqrt{c+d\tan(e+fx)}+b^{3/2}(b-ia)\sqrt{c-id}\,(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)+b^{3/2}(b+ia)\sqrt{c}}{b^{3/2}f\left(a^2+b^2\right.}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x]),x]
```

```
[Out] (b^(3/2)*((-I)*a + b)*(A - I*B - C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e
+ f*x]]/Sqrt[c - I*d]] + b^(3/2)*(I*a + b)*(A + I*B - C)*Sqrt[c + I*d]*ArcT
```

```
anh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]] - 2*(A*b^2 + a*(-(b*B) + a*C))*
Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]]
 + 2*Sqrt[b]*(a^2 + b^2)*C*Sqrt[c + d*Tan[e + f*x]])/(b^(3/2)*(a^2 + b^2)*f
)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.75, size = 3576, normalized size = 15.28

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x)
```

```
[Out] -2/f*b/(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan((c+d*tan(f*x+e))^(1/2)*b/((a*d-
b*c)*b)^(1/2))*B*a*c-2/f/b/(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan((c+d*tan(f*
x+e))^(1/2)*b/((a*d-b*c)*b)^(1/2))*a^3*C*d+1/4/f/(a^2+b^2)/d*ln((c+d*tan(f*
x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A
*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a+1/4/f/(a^2+b^2)/d*ln((c+d*
tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1
/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b-1/4/f/(a^2+b^2)*ln(d
*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2
)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b+1/4/f/(a^2+b^2)/d*ln((c+d*tan(f*
x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C
*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c+2*C*(c+d*tan(f*x+e))^(1/2)/b/f+1/f/(a^2+
```

b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)*a+1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*(c^2+d^2)^(1/2)*b-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*(c^2+d^2)^(1/2)*b+1/f/(a^2+b^2)*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*b*c+2/f*b^2/(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan((c+d*tan(f*x+e))^(1/2)*b/((a*d-b*c)*b)^(1/2))*A*c+2/f/(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan((c+d*tan(f*x+e))^(1/2)*b/((a*d-b*c)*b)^(1/2))*B*a^2*d+2/f/(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan((c+d*tan(f*x+e))^(1/2)*b/((a*d-b*c)*b)^(1/2))*C*a^2*c+1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*b*c+1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*(c^2+d^2)^(1/2)*b+1/f/(a^2+b^2)*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a-1/f/(a^2+b^2)*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)*a+1/f/(a^2+b^2)*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b-1/f/(a^2+b^2)*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a-1/f/(a^2+b^2)*d/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*b*c-1/4/f/(a^2+b^2)/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*b+1/4/f/(a^2+b^2)/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c+1/4/f/(a^2+b^2)/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b*c+1/4/f/(a^2+b^2)/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a-1/4/f/(a^2+b^2)/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c-2/f*b/(a^2+b^2)/((a*d-b*c)*b)^(1/2)*arctan((c+d*tan(f*x+e))^(1/2)*b/((a*d-b*c)*b)^(1/2))*A*a*d-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a*c+1/f/(a^2+b^2)/(2*(c^2

```
+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+
2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*b*c+1/f/(a^2+b^2)/(2*(c^2+d^2)
^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^
(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a*c-1/f/(a^2+b^2)/(2*(c^2+d^2)^(1/2
)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2)
)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*(c^2+d^2)^(1/2)*b-1/4/f/(a^2+b^2)/d*ln((
c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2
)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b*c-1/4/f/(a^2+b^2)/d*ln((c+d*tan(
f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))
*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a-1/4/f/(a^2+b^2)/d*ln(d*t
an(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^
(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*a-1/4/f/(a^2+b^2)/d*
ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2
+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a*c-1/4/f/(a^2+b^2)*ln(d*tan(f
*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2
))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a+1/4/f/(a^2+b^2)*ln((c+d*tan(f*x+e))^(1
/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2
+d^2)^(1/2)+2*c)^(1/2)*b-1/4/f/(a^2+b^2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+
2*c)^(1/2)*b+1/4/f/(a^2+b^2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2
*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a
+1/4/f/(a^2+b^2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [B] time = 36.22, size = 62245, normalized size = 266.00

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x)),x)
```

```
[Out] atan(((((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f
^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8
*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 +
8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4
))/(b*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2
*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f
^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b
*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 +
 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*
c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^
2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4
 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 1
6*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^
2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(1
6*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1
/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^
2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 -
 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^
2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2
*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a
^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2)
)/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 -
(C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C
^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 +
2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^
2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24
*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10
*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*
b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 +
 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^
2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 +
32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^
2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x
))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^
2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 +
 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*((((8*C^2*a^2*c*f^2 -
8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 +
 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 -
8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((
32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^
5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*
c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6
*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5)
+ (32*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16
*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2
```

```
*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16
*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8
*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4
 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 +
 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^
3*c*d^9*f^4))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d
*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))
^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4
 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2
*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a
^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*
c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a
^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 +
4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*
d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*
((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 +
C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^
2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f
^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7
*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^
4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^
3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9
*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5
*b^2*c^3*d^9*f^2))/(b*f^5))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*
a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*
f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4
*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(C
^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^
4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*
b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c
*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4
 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d
*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(4*C*a*b
^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*
f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4
- 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^
4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) - (32*(c +
d*tan(e + f*x))^(1/2)*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*
f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^
(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 +
 b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 -
 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b
^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*
d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4
))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 -
```

(C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*
C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 +
2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11
*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11
*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2
+ 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^
10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b
^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2
*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*((((8*C^2*a
^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(1
6*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b
^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)
+ (32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f
^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^
2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c
^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^
3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^
9*f^2))/(b*f^5))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^
2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2)
- 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*
f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12
- 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d
^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 -
4*C^4*a^5*b*c^3*d^9))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C
^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b
^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(
a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(4*C*a*b^8*d^11*f^4 -
4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*
c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*
c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^
5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x)
)^(1/2)*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^
4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*
a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a
^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^
10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4
+ 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a
^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((
((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^
4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2
+ 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4
)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a
^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*
a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7
*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^

2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2
 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c
^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C
^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16
*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C
^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3
*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c
^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5
*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 +
9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*
d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5
))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2
 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*C^2*a^2*c
*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^
2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d
^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4
*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^
3*d^9))/(b*f^4))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^
2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2)
 - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*
f^4 + 2*a^2*b^2*f^4)))^(1/2) - (64*(C^5*a^5*d^13 - C^5*a^3*b^2*d^13 + C^5*a
^5*c^2*d^11 + 2*C^5*a^2*b^3*c^3*d^10 + C^5*a^2*b^3*c^5*d^8 - 2*C^5*a^3*b^2*
c^2*d^11 - C^5*a^3*b^2*c^4*d^9 - C^5*a^4*b*c*d^12 + C^5*a^2*b^3*c*d^12 - C^
5*a^4*b*c^3*d^10))/(b*f^5)))*((((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2
*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2
*f^4))^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(16*(a^
4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i - atan(((((32*(15*B^3*a^3*b^3*d
^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 +
 B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 -
22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10
*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3
*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^1
1*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4
 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^
4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8
*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 - (32*(c + d*ta
n(e + f*x))^(1/2)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)
^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2
) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4
*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a
^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2
*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^
4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4
)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2
+ B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*

```
f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2
*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4
*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*
B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12
*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^
8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5
*b^2*c^2*d^9*f^2))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*
d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)
)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4
 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2
+ 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32
*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)
/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))
^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d
^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^
9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*((((8*B^2*a^2*c*f^2
- 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4
 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2
+ 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((
32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*
B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*
a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 -
4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*
d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - ((
(32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4
*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*
B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 -
 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))
/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2
 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 3
2*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2
)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2
*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*
f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^
4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7
*b^2*c*d^9*f^4))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*
f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^
(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 +
 b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*
a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2
*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*
b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 1
2*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*
d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2
*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f
```

^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b
*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*((((8*B^2*a^2*c*f^
2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f
^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^
2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*
(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4
*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 +
4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*
((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 +
B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^
2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f
^4)))^(1/2)*1i)/((((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3
*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c
^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a
^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 +
10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^
3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*
a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5
*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a
^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*
B*a*b^7*c*d^10*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c*f
^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*
f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f
^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b
^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f
^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 -
8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b
^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c
*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4
+ 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d
f^2)/(16(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e +
f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*
c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*
d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^
4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 -
10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*B^2*a
^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(1
6*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b
^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)
)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2
+ B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*
f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2
f^4)))^(1/2) + (32(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^
2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^
4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^

4*b*c^2*d^10))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (64*(B^5*a*b^3*c*d^12 - 3*B^5*a^2*b^2*c^2*d^11 - 2*B^5*a^2*b^2*c^4*d^9 - B^5*a^2*b^2*d^13 + B^5*a^3*b*c*d^12 + 2*B^5*a*b^3*c^3*d^10 + B^5*a*b^3*c^5*d^8 + B^5*a^3*b*c^3*d^10))/f^5))*((((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*B^2*a

^2*c*f^2 - 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^
2*b^2*f^4)))^(1/2)*2i - atan(((((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*
d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 +
6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*
f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*
c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B
^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^
11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*
f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^
8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2
*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*(-((
(8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4
d^2)(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 +
4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)
))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*
a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*
c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*
f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*B^2*a^2*c*f^
2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f
^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^
2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*
(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^
2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 -
8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*
f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^
4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f
^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c
^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2
*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*
b^2*f^4)))^(1/2))*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2
)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/
2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^
4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^
12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^
3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*
c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2
+ 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 3
2*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2
)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((32*(15*B^3*a^3*b^
3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^
2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2
- 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d
^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*
a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*
d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*

f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9

*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c
^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^
4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*
B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*
b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*
(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2
)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^
2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 +
 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^
10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^
4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*B^2*a^2*c*f^2
- 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4
 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2
- 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*(-(((8*
B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^
2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*
B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^
(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 +
 B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^
3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*
d^10))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4
- (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4
*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4
+ 2*a^2*b^2*f^4)))^(1/2) + (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9
*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B
^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2
 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3
*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*
a^4*b^2*c^3*d^9*f^2))/f^5 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*
f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4
 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f
^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^
9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*
B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^
2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*
B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^
(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6
*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2
*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4
 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*B^2*a^2*c*f^2 -
 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4
+ 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 -
 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c
+ d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 -
 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*

B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2))*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4)*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (64*(B^5*a*b^3*c*d^12 - 3*B^5*a^2*b^2*c^2*d^11 - 2*B^5*a^2*b^2*c^4*d^9 - B^5*a^2*b^2*d^13 + B^5*a^3*b*c*d^12 + 2*B^5*a*b^3*c^3*d^10 + B^5*a*b^3*c^5*d^8 + B^5*a^3*b*c^3*d^10))/f^5))*(-(((8*B^2*a^2*c*f^2 - 8*B^2*b^2*c*f^2 + 16*B^2*a*b*d*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i + atan((((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)

/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12
*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 -
12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f
^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5
*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12
*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*(-(((8*C^2
*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*
(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2
*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/
2) - (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*
a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^1
0 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f
^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*
c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^
2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2
*b^2*f^4)))^(1/2)*1i - (((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8
*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^
8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b
^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^
5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2
*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*
a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2
*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(
16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d
^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*
f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24
*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 -
8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 +
16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 +
8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c +
d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10
*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C
^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^
2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2
- 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2
*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a
^6*b^2*c^3*d^8*f^2))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C
^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b
^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(
a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12*f^2 -
C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3
*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 1
2*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d
^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a
^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*(-(((8*C^2*a^2*c

*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*(-(((8*

C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*(-(((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (64*(C^5*a^5*d^13 - C^5*a^3*b^2*d^13 + C^5*a^5*c^2*d^11 + 2*C^5*a^2*b^3*c^3*d^10 + C^5*a^2*b^3*c^5*d^8 - 2*C^5*a^3*b^2*c^2*d^11 - C^5*a^3*b^2*c^4*d^9 - C^5*a^4*b*c*d^12 + C^5*a^2*b^3*c*d^12 - C^5*a^4*b*c^3*d^10))/(b*f^5)))*(-(

((8*C^2*a^2*c*f^2 - 8*C^2*b^2*c*f^2 + 16*C^2*a*b*d*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i + atan((((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4

*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a
^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2
*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^
4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4
)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2
+ A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*
f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2
f^4)))^(1/2) - (32(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2
*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*
A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 -
10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^
2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((
((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^
4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2
+ 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4
)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6
*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*
b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A
^3*a^4*b^2*c^2*d^10*f^2))/f^5)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A
^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b
^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(
a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)
*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2
*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*((((8*A^2*a^2*c*f^2
- 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^
4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2
- 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/(((
(((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 +
12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*
A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 +
24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9
f^4))/f^5 - (32(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2
*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f
^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b
*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 +
16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^
2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*
d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 +
8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*
a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*
f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4
*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(2
0*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 -
14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 3
6*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d

```
^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5
*b^2*c^2*d^9*f^2))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*
d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)
)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4
 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^
4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*
b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A
^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*((((8*A^2*a^2*c*f^
2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f
^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^
2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*
(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c
^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9)
)/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4
*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a
^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^
2*b^2*f^4)))^(1/2) - (64*(A^5*b^4*c^3*d^10 - A^5*a*b^3*d^13 + A^5*b^4*c*d^1
2 - A^5*a*b^3*c^2*d^11))/f^5 + (((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^
10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f
^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10
*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c
^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2
)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2
+ A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*
f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2
*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4
- 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4
*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c
*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*
c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a
^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*
c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) -
(32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^1
1*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f
^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*
c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^
2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*((((8*A^2*a^2*c*f^
2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f
^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^
2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*
(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 +
3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2
+ 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d
^10*f^2))/f^5)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/
4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) -
```

4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)))*((((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i + atan((((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c* d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b

```
^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f
^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(
1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 +
b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a
^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*
a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a
^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2
+ 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^
2*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)
^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2
) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4
*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*
d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*
d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2
*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*(-(((8*A^2*a^2*c*f^2 - 8
*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 +
16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8
*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c +
d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^
8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4
)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2
 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c
*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^
2*f^4)))^(1/2) - (64*(A^5*b^4*c^3*d^10 - A^5*a*b^3*d^13 + A^5*b^4*c*d^12 -
A^5*a*b^3*c^2*d^11))/f^5 + (((((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f
^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 +
 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4
 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d
^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(-
(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A
^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2
 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^
4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 1
6*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^
5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^
9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c*
f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4
*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*
f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (3
2*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*
f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2
 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*
d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*
a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c*f^2
 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^
```

4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)))*(-(((8*A^2*a^2*c*f^2 - 8*A^2*b^2*c*f^2 + 16*A^2*a*b*d*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i + (atan((((C^2*a^5*d - C^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5) + ((C^2*a^5*d - C^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) - (32*(C^2*a^5*d - C^2*a^4*b*c)*(c + d*tan(e + f*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4*(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2))))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2)))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*

b^2*c^4*d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*1i)/(-(C^
2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) - ((C
^2*a^5*d - C^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(15*C^3*a^4*b^3*d^
12*f^2 - C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2
- 12*C^3*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11
*f^2 - 12*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b
^5*c^4*d^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 -
12*C^3*a^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5) + ((C^2*a
^5*d - C^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(4*C*a*b^8*d^11*f^4 -
4*C*b^9*c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*
c^3*d^8*f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*
c*d^10*f^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^
5*c^3*d^8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(C^2*a^5*d - C^2*a^
4*b*c)*(c + d*tan(e + f*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 -
16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b
^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*
d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4
))/(b*f^4*(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^
2))^(1/2))))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3
f^2))^(1/2) + (32(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^
2*a^5*b^3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C
^2*a^7*b*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*
b^7*c^2*d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24
*C^2*a^6*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*
f^2 + 4*C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^
3*c^2*d^9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^
4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2)))/(-(C^2*a^5*d - C^2*
a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) + (32*(c + d*tan(e
+ f*x))^(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b
^6*c^2*d^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*
d^8 + 4*C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4))*1i)/(-(C^2*a^5*d
- C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2))/(((C^2*a^5*d
- C^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(15*C^3*a^4*b^3*d^12*f^2 -
C^3*a^2*b^5*d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3
*a^6*b*d^12*f^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 1
2*C^3*a^6*b*c^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d
^8*f^2 - 24*C^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a
^4*b^3*c^4*d^8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5) + ((C^2*a^5*d - C
^2*a^4*b*c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*
c*d^10*f^4 + 8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*
f^4 + 4*C*a*b^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f
^4 - 8*C*a^2*b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^
8*f^4 + 4*C*a^5*b^4*c^2*d^9*f^4))/(b*f^5) - (32*(C^2*a^5*d - C^2*a^4*b*c)*(
c + d*tan(e + f*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*
b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d

^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4
+ 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^
4*(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2
))))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(
1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^
3*d^11*f^2 - 10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b
*d^11*f^2 + 8*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*
d^9*f^2 + 12*C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6
*b^2*c*d^10*f^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*
C^2*a^3*b^5*c^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^
9*f^2 + 8*C^2*a^6*b^2*c^3*d^8*f^2))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^4*b*c)*(
b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2)))/(-(C^2*a^5*d - C^2*a^4*b*c)
*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^
(1/2)*(C^4*b^6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d
^10 + C^4*b^6*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*
C^4*a^5*b*c*d^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^4*b
c)(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) - (64*(C^5*a^5*d^13 - C^
5*a^3*b^2*d^13 + C^5*a^5*c^2*d^11 + 2*C^5*a^2*b^3*c^3*d^10 + C^5*a^2*b^3*c^
5*d^8 - 2*C^5*a^3*b^2*c^2*d^11 - C^5*a^3*b^2*c^4*d^9 - C^5*a^4*b*c*d^12 + C
^5*a^2*b^3*c*d^12 - C^5*a^4*b*c^3*d^10))/(b*f^5) + ((C^2*a^5*d - C^2*a^4*b*
c)*(((C^2*a^5*d - C^2*a^4*b*c)*((32*(15*C^3*a^4*b^3*d^12*f^2 - C^3*a^2*b^5*
d^12*f^2 + C^3*b^7*c^2*d^10*f^2 + C^3*b^7*c^4*d^8*f^2 - 12*C^3*a^6*b*d^12*f
^2 - 24*C^3*a^3*b^4*c*d^11*f^2 + 24*C^3*a^5*b^2*c*d^11*f^2 - 12*C^3*a^6*b*c
^2*d^10*f^2 + 8*C^3*a^2*b^5*c^2*d^10*f^2 + 9*C^3*a^2*b^5*c^4*d^8*f^2 - 24*C
^3*a^3*b^4*c^3*d^9*f^2 + 3*C^3*a^4*b^3*c^2*d^10*f^2 - 12*C^3*a^4*b^3*c^4*d^
8*f^2 + 24*C^3*a^5*b^2*c^3*d^9*f^2))/(b*f^5) + ((C^2*a^5*d - C^2*a^4*b*c)*(
((C^2*a^5*d - C^2*a^4*b*c)*((32*(4*C*a*b^8*d^11*f^4 - 4*C*b^9*c*d^10*f^4 +
8*C*a^3*b^6*d^11*f^4 + 4*C*a^5*b^4*d^11*f^4 - 4*C*b^9*c^3*d^8*f^4 + 4*C*a*b
^8*c^2*d^9*f^4 - 8*C*a^2*b^7*c*d^10*f^4 - 4*C*a^4*b^5*c*d^10*f^4 - 8*C*a^2*
b^7*c^3*d^8*f^4 + 8*C*a^3*b^6*c^2*d^9*f^4 - 4*C*a^4*b^5*c^3*d^8*f^4 + 4*C*a
^5*b^4*c^2*d^9*f^4))/(b*f^5) + (32*(C^2*a^5*d - C^2*a^4*b*c)*(c + d*tan(e +
f*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4
- 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^
4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*
c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4*(-(C^2*a^5*
d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2))))/(-(C^2*a
^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) + (32*(c
+ d*tan(e + f*x))^(1/2)*(14*C^2*a*b^7*d^11*f^2 - 2*C^2*a^5*b^3*d^11*f^2 -
10*C^2*b^8*c^3*d^8*f^2 - 4*C^2*a^3*b^5*d^11*f^2 - 16*C^2*a^7*b*d^11*f^2 + 8
*C^2*a^8*c*d^10*f^2 - 6*C^2*b^8*c*d^10*f^2 + 18*C^2*a*b^7*c^2*d^9*f^2 + 12*
C^2*a^2*b^6*c*d^10*f^2 + 2*C^2*a^4*b^4*c*d^10*f^2 + 24*C^2*a^6*b^2*c*d^10*f
^2 - 16*C^2*a^7*b*c^2*d^9*f^2 + 4*C^2*a^2*b^6*c^3*d^8*f^2 + 4*C^2*a^3*b^5*c
^2*d^9*f^2 - 10*C^2*a^4*b^4*c^3*d^8*f^2 + 2*C^2*a^5*b^3*c^2*d^9*f^2 + 8*C^2
*a^6*b^2*c^3*d^8*f^2))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a
^2*b^5*f^2 + a^4*b^3*f^2))^(1/2)))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2

*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(C^4*b^
6*d^12 - 2*C^4*a^6*d^12 + 2*C^4*a^6*c^2*d^10 + 2*C^4*b^6*c^2*d^10 + C^4*b^6
*c^4*d^8 - 2*C^4*a^4*b^2*c^2*d^10 + 2*C^4*a^4*b^2*c^4*d^8 + 4*C^4*a^5*b*c*d
^11 - 4*C^4*a^5*b*c^3*d^9))/(b*f^4)))/(-(C^2*a^5*d - C^2*a^4*b*c)*(b^7*f^2
+ 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2)))*(C^2*a^5*d - C^2*a^4*b*c)*2i)/(-(C^
2*a^5*d - C^2*a^4*b*c)*(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2))^(1/2) + (at
an((((((32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2
*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*
c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a
^4*b^2*c^2*d^10*f^2))/f^5 + ((((A^2*b^2*c - A^2*a*b*d)*((32*(12*A*a*b^7*d^1
1*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4
 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4
- 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9
*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 - (32*(A^2
*b^2*c - A^2*a*b*d)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^
7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4
 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 +
 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^
2*c*d^9*f^4))/(f^4*((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*
f^2))^(1/2))))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2)
)^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a
^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^
7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^
2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*
f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(A^2*b^2
*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^
2))^(1/2))*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4
*f^2 + 2*a^2*b^2*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12
 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*
a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(A^2*b^2*c - A^2*a*b*d)*1i)/((A^2
*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) - (((((32*(1
3*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*
A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 +
12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^1
0*f^2))/f^5 + ((((A^2*b^2*c - A^2*a*b*d)*((32*(12*A*a*b^7*d^11*f^4 - 12*A*b
^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^
3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4
*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^
4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(A^2*b^2*c - A^2*a
*b*d)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 1
6*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*
c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9
*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/
(f^4*((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))))
/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) - (32*

(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(A^2*b^2*c - A^2*a*b*d)*1i)/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))/((((((32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5 + ((((A^2*b^2*c - A^2*a*b*d)*((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 - (32*(A^2*b^2*c - A^2*a*b*d)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9))/f^4)*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) - (64*(A^5*b^4*c^3*d^10 - A^5*a*b^3*d^13 + A^5*b^4*c*d^12 - A^5*a*b^3*c^2*d^11))/f^5 + (((((32*(13*A^3*a^2*b^4*d^12*f^2 + A^3*a^4*b^2*d^12*f^2 + 3*A^3*b^6*c^2*d^10*f^2 + 3*A^3*b^6*c^4*d^8*f^2 - 16*A^3*a*b^5*c*d^11*f^2 - 16*A^3*a*b^5*c^3*d^9*f^2 + 12*A^3*a^2*b^4*c^2*d^10*f^2 - A^3*a^2*b^4*c^4*d^8*f^2 + A^3*a^4*b^2*c^2*d^10*f^2))/f^5 + ((((A^2*b^2*c - A^2*a*b*d)*((32*(12*A*a*b^7*d^11*f^4 - 12*A*b^8*c*d^10*f^4 + 24*A*a^3*b^5*d^11*f^4 + 12*A*a^5*b^3*d^11*f^4 - 12*A*b^8*c^3*d^8*f^4 + 12*A*a*b^7*c^2*d^9*f^4 - 24*A*a^2*b^6*c*d^10*f^4 - 12*A*a^4*b^4*c*d^10*f^4 - 24*A*a^2*b^6*c^3*d^8*f^4 + 24*A*a^3*b^5*c^2*d^9*f^4 - 12*A*a^4*b^4*c^3*d^8*f^4 + 12*A*a^5*b^3*c^2*d^9*f^4))/f^5 + (32*(A^2*b^2*c - A^2*a*b*d)*(c + d*tan(e +

```
f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 -
16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b
^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d
^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*((A^2*b^2*c - A^
2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))))/((A^2*b^2*c - A^2*a*
b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) - (32*(c + d*tan(e + f*x))^
(1/2)*(20*A^2*a^3*b^4*d^11*f^2 + 2*A^2*a^5*b^2*d^11*f^2 + 18*A^2*b^7*c^3*d^
8*f^2 - 14*A^2*a*b^6*d^11*f^2 + 6*A^2*b^7*c*d^10*f^2 - 18*A^2*a*b^6*c^2*d^9
*f^2 - 36*A^2*a^2*b^5*c*d^10*f^2 - 10*A^2*a^4*b^3*c*d^10*f^2 - 12*A^2*a^2*b
^5*c^3*d^8*f^2 + 12*A^2*a^3*b^4*c^2*d^9*f^2 + 2*A^2*a^4*b^3*c^3*d^8*f^2 - 2
*A^2*a^5*b^2*c^2*d^9*f^2))/f^4)*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*
a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2))*(A^2*b^2*c - A^2*a*b*d))
/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) + (32*
(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^12 - 2*A^4*a^2*b^3*d^12 + 3*A^4*b^5*c
^4*d^8 + 2*A^4*a^2*b^3*c^2*d^10 + 4*A^4*a*b^4*c*d^11 - 4*A^4*a*b^4*c^3*d^9)
)/f^4)*(A^2*b^2*c - A^2*a*b*d))/((A^2*b^2*c - A^2*a*b*d)*(a^4*f^2 + b^4*f^2
 + 2*a^2*b^2*f^2))^(1/2)))*(A^2*b^2*c - A^2*a*b*d)*2i)/((A^2*b^2*c - A^2*a*
b*d)*(a^4*f^2 + b^4*f^2 + 2*a^2*b^2*f^2))^(1/2) + (2*C*(c + d*tan(e + f*x))
^(1/2))/(b*f) - (atan(((((32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B
^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^
10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 -
 2*B^4*a^4*b*c^2*d^10))/f^4 + (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*
d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 +
6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*
f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*
c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B
^3*a^4*b^2*c^3*d^9*f^2))/f^5 - ((-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*
b*f^2 + 2*a^2*b^3*f^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b
^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^
6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c
^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2
*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f
^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4 + (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a
^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^
5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b
^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^
6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 - (32*(-(B^2*a^3*d - B^2*a^2
*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/
2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^
3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^
8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 +
24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2))*(-(B^2*
a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(
a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2
 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d -
```

B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*1i)/(b*f^2*(a^2 + b^2)^2) + (((32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4 - (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 + ((-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*1i)/(b*f^2*(a^2 + b^2)^2))/((64*(B^5*a*b^3*c*d^12 - 3*B^5*a^2*b^2*c^2*d^11 - 2*B^5*a^2*b^2*c^4*d^9 - B^5*a^2*b^2*d^13 + B^5*a^3*b*c*d^12 + 2*B^5*a*b^3*c^3*d^10 + B^5*a*b^3*c^5*d^8 + B^5*a^3*b*c^3*d^10))/f^5 + (((32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4 + (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 - ((-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3* c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4 + (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 - (32*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2) - (((32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^5*d^12 + 2*B^4*b^5*c^2*d^10 + B^4*b^5*c^4*d^8 + 2*B^4*a^4*b*d^12 + 2*B^4*a^2*b^3*c^2*d^10 - 2*B^4*a^2*b^3*c^4*d^8 + 4*B^4*a^3*b^2*c^3*d^9 - 4*B^4*a^3*b^2*c*d^11 - 2*B^4*a^4*b*c^2*d^10))/f^4 - (((32*(15*B^3*a^3*b^3*d^12*f^2 + B^3*b^6*c^3*d^9*f^2 - B^3*a*b^5*d^12*f^2 - 4*B^3*a^5*b*d^12*f^2 + B^3*b^6*c*d^11*f^2 + 6*B^3*a*b^5*c^2*d^10*f^2 + 7*B^3*a*b^5*c^4*d^8*f^2 - 22*B^3*a^2*b^4*c*d^11*f^2 + 9*B^3*a^4*b^2*c*d^11*f^2 - 4*B^3*a^5*b*c^2*d^10*f^2 - 22*B^3*a^2*b^4*c^3*d^9*f^2 + 10*B^3*a^3*b^3*c^2*d^10*f^2 - 5*B^3*a^3*b^3*c^4*d^8*f^2 + 9*B^3*a^4*b^2*c^3*d^9*f^2))/f^5 + ((-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(14*B^2*a^5*b^2*d^11*f^2 - 4*B^2*a^3*b^4*d^11*f^2 - 10*B^2*b^7*c^3*d^8*f^2 + 14*B^2*a*b^6*d^11*f^2 - 6*B^2*b^7*c*d^10*f^2 - 8*B^2*a^6*b*c*d^10*f^2 + 18*B^2*a*b^6*c^2*d^9*f^2 + 12*B^2*a^2*b^5*c*d^10*f^2 - 22*B^2*a^4*b^3*c*d^10*f^2 + 12*B^2*a^2*b^5*c^3*d^8*f^2 + 4*B^2*a^3*b^4*c^2*d^9*f^2 - 10*B^2*a^4*b^3*c^3*d^8*f^2 + 18*B^2*a^5*b^2*c^2*d^9*f^2))/f^4 - (((32*(4*B*a^2*b^6*d^11*f^4 + 8*B*a^4*b^4*d^11*f^4 + 4*B*a^6*b^2*d^11*f^4 - 4*B*a*b^7*c^3*d^8*f^4 - 8*B*a^3*b^5*c*d^10*f^4 - 4*B*a^5*b^3*c*d^10*f^4 + 4*B*a^2*b^6*c^2*d^9*f^4 - 8*B*a^3*b^5*c^3*d^8*f^4 + 8*B*a^4*b^4*c^2*d^9*f^4 - 4*B*a^5*b^3*c^3*d^8*f^4 + 4*B*a^6*b^2*c^2*d^9*f^4 - 4*B*a*b^7*c*d^10*f^4))/f^5 + (32*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2))/(b*f^2*(a^2 + b^2)^2)))*(-(B^2*a^3*d - B^2*a^2*b*c)*(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2))^(1/2)*2i)/(b*f^2*(a^2 + b^2)^2)

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c + d\tan\left(e + fx\right)}\left(A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)\right)}{a + b\tan\left(e + fx\right)}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e)),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x)), x)
```

3.95 $\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$

Optimal. Leaf size=317

$$-\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{bf\left(a^2+b^2\right)(a+b\tan(e+fx))}-\frac{\left(a^4Cd+a^3bBd-a^2b^2(3Ad+2Bc-5Cd)+ab^3(4Ac-3Bd-4cC)+\right.}{b^{3/2}f\left(a^2+b^2\right)^2\sqrt{bc-ad}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))*(c-I*d)^(1/2)/(a
-I*b)^2/f-(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))*(c+I*d)
^(1/2)/(a+I*b)^2/f-(a^3*b*B*d+a^4*C*d+b^4*(A*d+2*B*c)+a*b^3*(4*A*c-3*B*d-4*
C*c)-a^2*b^2*(3*A*d+2*B*c-5*C*d))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-
a*d+b*c)^(1/2))/b^(3/2)/(a^2+b^2)^2/f/(-a*d+b*c)^(1/2)-(A*b^2-a*(B*b-C*a))*
(c+d*tan(f*x+e))^(1/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] time = 1.44, antiderivative size = 317, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3645, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{bf\left(a^2+b^2\right)(a+b\tan(e+fx))}-\frac{\left(-a^2b^2(3Ad+2Bc-5Cd)+a^3bBd+a^4Cd+ab^3(4Ac-3Bd-4cC)\right.}{b^{3/2}f\left(a^2+b^2\right)^2\sqrt{bc-ad}}$$

Antiderivative was successfully verified.

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x])^2,x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c -
I*d]])/((a - I*b)^2*f)) - ((B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((a + I*b)^2*f) - ((a^3*b*B*d + a^4*C*d + b^
4*(2*B*c + A*d) + a*b^3*(4*A*c - 4*c*C - 3*B*d) - a^2*b^2*(2*B*c + 3*A*d -
5*C*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(b^(3/
2)*(a^2 + b^2)^2*Sqrt[b*c - a*d]*f) - ((A*b^2 - a*(b*B - a*C))*Sqrt[c + d*T
an[e + f*x]])/(b*(a^2 + b^2)*f*(a + b*Tan[e + f*x]))
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
```

```
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}+\frac{\int}{}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}+\frac{\int}{}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}+\frac{((}{}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}-\frac{(i}{}$$

$$= -\frac{\left(a^3bBd+a^4Cd+b^4(2Bc+Ad)+ab^3(4Ac-\right.}{b^3}$$

$$= -\frac{(B+i(A-C))\sqrt{c-id}\,\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right.}{(a-ib)^2f}$$

Mathematica [B] time = 6.39, size = 764, normalized size = 2.41

$$-\frac{2C\sqrt{c+d\tan(e+fx)}}{bf(a+b\tan(e+fx))}-\frac{2\left(-\frac{\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}b^2(-aCd-Abc+2bcC)-a\left(-\frac{1}{2}a(-aCd-bBd+bcC)-\frac{1}{2}b^2(d(A-C)+Bc)\right)\right)}{f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))}-\frac{2\sqrt{bc-ad}\left(-\frac{1}{4}a^2d(bc\right.}{}\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x])^2,x]
```

```
[Out] (-2*C*Sqrt[c + d*Tan[e + f*x]])/(b*f*(a + b*Tan[e + f*x])) - (2*(-((((I*Sqr
t[c - I*d]*((b*(b*c - a*d)*(a^2*(A*c - c*C - B*d) - b^2*(A*c - c*C - B*d) +
 2*a*b*(B*c + (A - C)*d)))/2 + (I/2)*b*(b*c - a*d)*(2*a*b*(A*c - c*C - B*d)
 - a^2*(B*c + (A - C)*d) + b^2*(B*c + (A - C)*d)))*ArcTanh[Sqrt[c + d*Tan[e
 + f*x]]/Sqrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*((b*(b*c - a*d)*
(a^2*(A*c - c*C - B*d) - b^2*(A*c - c*C - B*d) + 2*a*b*(B*c + (A - C)*d)))/
2 - (I/2)*b*(b*c - a*d)*(2*a*b*(A*c - c*C - B*d) - a^2*(B*c + (A - C)*d) +
b^2*(B*c + (A - C)*d)))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((
-c - I*d)*f))/(a^2 + b^2) + (2*Sqrt[b*c - a*d]*(-1/4*(a^2*(A*b^2 - a*b*B -
a^2*C - 2*b^2*C)*d*(b*c - a*d)) + (a*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C
 - a*A*d - b*B*d + a*C*d))/2 + (b^2*(b*c - a*d)*(a^2*C*d + b^2*(2*B*c + A*d
) + a*b*(2*A*c - 2*c*C - B*d)))/4)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]
])/Sqrt[b*c - a*d]])/(Sqrt[b]*(a^2 + b^2)*(-(b*c) + a*d)*f))/((a^2 + b^2)*(
b*c - a*d))) - (((b^2*(-(A*b*c) + 2*b*c*C - a*C*d))/2 - a*(-1/2*(b^2*(B*c +
 (A - C)*d)) - (a*(b*c*C - b*B*d - a*C*d))/2))*Sqrt[c + d*Tan[e + f*x]])/((
a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]))))/b
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.87, size = 5778, normalized size = 18.23

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [B] time = 45.42, size = 138318, normalized size = 436.33

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^2,x)
```

```
[Out] atan(((((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^
6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^
3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 12
4*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d
^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^
10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f
^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4
*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^
2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b
^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))
/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8
*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 7
20*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 -
16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 -
224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^1
0*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*
c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*
```

B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8
*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6
*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B
*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5
+ 4*a^6*b^2*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*
B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*
f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 +
96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2
 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8
*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32
*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^
11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d
^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8
*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^
2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3
*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*
c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c
*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*
f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2
+ 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^
2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^
4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c
*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*
(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)
 - (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^
8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^1
1*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*
b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 12
0*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*
d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*
a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d
^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^
7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2
 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b
^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f
^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2
*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6
*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^
4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(
16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1
/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a
^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 +
16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B
^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 +
24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^

4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i - (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^1

2*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3
*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9
*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*
b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*
b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*
B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f
^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2)
- 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f
^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b
^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^
9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^
2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 6
0*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 +
100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*
c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^
2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*
d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*
a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f
^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11
*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a
^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 +
32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^
8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/
2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*
d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^
4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2
- 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^
4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4
 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^
3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4
 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(
e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12
- 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*
c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*
c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3
*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*
d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11
 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b
^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c
*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4
- (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^
4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2
*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^
8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i)/((((8*(1
56*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^

2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^
8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*
c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128
*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10
*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*
a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*
f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^
6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f
^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b
^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14
*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10
*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2
*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^1
3*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*
a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 -
224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^
3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*
a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4
+ 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2
*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f
^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2
 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B
^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^
4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b
^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^
4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^
4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 -
 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*
a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^
4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 4
8*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4
+ 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 5
60*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 11
2*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b
^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c
*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4
- (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^
4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2
*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^
8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d
*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 -
40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^
2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f
^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^1
1*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 14

8*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*
d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^
2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9
*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^1
0*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^
2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^
4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2
/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4
*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*
B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 +
b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*B
^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 -
48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 +
64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2
- 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^
2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2
f^4)))^(1/2) - (16(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*
b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B
^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^
3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a
^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^
7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6
*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10
))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(
((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f
^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f
^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c
*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a
^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^
6*b^2*f^4)))^(1/2) + (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f
^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d
^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*
c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*
B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11
*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^
2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f
^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*
b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2
+ 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^
2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*
b^2*f^5) + (((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^
12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*
b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14
*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*
B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 +

16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2
*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*
a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4
+ 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3
*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 +
6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2
*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48
*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64
*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 -
4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*
c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f
^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f
^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 -
160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a
^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 +
400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^
4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4
+ 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 +
336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/
(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8
*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2
- 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4
+ 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^
2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*
b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b
^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 -
168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*
f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^1
1*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c
^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112
*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^1
0*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^
2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*
f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*
b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2
))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(
((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f
^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f
^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c
*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a
^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^
6*b^2*f^4)))^(1/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d
*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)
*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^
4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2

*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(2*B^5*a^3*b^4*d^13 + 4*B^5*b^7*c^3*d^10 - 6*B^5*a*b^6*d^13 + 4*B^5*b^7*c*d^12 - 9*B^5*a^2*b^5*c^3*d^10 + 4*B^5*a^2*b^5*c^5*d^8 - 12*B^5*a^3*b^4*c^2*d^11 - 14*B^5*a^3*b^4*c^4*d^9 + 2*B^5*a^4*b^3*c^3*d^10 - 4*B^5*a^4*b^3*c^5*d^8 + 4*B^5*a^5*b^2*c^2*d^11 + 4*B^5*a^5*b^2*c^4*d^9 - B^5*a^6*b*c*d^12 + 6*B^5*a*b^6*c^4*d^9 - 13*B^5*a^2*b^5*c*d^12 + 6*B^5*a^4*b^3*c*d^12 - B^5*a^6*b*c^3*d^10))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5)))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2i - atan(((((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^1

2*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b
^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-
(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*
f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*
f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*
c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*
a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a
^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^1
4*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d
^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^
4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*
d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*
c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*
c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c
*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*
d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*
f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*
a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 +
16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*
C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 -
24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f
^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^
11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2
*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 6
0*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*
C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^
2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*
b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 +
116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*
b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2
+ 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^
5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/
(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-((
(8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^
2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^
4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*
f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^
2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6
*b^2*f^4)))^(1/2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*
f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*
(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4
))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*
a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(
2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12

- 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10 *c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c ^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^ 6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^ 3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c* d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^ 4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2* c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f ^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a ^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1 i - (((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7 *b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a *b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a* b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^ 4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^ 8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b ^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^ 2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f ^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9* c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2 *b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9* c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C *a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f ^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4* b^5*f^5 + 4*a^6*b^3*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c* f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^ 2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^ 6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b ^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/ (16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 16 0*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a ^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^1 6*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a ^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 1 6*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336 *a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*

a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*
f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^
2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 4
8*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 6
4*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 +
4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2
*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*
f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 1
28*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f
^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^
11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10
*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4
*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2
- 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^
12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^
2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b
^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 +
4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*
b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^
2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*
b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*
f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4
*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(1
6*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/
2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^
3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 1
6*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^
2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 2
4*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4
+ 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12
- C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^
4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C
^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^
4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*
C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C
^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*
a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f
^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2
- 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C
^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^
4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b
^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^
4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i)/((((8*(304*C
^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 -
148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2

- 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f
^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*
b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 -
16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9
*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 -
72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^
6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2
- 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a
^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12
*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b
^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*
b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 32
0*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f
^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^
11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 9
60*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d
^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^1
5*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6
*b^3*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4
*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/
4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*
b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C
^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 +
b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d
^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10
*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^
4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 +
624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8
*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*
d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9
*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*
f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4
+ 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8
*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c
*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 +
96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^
2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^
8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) +
(16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*
d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^
11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*
a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a
*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^
2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*
b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2

+ 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5 *b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4 *c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b ^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c *f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^ 4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2 *a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^ 8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*C^2* a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48* C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64* a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4 *C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c *f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^ 4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^1 2 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4* a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d ^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3 *d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2* d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5* f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3 *d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^ 2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2* f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C ^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f ^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (((8*(304*C^3*a^3*b^9*d^12*f^ 2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d ^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^ 12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11 *c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 5 44*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2* d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 12 8*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^ 2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 32 0*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3* c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6 *b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C* a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C *a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10 *f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^1 3*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 9

```
60*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d
^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11
*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^
9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (16*(c
 + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*
b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4
*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b
^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 1
6*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^
6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*
b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^1
0*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^
10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*
d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8
*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^1
7*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^1
1*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5
*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 +
 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 3
2*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c
^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 +
64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d
*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 +
4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e +
 f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2
*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 -
20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 -
4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2
+ 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^
8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 4
4*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11
*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 +
 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c
^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*
C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5
*f^4 + 4*a^6*b^3*f^4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^
3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d
^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2
*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*
C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*
f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*
b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)
^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a
^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 1
6*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4
```

+ b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(
c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*
d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4
*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*
c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b
^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*
b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b
*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c
*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^
4))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^
3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 1
6*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^
2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 2
4*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4
+ 4*a^6*b^2*f^4)))^(1/2) - (16*(C^5*a^8*d^13 + 10*C^5*a^2*b^6*d^13 + 27*C^
5*a^4*b^4*d^13 + 10*C^5*a^6*b^2*d^13 + C^5*a^8*c^2*d^11 + 36*C^5*a^2*b^6*c^
2*d^11 + 26*C^5*a^2*b^6*c^4*d^9 - 40*C^5*a^3*b^5*c^3*d^10 + 29*C^5*a^4*b^4*
c^2*d^11 + 2*C^5*a^4*b^4*c^4*d^9 - 8*C^5*a^5*b^3*c^3*d^10 + 10*C^5*a^6*b^2*
c^2*d^11 - 8*C^5*a*b^7*c*d^12 - 16*C^5*a*b^7*c^3*d^10 - 8*C^5*a*b^7*c^5*d^8
- 40*C^5*a^3*b^5*c*d^12 - 8*C^5*a^5*b^3*c*d^12))/(b^9*f^5 + a^8*b*f^5 + 4*
a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5)))*(-(((8*C^2*a^4*c*f^2 + 8*C^2
*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2
)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*
a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 -
16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^
4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2i - a
tan(((((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7
*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a
*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A
^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*
f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b
^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 -
168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b
^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 -
4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^
4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4
- 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4
- 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4
+ 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*
d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b
^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 -
320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*
d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^1
1*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a
^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (16*(c

+ d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^

10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60
*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 +
4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2
+ 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*
a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(
1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*
b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*
a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i - (((8*(128*A^3*a^3*b^8*d^12*f^2 +
24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2
 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^
2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^
9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A
^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d
^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^
3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d
^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5
 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^
15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b
^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*
c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a
^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 +
64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d
^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7
*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 -
288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*
d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a
^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*
c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*
a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*
b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2
*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2
)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))
^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 +
160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*
a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^
15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*
a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 1
6*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336
*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*
a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*
f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a
^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A
^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a
^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*
A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*

f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A ^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4 *A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^ 10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a ^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f ^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a ^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^ 2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b ^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^ 6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f ^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4* f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a *b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8* f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*A^2*a^4 *c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2 *a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2 *b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^ 2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^ 2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^ 12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4* b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3 *b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a ^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8 *c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f ^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2* c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f ^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a ^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1 i)/((16*(A^5*b^7*d^13 - 9*A^5*a^4*b^3*d^13 + 3*A^5*b^7*c^2*d^11 + 2*A^5*b^7 *c^4*d^9 - 22*A^5*a^2*b^5*c^2*d^11 - 22*A^5*a^2*b^5*c^4*d^9 + 24*A^5*a^3*b^ 4*c^3*d^10 - 9*A^5*a^4*b^3*c^2*d^11 + 8*A^5*a*b^6*c^3*d^10 + 8*A^5*a*b^6*c^ 5*d^8 + 24*A^5*a^3*b^4*c*d^12))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4* b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6* d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11* c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^ 10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^ 11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3 *a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d

^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3
*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*
a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 9
6*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 2
88*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 6
4*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*
f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c
*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*
a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^
4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*
c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*
a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*
a^6*b^2*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b
^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^
2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^
4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16
*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4
+ b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17
*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^
10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f
^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4
+ 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8
*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*
d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9
*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*
f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 +
4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^
2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^
2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96
*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 +
16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f
^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16
*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11
*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d
^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d
^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2
*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f
^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^
3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*
f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*
b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2
- 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b
^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b
^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*
d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^

2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16
*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6
*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*A^2*a^4*c*f^2 + 8*A^2*
b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)
^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a
^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 1
6*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4
 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(
c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4
*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 6
3*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 -
123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9
+ 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A
^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*
f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2
 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A
^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^
4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b
^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^
4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (((8*(128*A^3*
a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A
^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20
*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f
^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b
^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 12
8*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^
2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 16
0*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*
d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f
^5) + (((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^1
1*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^
11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c
*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^
9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 3
20*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*
d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9
*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 -
 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 +
4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(
1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^
3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 1
6*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^
2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 2
4*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4
 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a

^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^

2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6
*b^2*f^4)))^(1/2)))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*
f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*
(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4
))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*
a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2i - atan(((((8*(128*A^3*a^3*b^8*d
^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^
2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11
*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*
A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11
*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3
*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^
2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7
*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2)
)/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((
8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 4
80*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 +
32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4
 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d
^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*
b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 +
 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*
d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^1
2*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6
*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-((
(8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^
2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^
4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*
f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^
2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6
*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*
d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10
*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 +
 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8
*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*
d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^
9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9
*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*
f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*
(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*
d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^
8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^
4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^
2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4

*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^1
1*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*
d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^1
2*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a
^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2
 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^
11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2
 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^
6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 +
4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^
4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 +
8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*
c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
+ 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f
^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a
^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*
(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*
d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^
8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^
4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^
2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4
*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*
A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2
*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^
8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*
d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11
 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a
^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A
^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 -
48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 +
64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2
+ 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^
2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2
*f^4)))^(1/2)*1i - (((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2
 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*
f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d
^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a
^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 -
 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7
*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 +
 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3
*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*
f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*
b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^1
0*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^

14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 64
0*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^
4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10
*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*
A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*
f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c
*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*
f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^
2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (
A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f
^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*
b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f
^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f
^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4
- 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32
*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a
^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 +
48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4
 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 +
560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 1
12*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*
b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*
c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4
 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b
^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^
2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b
^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c +
d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 +
 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^
2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^
2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b
^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 5
6*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10
*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 +
8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^
3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^
2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4
 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*
f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*
(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4
))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*
a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*
c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4
 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b

^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i)/((16*(A^5*b^7*d^13 - 9*A^5*a^4*b^3*d^13 + 3*A^5*b^7*c^2*d^11 + 2*A^5*b^7*c^4*d^9 - 22*A^5*a^2*b^5*c^2*d^11 - 22*A^5*a^2*b^5*c^4*d^9 + 24*A^5*a^3*b^4*c^3*d^10 - 9*A^5*a^4*b^3*c^2*d^11 + 8*A^5*a*b^6*c^3*d^10 + 8*A^5*a*b^6*c^5*d^8 + 24*A^5*a^3*b^4*c*d^12))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8* b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6 *f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c

^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 1
68*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^
2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^
2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 +
4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a^4*c*f^2 + 8*A^
2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^
2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96
*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 -
16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f
^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*(-((
(8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^
2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^
4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*
f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^
2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6
*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*
a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^1
0 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 +
96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8
- 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8
*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f
^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-(((8*A^2*a
^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A
^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a
^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*A^2*a^4*c*f^2 + 4*
A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*
f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4
)))^(1/2)))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 3
2*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (A^4*c^2 + A^4*d^2)*(16*a^8
*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2
) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d
*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4
*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2i - atan(((((8*(304*C^3*a^3*b^9*d^12*f^2
+ 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^
12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^1
2*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*
c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 54
4*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d
^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128
*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2
*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320
*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c
^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*
b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a
^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*

a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 6
40*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*
f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13
*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 96
0*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^
9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*
b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9
*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (16*(c
+ d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^
3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d
^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2
*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*
C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*
f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^
16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*
d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10
*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^
8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c
^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*
c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*
c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c
*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6
*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C
^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2
+ C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*
a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^
2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a
^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*
x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^
7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*
C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C
^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 6
0*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c
*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C
^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^
2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*
C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*
d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2
*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^
4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*
f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*
(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4
))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*
a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c

*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4
- (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^
4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2
*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^
8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))))^(1/2) + (16*(c + d
*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12
+ 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10
*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d
^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^
2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c
^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^
11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11
))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(
(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*
f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*
f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*
c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*
a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a
^6*b^2*f^4)))^(1/2)*1i - (((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d
^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*
c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c
*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^
3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11
*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^
2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8
*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*
a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^1
0*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f
^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*
C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 9
60*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 -
64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10
*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*
c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C
*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9
*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^
7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96
*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4
*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (16*(c + d*tan(e + f*x))^(1
/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3
*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16
*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2
*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24
*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4
+ 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^

4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*
b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d
^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12
*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12
*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*
b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*
b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b
^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6
*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*
C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f
^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2)
- 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f
^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b
^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^
3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380
*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2
 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 -
 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^1
0*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*
a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f
^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*
a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*
f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^
9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^
2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*
((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d
*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8
*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4
*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2
*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*
a^6*b^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*
d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2
)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f
^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^
2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^
4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)
*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^1
2 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^
10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8
*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*
b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*
b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*
c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*
f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 +
 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2

```
*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*
f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(
a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*
1i)/((((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^
7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*
a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a
*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^
2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b
^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 +
 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b
^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2
+ 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*
b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a
^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4
+ 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^
4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*
f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9
*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^
2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^
4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9
*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*
C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*
f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4
*b^5*f^5 + 4*a^6*b^3*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*
f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^
2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^
6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b
^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/
(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(
1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 16
0*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a
^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^1
6*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a
^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 1
6*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336
*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*
a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*
f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2
*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48
*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64
*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 -
4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*
c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f
^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 12
```

8*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^
2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^1
1*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*
f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*
b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 -
220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^1
2*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2
+ 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^
7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 +
4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b
*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2
+ 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^
2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^
4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c
*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*
(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)
)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b
*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b
^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a
^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C
^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 +
4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 -
C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d
^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*
b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a
^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4
*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*
a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5
*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4
+ 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 3
2*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c
^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 +
64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d
*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 +
4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (((8*(304*C^3*a^3*
b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^
3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^
3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64
*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d
^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*
a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^
10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3
*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d
^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C
^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*

f^5 + 4*a^6*b^3*f^5) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f
^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11
*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*
d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9
*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 32
0*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*
d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^
8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4
- 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^1
0*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^
5) + (16*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 -
32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4
*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4
+ 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3
*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4
+ 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^18*d^10*f^4
+ 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 1
60*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a
^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4
*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 4
8*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4
+ 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 5
60*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 11
2*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/(b^9*f^4 + a^8*b*f^4 + 4*a^2
*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*
c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4
- (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b
^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^
2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b
^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c +
d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2
+ 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^
11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^
11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2
*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C
^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^1
0*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2
*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*
d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*
a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*
f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 +
6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*
C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2
+ C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64
*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f

^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(C^5*a^8*d^13 + 10*C^5*a^2*b^6*d^13 + 27*C^5*a^4*b^4*d^13 + 10*C^5*a^6*b^2*d^13 + C^5*a^8*c^2*d^11 + 36*C^5*a^2*b^6*c^2*d^11 + 26*C^5*a^2*b^6*c^4*d^9 - 40*C^5*a^3*b^5*c^3*d^10 + 29*C^5*a^4*b^4*c^2*d^11 + 2*C^5*a^4*b^4*c^4*d^9 - 8*C^5*a^5*b^3*c^3*d^10 + 10*C^5*a^6*b^2*c^2*d^11 - 8*C^5*a*b^7*c*d^12 - 16*C^5*a*b^7*c^3*d^10 - 8*C^5*a*b^7*c^5*d^8 - 40*C^5*a^3*b^5*c*d^12 - 8*C^5*a^5*b^3*c*d^12))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5)))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (C^4*c^2 + C^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2i + atan(((((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^

4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^
4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c
*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*
b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 -
400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3
*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9
*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 -
16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*
f^5 + 4*a^6*b^2*f^5) - (16*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 +
 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2
*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*
f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(
a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*
(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6
*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^
5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*
d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9
*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*
b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^
12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^
6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b
^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^
2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*
b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*
f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4
*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(1
6*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/
2) - (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*
b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a
^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^
2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 +
120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*
c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^
2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3
*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*
a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f
^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 +
 b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*
f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^
2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^
6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b
^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/
(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(
1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a

^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i - (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a

^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4

+ 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3
*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4
+ 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*1i)/((((8*(156*B^3
*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48
*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2
- 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^
9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a
^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 +
44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^
8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 +
200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*
c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 2
0*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5
+ 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14*d^11*
f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*
f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*
f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^
10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^
7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*
a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*
f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^
8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B
*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f
^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) -
(16*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B
^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2
+ B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*
a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^
2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a
^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*(32*b^17*d^10*f^4 + 160
*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^
8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^
3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*
c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*
b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*
b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*
b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*
b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4
+ 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 3
2*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c
^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 +
64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d
*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 +
4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e +
f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a

^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2) + (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4

+ 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^
4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^
4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c
*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*
b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 -
400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3
*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9
*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 -
16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*
f^5 + 4*a^6*b^2*f^5) + (16*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 +
8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2
*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4
+ 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*
f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(
a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*
(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6
*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^
5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*
d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9
*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*
b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^
12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^
6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/(a^8*f^4 + b
^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*f^
2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*
b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*
f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4
*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(1
6*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/
2) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*
b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a
^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^
2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 +
120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*
c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^
2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3
*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*
a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f
^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 +
b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*((((8*B^2*a^4*c*
f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^
2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^
6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b
^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/
(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(

1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a
^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 +
16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B
^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 -
24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^
4 + 4*a^6*b^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12
 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^
9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^
8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d
^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d
^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 6
4*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a
^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*
b^2*f^4))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B
^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^
4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) +
 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^
2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^
4*f^4 + 4*a^6*b^2*f^4)))^(1/2) - (16*(2*B^5*a^3*b^4*d^13 + 4*B^5*b^7*c^3*d^
10 - 6*B^5*a*b^6*d^13 + 4*B^5*b^7*c*d^12 - 9*B^5*a^2*b^5*c^3*d^10 + 4*B^5*a
^2*b^5*c^5*d^8 - 12*B^5*a^3*b^4*c^2*d^11 - 14*B^5*a^3*b^4*c^4*d^9 + 2*B^5*a
^4*b^3*c^3*d^10 - 4*B^5*a^4*b^3*c^5*d^8 + 4*B^5*a^5*b^2*c^2*d^11 + 4*B^5*a^
5*b^2*c^4*d^9 - B^5*a^6*b*c*d^12 + 6*B^5*a*b^6*c^4*d^9 - 13*B^5*a^2*b^5*c*d
^12 + 6*B^5*a^4*b^3*c*d^12 - B^5*a^6*b*c^3*d^10))/(a^8*f^5 + b^8*f^5 + 4*a^
2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5)))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^
4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2
/4 - (B^4*c^2 + B^4*d^2)*(16*a^8*f^4 + 16*b^8*f^4 + 64*a^2*b^6*f^4 + 96*a^4
*b^4*f^4 + 64*a^6*b^2*f^4))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*
B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(a^8*f^4 +
 b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)))^(1/2)*2i + (ata
n(-((((((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a
^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3
*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*
a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f
^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*
b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2
+ 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*
b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2
 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8
*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*
a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (((16*(c + d*tan(e + f*x))^(
1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7
*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b
^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^
14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2

*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10
*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^
12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9
*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a
^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f
^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11
*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4
*a^6*b^3*f^4) + (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 96
0*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 +
96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^
4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*
d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3
*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4
 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c
^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*
a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))
/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (4
*(c + d*tan(e + f*x))^(1/2)*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a
^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(
b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4
*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^
2 - a*b^11*d*f^2))^(1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^
4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*
b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d
^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12
*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12
*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*
b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*
b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b
^3*c*d^9*f^4))/((b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^
6*b^3*f^4)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f
^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 -
a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C
^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*
d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8
*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*
d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8
*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^
2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*
C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*
c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 +
 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^
7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*
b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*
d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))

```
*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2
*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f
^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 -
6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(
4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*
b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d
*f^2 - a*b^11*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C
^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^
12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b
^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^
3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*
a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a
^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*
b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 +
 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*
C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c
*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^
8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3
*d*f^2 - a*b^11*d*f^2))^(1/2)*1i)/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4
*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*
d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)) - (((((8*(304*C^3*
a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 14
8*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 1
6*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2
+ 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8
*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*
C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^
2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72
*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c
^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 1
48*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*
b^5*f^5 + 4*a^6*b^3*f^5) + (((16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^1
1*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*
a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60
*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C
^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2
 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b
^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 +
116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b
^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 +
 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5
*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(
b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4) - (((8
*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^
4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f
```

^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c
*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11
*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 +
480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^
2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a
^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4
 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^
5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (4*(c + d*tan(e + f*x)
)^(1/2)*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*
a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b
^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d
*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(
1/2)*(32*b^18*d^10*f^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 16
0*a^6*b^12*d^10*f^4 - 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a
^12*b^6*d^10*f^4 - 32*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^1
6*c^2*d^8*f^4 + 624*a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a
^8*b^10*c^2*d^8*f^4 + 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 1
6*a^14*b^4*c^2*d^8*f^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336
*a^5*b^13*c*d^9*f^4 + 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*
a^11*b^7*c*d^9*f^4 + 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/((b^9
*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4)*(b^12*c*f
^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 -
 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^
11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*
C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a
^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b
^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2
))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c
*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2
 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25
*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*
c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a
^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^
3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b
^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*
f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 1
6*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^
5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2
 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*
a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^
2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^
9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)
) - (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4
*a^2*b^8*d^12 + 27*C^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d
^12 + C^4*a^10*c^2*d^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4

```
*a^2*b^8*c^2*d^10 - 12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*
C^4*a^4*b^6*c^2*d^10 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17
*C^4*a^6*b^4*c^2*d^10 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*
C^4*a^9*b*c*d^11 - 40*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*
a^7*b^3*c*d^11))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a
^6*b^3*f^4))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10
*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*
a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*
b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^
2))^(1/2)*1i)/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b
^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*
f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))/((((((8*(304*C^3*a^3*b^9*d^12*f^2 + 1
20*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f
^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^
2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*
d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^
3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*
f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3
*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^1
0*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3
*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d
^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*
f^5) - (((16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2
*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 1
00*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2
 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 +
 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*
c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*
C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3
*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 10
4*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2
*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2
*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4
+ 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4) + (((8*(96*C*a^2*b^14*d^11
*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^
11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3
*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^
7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96
*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d
^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^
7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 +
 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2
*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 +
6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) - (4*(c + d*tan(e + f*x))^(1/2)*(4*(C^2*a^8*
d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2
```

*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^
8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f
^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2)*(32*b^18*d^10*f
^4 + 160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4
- 160*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 3
2*a^14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*
a^4*b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4
+ 48*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f
^4 + 16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4
+ 560*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 +
 112*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/((b^9*f^4 + a^8*b*f^4 + 4
*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4)*(b^12*c*f^2 + 4*a^2*b^10*c*f^
2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6
*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*
a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40
*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^
4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7
*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c
*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2
 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*
b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 1
0*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4
*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3
*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f
^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6
*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^
2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 +
25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^
3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 +
 a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*
b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4
*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*
d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)) + (16*(c + d*tan(e
 + f*x))^(1/2)*(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C
^4*a^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d
^10 + 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 -
12*C^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10
 + 18*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^1
0 - 8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 4
0*C^4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^
9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(4*(C^2
*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 4
0*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a
^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^
7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*

c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)) - (16*(C^5*a^8*d^13 + 10*C^5*a^2*b^6*d^13 + 27*C^5*a^4*b^4*d^13 + 10*C^5*a^6*b^2*d^13 + C^5*a^8*c^2*d^11 + 36*C^5*a^2*b^6*c^2*d^11 + 26*C^5*a^2*b^6*c^4*d^9 - 40*C^5*a^3*b^5*c^3*d^10 + 29*C^5*a^4*b^4*c^2*d^11 + 2*C^5*a^4*b^4*c^4*d^9 - 8*C^5*a^5*b^3*c^3*d^10 + 10*C^5*a^6*b^2*c^2*d^11 - 8*C^5*a*b^7*c*d^12 - 16*C^5*a*b^7*c^3*d^10 - 8*C^5*a*b^7*c^5*d^8 - 40*C^5*a^3*b^5*c*d^12 - 8*C^5*a^5*b^3*c*d^12))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (((((8*(304*C^3*a^3*b^9*d^12*f^2 + 120*C^3*a^5*b^7*d^12*f^2 - 320*C^3*a^7*b^5*d^12*f^2 - 148*C^3*a^9*b^3*d^12*f^2 + 4*C^3*b^12*c^3*d^9*f^2 - 4*C^3*a*b^11*d^12*f^2 - 16*C^3*a^11*b*d^12*f^2 + 4*C^3*b^12*c*d^11*f^2 + 60*C^3*a*b^11*c^2*d^10*f^2 + 64*C^3*a*b^11*c^4*d^8*f^2 - 320*C^3*a^2*b^10*c*d^11*f^2 + 104*C^3*a^4*b^8*c*d^11*f^2 + 544*C^3*a^6*b^6*c*d^11*f^2 + 116*C^3*a^8*b^4*c*d^11*f^2 - 16*C^3*a^11*b*c^2*d^10*f^2 - 320*C^3*a^2*b^10*c^3*d^9*f^2 + 176*C^3*a^3*b^9*c^2*d^10*f^2 - 128*C^3*a^3*b^9*c^4*d^8*f^2 + 104*C^3*a^4*b^8*c^3*d^9*f^2 - 72*C^3*a^5*b^7*c^2*d^10*f^2 - 192*C^3*a^5*b^7*c^4*d^8*f^2 + 544*C^3*a^6*b^6*c^3*d^9*f^2 - 320*C^3*a^7*b^5*c^2*d^10*f^2 + 116*C^3*a^8*b^4*c^3*d^9*f^2 - 148*C^3*a^9*b^3*c^2*d^10*f^2))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (((16*(c + d*tan(e + f*x))^(1/2)*(52*C^2*a^3*b^11*d^11*f^2 + 128*C^2*a^5*b^9*d^11*f^2 + 424*C^2*a^7*b^7*d^11*f^2 + 380*C^2*a^9*b^5*d^11*f^2 + 100*C^2*a^11*b^3*d^11*f^2 - 20*C^2*b^14*c^3*d^8*f^2 + 60*C^2*a*b^13*d^11*f^2 + 8*C^2*a^13*b*d^11*f^2 - 4*C^2*a^14*c*d^10*f^2 - 12*C^2*b^14*c*d^10*f^2 + 84*C^2*a*b^13*c^2*d^9*f^2 + 60*C^2*a^2*b^12*c*d^10*f^2 - 116*C^2*a^4*b^10*c*d^10*f^2 - 604*C^2*a^6*b^8*c*d^10*f^2 - 596*C^2*a^8*b^6*c*d^10*f^2 - 220*C^2*a^10*b^4*c*d^10*f^2 - 44*C^2*a^12*b^2*c*d^10*f^2 + 116*C^2*a^2*b^12*c^3*d^8*f^2 + 108*C^2*a^3*b^11*c^2*d^9*f^2 + 216*C^2*a^4*b^10*c^3*d^8*f^2 + 104*C^2*a^5*b^9*c^2*d^9*f^2 + 8*C^2*a^6*b^8*c^3*d^8*f^2 + 248*C^2*a^7*b^7*c^2*d^9*f^2 - 68*C^2*a^8*b^6*c^3*d^8*f^2 + 196*C^2*a^9*b^5*c^2*d^9*f^2 + 4*C^2*a^10*b^4*c^3*d^8*f^2 + 28*C^2*a^11*b^3*c^2*d^9*f^2))/(b^9*f^4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4) - (((8*(96*C*a^2*b^14*d^11*f^4 + 480*C*a^4*b^12*d^11*f^4 + 960*C*a^6*b^10*d^11*f^4 + 960*C*a^8*b^8*d^11*f^4 + 480*C*a^10*b^6*d^11*f^4 + 96*C*a^12*b^4*d^11*f^4 - 64*C*a*b^15*c^3*d^8*f^4 - 320*C*a^3*b^13*c*d^10*f^4 - 640*C*a^5*b^11*c*d^10*f^4 - 640*C*a^7*b^9*c*d^10*f^4 - 320*C*a^9*b^7*c*d^10*f^4 - 64*C*a^11*b^5*c*d^10*f^4 + 96*C*a^2*b^14*c^2*d^9*f^4 - 320*C*a^3*b^13*c^3*d^8*f^4 + 480*C*a^4*b^12*c^2*d^9*f^4 - 640*C*a^5*b^11*c^3*d^8*f^4 + 960*C*a^6*b^10*c^2*d^9*f^4 - 640*C*a^7*b^9*c^3*d^8*f^4 + 960*C*a^8*b^8*c^2*d^9*f^4 - 320*C*a^9*b^7*c^3*d^8*f^4 + 480*C*a^10*b^6*c^2*d^9*f^4 - 64*C*a^11*b^5*c^3*d^8*f^4 + 96*C*a^12*b^4*c^2*d^9*f^4 - 64*C*a*b^15*c*d^10*f^4))/(b^9*f^5 + a^8*b*f^5 + 4*a^2*b^7*f^5 + 6*a^4*b^5*f^5 + 4*a^6*b^3*f^5) + (4*(c + d*tan(e + f*x))^(1/2)*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2)*(32*b^18*d^10*f^4 +

160*a^2*b^16*d^10*f^4 + 288*a^4*b^14*d^10*f^4 + 160*a^6*b^12*d^10*f^4 - 16
0*a^8*b^10*d^10*f^4 - 288*a^10*b^8*d^10*f^4 - 160*a^12*b^6*d^10*f^4 - 32*a^
14*b^4*d^10*f^4 + 48*b^18*c^2*d^8*f^4 + 272*a^2*b^16*c^2*d^8*f^4 + 624*a^4*
b^14*c^2*d^8*f^4 + 720*a^6*b^12*c^2*d^8*f^4 + 400*a^8*b^10*c^2*d^8*f^4 + 48
*a^10*b^8*c^2*d^8*f^4 - 48*a^12*b^6*c^2*d^8*f^4 - 16*a^14*b^4*c^2*d^8*f^4 +
16*a*b^17*c*d^9*f^4 + 112*a^3*b^15*c*d^9*f^4 + 336*a^5*b^13*c*d^9*f^4 + 56
0*a^7*b^11*c*d^9*f^4 + 560*a^9*b^9*c*d^9*f^4 + 336*a^11*b^7*c*d^9*f^4 + 112
*a^13*b^5*c*d^9*f^4 + 16*a^15*b^3*c*d^9*f^4))/((b^9*f^4 + a^8*b*f^4 + 4*a^2
*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 +
6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5
*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*
d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2
*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^
8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f
^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2
+ 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4
*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11
*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^
2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2
*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9
*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))
^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f
^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 -
a^9*b^3*d*f^2 - a*b^11*d*f^2)))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C
^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*
d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8
*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*
d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8
*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^
2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)) - (16*(c + d*tan(e + f
x))^(1/2)(2*C^4*b^10*d^12 - C^4*a^10*d^12 + 4*C^4*a^2*b^8*d^12 + 27*C^4*a
^4*b^6*d^12 - 15*C^4*a^6*b^4*d^12 - 9*C^4*a^8*b^2*d^12 + C^4*a^10*c^2*d^10
+ 4*C^4*b^10*c^2*d^10 + 2*C^4*b^10*c^4*d^8 + 24*C^4*a^2*b^8*c^2*d^10 - 12*C
^4*a^2*b^8*c^4*d^8 + 104*C^4*a^3*b^7*c^3*d^9 - 197*C^4*a^4*b^6*c^2*d^10 + 1
8*C^4*a^4*b^6*c^4*d^8 - 32*C^4*a^5*b^5*c^3*d^9 - 17*C^4*a^6*b^4*c^2*d^10 -
8*C^4*a^7*b^3*c^3*d^9 + 9*C^4*a^8*b^2*c^2*d^10 + 4*C^4*a^9*b*c*d^11 - 40*C^
4*a^3*b^7*c*d^11 + 132*C^4*a^5*b^5*c*d^11 + 48*C^4*a^7*b^3*c*d^11))/(b^9*f^
4 + a^8*b*f^4 + 4*a^2*b^7*f^4 + 6*a^4*b^5*f^4 + 4*a^6*b^3*f^4))*(4*(C^2*a^8
*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*C^2*a^6*b^2*d^2 - 40*C^
2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b
^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*
f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2))^(1/2))/(4*(b^12*c*f^
2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 -
4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^1
1*d*f^2))))*(4*(C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 25*C^2*a^4*b^4*d^2 + 10*

C^2*a^6*b^2*d^2 - 40*C^2*a^3*b^5*c*d - 8*C^2*a^5*b^3*c*d)*(b^12*c*f^2 + 4*a
^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b
^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2
))^(1/2)*1i)/(2*(b^12*c*f^2 + 4*a^2*b^10*c*f^2 + 6*a^4*b^8*c*f^2 + 4*a^6*b^
6*c*f^2 + a^8*b^4*c*f^2 - 4*a^3*b^9*d*f^2 - 6*a^5*b^7*d*f^2 - 4*a^7*b^5*d*f
^2 - a^9*b^3*d*f^2 - a*b^11*d*f^2)) + (atan(((((16*(c + d*tan(e + f*x))^(1/
2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b
^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*
B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14
*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4
*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4
*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*
b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^
4*b^4*f^4 + 4*a^6*b^2*f^4) - (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^
7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^
11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3
*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^
2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^
2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 11
2*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^
2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*
B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3
*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^
3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5
+ 4*a^6*b^2*f^5) + (((((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 38
4*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 +
240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 8
0*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*
f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*
d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^
3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^
4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^
3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^
12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2
*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) - (4*(c + d*tan(e + f*x))^(1/2)*(
4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*
B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d
 - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6
*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*
d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*
d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10
*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4
 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^
4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^
8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9

*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9
*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*
f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b
^4*f^4 + 4*a^6*b^2*f^4)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4
*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*
b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 -
8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d
^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 +
4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^
3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2
))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*
f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 -
a*b^9*d*f^2 - a^9*b*d*f^2)) - (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b
^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a
^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B
^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100
*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d
^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a
^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9
*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6
*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2
+ 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^
2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*
b^2*f^4))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b
^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^
2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*
f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 -
4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^
2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7
*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(
4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*
B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d
- 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6
*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*
d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2
+ 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*
a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*
d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4
*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^
5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2
+ a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b
^9*d*f^2 - a^9*b*d*f^2))^(1/2)*1i)/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4
*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*
d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) + (((16*(c + d*tan(e
+ f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 -

7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4) + (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (4*(c + d*tan(e + f*x))^(1/2)*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^

2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)*1i)/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))/((16*(2*B^5*a^3*b^4*d^13 + 4*B^5*b^7*c^3*d^10 - 6*B^5*a*b^6*d^13 + 4*B^5*b^7*c*d^12 - 9*B^5*a^2*b^5*c^3*d^10 + 4*B^5*a^2*b^5*c^5*d^8 - 12*B^5*a^3*b^4*c^2*d^11 - 14*B^5*a^3*b^4*c^4*d^9 + 2*B^5*a^4*b^3*c^3*d^10 - 4*B^5*a^4*b^3*c^5*d^8 + 4*B^5*a^5*b^2*c^2*d^11 + 4*B^5*a^5*b^2*c^4*d^9 - B^5*a^6*b*c*d^12 + 6*B^5*a*b^6*c^4*d^9 - 13*B^5*a^2*b^5*c*d^12 + 6*B^5*a^4*b^3*c*d^12 - B^5*a^6*b*c^3*d^10))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d

^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*
d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3
*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*
d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 +
44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^
4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4) - (((8*(156*B^
3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12*f^2 + 4
8*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2
 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d
^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*
a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2
+ 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b
^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 +
 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5
*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 -
20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^
5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(80*B*a*b^14*d^
11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^
11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^
11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c
*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8
*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224
*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d
^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7
*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 3
2*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^
9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5)
 - (4*(c + d*tan(e + f*x))^(1/2)*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^
2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*
B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b
^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*
f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)
*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^
6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b
^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2
*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^
9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14
*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b
^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b
^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 +
 b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)*(b^10*c*f^2 + 4*a
^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^
7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*
(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9
*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*

d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) - (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2*d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) - (((16*(c + d*tan(e + f*x))^(1/2)*(2*B^4*b^9*d^12 - 5*B^4*a^2*b^7*d^12 + 17*B^4*a^4*b^5*d^12 - 7*B^4*a^6*b^3*d^12 + 6*B^4*b^9*c^4*d^8 + B^4*a^8*b*d^12 + 77*B^4*a^2*b^7*c^2*d^10 - 8*B^4*a^2*b^7*c^4*d^8 + 60*B^4*a^3*b^6*c^3*d^9 - 87*B^4*a^4*b^5*c^2*d^10 + 14*B^4*a^4*b^5*c^4*d^8 - 36*B^4*a^5*b^4*c^3*d^9 + 27*B^4*a^6*b^3*c^2*d^10 - 4*B^4*a^6*b^3*c^4*d^8 + 4*B^4*a^7*b^2*c^3*d^9 + 12*B^4*a*b^8*c*d^11 - 28*B^4*a*b^8*c^3*d^9 - 64*B^4*a^3*b^6*c*d^11 + 44*B^4*a^5*b^4*c*d^11 - 8*B^4*a^7*b^2*c*d^11 - B^4*a^8*b*c^2*d^10))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4) + (((8*(156*B^3*a^2*b^9*d^12*f^2 - 16*B^3*a^4*b^7*d^12*f^2 - 120*B^3*a^6*b^5*d^12 *f^2 + 48*B^3*a^8*b^3*d^12*f^2 + 12*B^3*b^11*c^2*d^10*f^2 + 12*B^3*b^11*c^4*d^8*f^2 - 4*B^3*a^10*b*d^12*f^2 - 124*B^3*a*b^10*c*d^11*f^2 - 124*B^3*a*b^10*c^3*d^9*f^2 + 224*B^3*a^3*b^8*c*d^11*f^2 + 200*B^3*a^5*b^6*c*d^11*f^2 - 128*B^3*a^7*b^4*c*d^11*f^2 + 20*B^3*a^9*b^2*c*d^11*f^2 - 4*B^3*a^10*b*c^2*d^10*f^2 + 44*B^3*a^2*b^9*c^2*d^10*f^2 - 112*B^3*a^2*b^9*c^4*d^8*f^2 + 224*B^3*a^3*b^8*c^3*d^9*f^2 - 40*B^3*a^4*b^7*c^2*d^10*f^2 - 24*B^3*a^4*b^7*c^4*d^8*f^2 + 200*B^3*a^5*b^6*c^3*d^9*f^2 - 40*B^3*a^6*b^5*c^2*d^10*f^2 + 80*B^3*a^6*b^5*c^4*d^8*f^2 - 128*B^3*a^7*b^4*c^3*d^9*f^2 + 28*B^3*a^8*b^3*c^2*d^10*f^2 - 20*B^3*a^8*b^3*c^4*d^8*f^2 + 20*B^3*a^9*b^2*c^3*d^9*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(80*B*a*b^14*d^11*f^4 - 48*B*b^15*c*d^10*f^4 + 384*B*a^3*b^12*d^11*f^4 + 720*B*a^5*b^10*d^11*f^4 + 640*B*a^7*b^8*d^11*f^4 + 240*B*a^9*b^6*d^11*f^4 - 16*B*a^13*b^2*d^11*f^4 - 48*B*b^15*c^3*d^8*f^4 + 80*B*a*b^14*c^2*d^9*f^4 - 224*B*a^2*b^13*c*d^10*f^4 - 400*B*a^4*b^11*c*d^10*f^4 - 320*B*a^6*b^9*c*d^10*f^4 - 80*B*a^8*b^7*c*d^10*f^4 + 32*B*a^10*b^5*c*d^10*f^4 + 16*B*a^12*b^3*c*d^10*f^4 - 224*B*a^2*b^13*c^3*d^8*f^4 + 384*B*a^3*b^12*c^2*d^9*f^4 - 400*B*a^4*b^11*c^3*d^8*f^4 + 720*B*a^5*b^10*c^2*d^9*f^4 - 320*B*a^6*b^9*c^3*d^8*f^4 + 640*B*a^7*b^8*c^2*d^9*f^4 - 80*B*a^8*b^7*c^3*d^8*f^4 + 240*B*a^9*b^6*c^2*d^9*f^4 + 32*B*a^10*b^5*c^3*d^8*f^4 + 16*B*a^12*b^3*c^3*d^8*f^4 - 16*B*a^13*b^2*c^2*d^9*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (4*(c + d*tan(e + f*x))^(1/2)*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(44*B^2*a^9*b^4*d^11*f^2 - 168*B^2*a^5*b^8*d^11*f^2 - 40*B^2*a^7*b^6*d^11*f^2 - 20*B^2*a^3*b^10*d^11*f^2 - 4*B^2*a^11*b^2 *d^11*f^2 - 36*B^2*b^13*c^3*d^8*f^2 + 60*B^2*a*b^12*d^11*f^2 - 12*B^2*b^13*c*d^10*f^2 + 4*B^2*a^12*b*c*d^10*f^2 + 100*B^2*a*b^12*c^2*d^9*f^2 + 120*B^2*a^2*b^11*c*d^10*f^2 + 156*B^2*a^4*b^9*c*d^10*f^2 - 112*B^2*a^6*b^7*c*d^10*f^2 - 148*B^2*a^8*b^5*c*d^10*f^2 - 8*B^2*a^10*b^3*c*d^10*f^2 + 68*B^2*a^2*b^11*c^3*d^8*f^2 + 124*B^2*a^3*b^10*c^2*d^9*f^2 + 184*B^2*a^4*b^9*c^3*d^8*f^2 + 8*B^2*a^5*b^8*c^2*d^9*f^2 + 40*B^2*a^6*b^7*c^3*d^8*f^2 + 24*B^2*a^7*b^6*c^2*d^9*f^2 - 20*B^2*a^8*b^5*c^3*d^8*f^2 + 20*B^2*a^9*b^4*c^2*d^9*f^2 + 20*B^2*a^10*b^3*c^3*d^8*f^2 - 20*B^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2))/(4*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))))*(4*(B^2*a^6*d^2 + 4*B^2*b^6*c^2 - 8*B^2*a^2*b^4*c^2 + 4*B^2*a^4*b^2*c^2 + 9*B^2*a^2*b^4*d^2 - 6*B^2*a^4*b^2*d^2 + 16*B^2*a^3*b^3*c*d - 12*B^2*a*b^5*c*d - 4*B^2*a^5*b*c*d)*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2))^(1/2)*1i)/(2*(b^10*c*f^2 + 4*a^2*b^8*c*f^2 + 6*a^4*b^6*c*f^2 + 4*a^6*b^4*c*f^2 + a^8*b^2*c*f^2 - 4*a^3*b^7*d*f^2 - 6*a^5*b^5*d*f^2 - 4*a^7*b^3*d*f^2 - a*b^9*d*f^2 - a^9*b*d*f^2)) - (atan((((((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b

^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(32*A*b^15*d^11*f^4 + 96*A *a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288* A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A *a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^ 10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5 *b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2 *d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b ^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6 *b^2*f^5) - (4*(c + d*tan(e + f*x))^(1/2)*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3 *c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b ^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6* b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f ^2 + a*b^8*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^ 4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b ^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^ 8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11* c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b ^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^ 14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^ 8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2 *c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^ 2*f^4)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b ^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^ 2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^ 2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^ 9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d *f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^ 3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11* f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^1 1*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^ 11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 29 6*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^ 3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 16 8*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2 *d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2 *a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4 *a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^ 2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^ 2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4 *a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8* b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) - (16*(c + d*tan(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*1i)/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) - (((((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(32*A*b^15*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (4*(c + d*tan(e + f*x))^(1/2)*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f^4 + 288*a^4*b

^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*
d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48*b^17*c^2*d^8*f
^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 720*a^6*b^11*c^2
*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*
c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*
c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c
*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*
d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f
^4)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*
c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 +
a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 +
9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c
*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^
2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/
(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c
*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 +
a*b^8*d*f^2)) - (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^10*d^11*f^2 -
88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^9*b^4*d^11*f^2
+ 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2*a*b^12*d^11*f
^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104*A^2*a^2*b^11*
c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^10*f^2 - 296*A
^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2*a^2*b^11*c^3*d
^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d^8*f^2 + 168*A
^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2*a^7*b^6*c^2*d^
9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9*f^2 + 4*A^2*a^
10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 + b^8*f^4 + 4*a^
2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b
^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a
*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^
6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c
*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6
*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^
7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^
3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*
b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6
*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*
f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*
a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7
*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(
3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9*A^4*a^6*b^3*d
^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7*c^2*d^10 - 12
*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b^5*c^2*d^10 +
18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b^3*c^2*d^10 +
12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*d^11 + 60*A^4*
a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6

```
*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^
2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2
- 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6
*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*1i)/(4
*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f
^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*
b^8*d*f^2)))/((16*(A^5*b^7*d^13 - 9*A^5*a^4*b^3*d^13 + 3*A^5*b^7*c^2*d^11 +
 2*A^5*b^7*c^4*d^9 - 22*A^5*a^2*b^5*c^2*d^11 - 22*A^5*a^2*b^5*c^4*d^9 + 24*
A^5*a^3*b^4*c^3*d^10 - 9*A^5*a^4*b^3*c^2*d^11 + 8*A^5*a*b^6*c^3*d^10 + 8*A^
5*a*b^6*c^5*d^8 + 24*A^5*a^3*b^4*c*d^12))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^
5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*
A^3*a^5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 +
20*A^3*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 +
 12*A^3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c
*d^11*f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*
a^8*b^3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*
f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a
^5*b^6*c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10
*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 +
b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(32*A*b^1
5*d^11*f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^
7*d^11*f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c
^2*d^9*f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^
5*b^10*c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 6
4*A*a^11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^
8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*
b^8*c^3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 2
88*A*a^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d
^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^
4*b^4*f^5 + 4*a^6*b^2*f^5) - (4*(c + d*tan(e + f*x))^(1/2)*(-4*(A^2*b^5*d^2
 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^
2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b
^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*
d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*
d^10*f^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10
*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4
 + 48*b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^
4 + 720*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^
8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9
*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9
*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*
f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b
^4*f^4 + 4*a^6*b^2*f^4)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^
5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d
*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 -
```

6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d
)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*
f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a
*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5
*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*
f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) + (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*
a^3*b^10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*
A^2*a^9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 +
68*A^2*a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2
 + 104*A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^
7*c*d^10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 11
6*A^2*a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9
*c^3*d^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 18
4*A^2*a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^
2*d^9*f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*
f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^
5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a
^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*
a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7
*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 -
 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*
a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5
*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^
3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a
^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*
b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 -
4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a
^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) - (16*(c + d*t
an(e + f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^
12 - 9*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a
^2*b^7*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4
*a^4*b^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4
*a^6*b^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*
b^6*c*d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6
*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^
2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^
9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 +
 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*
d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^
2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 -
 a^8*b*c*f^2 + a*b^8*d*f^2)) + (((((8*(128*A^3*a^3*b^8*d^12*f^2 + 24*A^3*a^
5*b^6*d^12*f^2 - 160*A^3*a^7*b^4*d^12*f^2 - 4*A^3*a^9*b^2*d^12*f^2 + 20*A^3
*b^11*c^3*d^9*f^2 - 52*A^3*a*b^10*d^12*f^2 + 20*A^3*b^11*c*d^11*f^2 + 12*A^
3*a*b^10*c^2*d^10*f^2 + 64*A^3*a*b^10*c^4*d^8*f^2 - 256*A^3*a^2*b^9*c*d^11*
f^2 + 72*A^3*a^4*b^7*c*d^11*f^2 + 352*A^3*a^6*b^5*c*d^11*f^2 + 4*A^3*a^8*b^

3*c*d^11*f^2 - 256*A^3*a^2*b^9*c^3*d^9*f^2 - 128*A^3*a^3*b^8*c^4*d^8*f^2 + 72*A^3*a^4*b^7*c^3*d^9*f^2 - 168*A^3*a^5*b^6*c^2*d^10*f^2 - 192*A^3*a^5*b^6 *c^4*d^8*f^2 + 352*A^3*a^6*b^5*c^3*d^9*f^2 - 160*A^3*a^7*b^4*c^2*d^10*f^2 + 4*A^3*a^8*b^3*c^3*d^9*f^2 - 4*A^3*a^9*b^2*c^2*d^10*f^2))/(a^8*f^5 + b^8*f^ 5 + 4*a^2*b^6*f^5 + 6*a^4*b^4*f^5 + 4*a^6*b^2*f^5) + (((((8*(32*A*b^15*d^11 *f^4 + 96*A*a^2*b^13*d^11*f^4 - 320*A*a^6*b^9*d^11*f^4 - 480*A*a^8*b^7*d^11 *f^4 - 288*A*a^10*b^5*d^11*f^4 - 64*A*a^12*b^3*d^11*f^4 + 32*A*b^15*c^2*d^9 *f^4 + 64*A*a*b^14*c^3*d^8*f^4 + 320*A*a^3*b^12*c*d^10*f^4 + 640*A*a^5*b^10 *c*d^10*f^4 + 640*A*a^7*b^8*c*d^10*f^4 + 320*A*a^9*b^6*c*d^10*f^4 + 64*A*a^ 11*b^4*c*d^10*f^4 + 96*A*a^2*b^13*c^2*d^9*f^4 + 320*A*a^3*b^12*c^3*d^8*f^4 + 640*A*a^5*b^10*c^3*d^8*f^4 - 320*A*a^6*b^9*c^2*d^9*f^4 + 640*A*a^7*b^8*c^ 3*d^8*f^4 - 480*A*a^8*b^7*c^2*d^9*f^4 + 320*A*a^9*b^6*c^3*d^8*f^4 - 288*A*a ^10*b^5*c^2*d^9*f^4 + 64*A*a^11*b^4*c^3*d^8*f^4 - 64*A*a^12*b^3*c^2*d^9*f^4 + 64*A*a*b^14*c*d^10*f^4))/(a^8*f^5 + b^8*f^5 + 4*a^2*b^6*f^5 + 6*a^4*b^4* f^5 + 4*a^6*b^2*f^5) + (4*(c + d*tan(e + f*x))^(1/2)*(-4*(A^2*b^5*d^2 + 16* A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f ^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2)*(32*b^17*d^10*f^4 + 160*a^2*b^15*d^10*f ^4 + 288*a^4*b^13*d^10*f^4 + 160*a^6*b^11*d^10*f^4 - 160*a^8*b^9*d^10*f^4 - 288*a^10*b^7*d^10*f^4 - 160*a^12*b^5*d^10*f^4 - 32*a^14*b^3*d^10*f^4 + 48* b^17*c^2*d^8*f^4 + 272*a^2*b^15*c^2*d^8*f^4 + 624*a^4*b^13*c^2*d^8*f^4 + 72 0*a^6*b^11*c^2*d^8*f^4 + 400*a^8*b^9*c^2*d^8*f^4 + 48*a^10*b^7*c^2*d^8*f^4 - 48*a^12*b^5*c^2*d^8*f^4 - 16*a^14*b^3*c^2*d^8*f^4 + 16*a*b^16*c*d^9*f^4 + 112*a^3*b^14*c*d^9*f^4 + 336*a^5*b^12*c*d^9*f^4 + 560*a^7*b^10*c*d^9*f^4 + 560*a^9*b^8*c*d^9*f^4 + 336*a^11*b^6*c*d^9*f^4 + 112*a^13*b^4*c*d^9*f^4 + 16*a^15*b^2*c*d^9*f^4))/((a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^ 2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2 *a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9 *d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d *f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) - (16*(c + d*tan(e + f*x))^(1/2)*(20*A^2*a^3*b^ 10*d^11*f^2 - 88*A^2*a^5*b^8*d^11*f^2 + 40*A^2*a^7*b^6*d^11*f^2 + 84*A^2*a^ 9*b^4*d^11*f^2 + 4*A^2*a^11*b^2*d^11*f^2 - 20*A^2*b^13*c^3*d^8*f^2 + 68*A^2 *a*b^12*d^11*f^2 - 8*A^2*b^13*c*d^10*f^2 + 116*A^2*a*b^12*c^2*d^9*f^2 + 104 *A^2*a^2*b^11*c*d^10*f^2 + 48*A^2*a^4*b^9*c*d^10*f^2 - 304*A^2*a^6*b^7*c*d^ 10*f^2 - 296*A^2*a^8*b^5*c*d^10*f^2 - 56*A^2*a^10*b^3*c*d^10*f^2 + 116*A^2* a^2*b^11*c^3*d^8*f^2 + 204*A^2*a^3*b^10*c^2*d^9*f^2 + 216*A^2*a^4*b^9*c^3*d ^8*f^2 + 168*A^2*a^5*b^8*c^2*d^9*f^2 + 8*A^2*a^6*b^7*c^3*d^8*f^2 + 184*A^2* a^7*b^6*c^2*d^9*f^2 - 68*A^2*a^8*b^5*c^3*d^8*f^2 + 100*A^2*a^9*b^4*c^2*d^9* f^2 + 4*A^2*a^10*b^3*c^3*d^8*f^2 - 4*A^2*a^11*b^2*c^2*d^9*f^2))/(a^8*f^4 +

```
b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2
+ 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2
*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^
5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d
*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2
*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^
4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)))*(-4*(A^2*b^5*d^2 +
 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*
c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5
*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*
f^2 - a^8*b*c*f^2 + a*b^8*d*f^2))^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*
b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4
*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)) + (16*(c + d*tan(e +
 f*x))^(1/2)*(3*A^4*b^9*d^12 - 3*A^4*a^2*b^7*d^12 + 17*A^4*a^4*b^5*d^12 - 9
*A^4*a^6*b^3*d^12 + 3*A^4*b^9*c^2*d^10 + 2*A^4*b^9*c^4*d^8 + 63*A^4*a^2*b^7
*c^2*d^10 - 12*A^4*a^2*b^7*c^4*d^8 + 96*A^4*a^3*b^6*c^3*d^9 - 123*A^4*a^4*b
^5*c^2*d^10 + 18*A^4*a^4*b^5*c^4*d^8 - 24*A^4*a^5*b^4*c^3*d^9 + 9*A^4*a^6*b
^3*c^2*d^10 + 12*A^4*a*b^8*c*d^11 - 8*A^4*a*b^8*c^3*d^9 - 56*A^4*a^3*b^6*c*
d^11 + 60*A^4*a^5*b^4*c*d^11))/(a^8*f^4 + b^8*f^4 + 4*a^2*b^6*f^4 + 6*a^4*b
^4*f^4 + 4*a^6*b^2*f^4))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*
b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^
2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3
*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)
)^(1/2))/(4*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*
a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b
*c*f^2 + a*b^8*d*f^2))))*(-4*(A^2*b^5*d^2 + 16*A^2*a^2*b^3*c^2 - 6*A^2*a^2*
b^3*d^2 + 9*A^2*a^4*b*d^2 - 24*A^2*a^3*b^2*c*d + 8*A^2*a*b^4*c*d)*(a^9*d*f^
2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 - 4*a^6*b^3*c*f^2 + 4*a^3
*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^8*b*c*f^2 + a*b^8*d*f^2)
)^(1/2)*1i)/(2*(a^9*d*f^2 - b^9*c*f^2 - 4*a^2*b^7*c*f^2 - 6*a^4*b^5*c*f^2 -
 4*a^6*b^3*c*f^2 + 4*a^3*b^6*d*f^2 + 6*a^5*b^4*d*f^2 + 4*a^7*b^2*d*f^2 - a^
8*b*c*f^2 + a*b^8*d*f^2)) - (A*b*d*(c + d*tan(e + f*x))^(1/2))/((a^2 + b^2)
*(b*f*(c + d*tan(e + f*x)) + a*d*f - b*c*f)) + (B*a*d*(c + d*tan(e + f*x))^
(1/2))/((a^2 + b^2)*(b*f*(c + d*tan(e + f*x)) + a*d*f - b*c*f)) - (C*a^2*d*
(c + d*tan(e + f*x))^(1/2))/(b*(a^2 + b^2)*(b*f*(c + d*tan(e + f*x)) + a*d*
f - b*c*f))
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c + d\tan\left(e + fx\right)}\,\left(A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)\right)}{\left(a + b\tan\left(e + fx\right)\right)^2}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**2,x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x))**2, x)
```

3.96 $\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$

Optimal. Leaf size=543

$$-\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{2bf\left(a^2+b^2\right)(a+b\tan(e+fx))^2}-\frac{\sqrt{c+d\tan(e+fx)}\left(a^4Cd+3a^3bBd-a^2b^2(7Ad+4Bc-9Cd)+ab^3\right.}{4bf\left(a^2+b^2\right)^2(bc-ad)(a+b\tan(e+\right.}$$

```
[Out] 1/4*(3*a^5*b*B*d^2+a^6*C*d^2-3*a^4*b^2*d*(5*A*d+4*B*c-6*C*d)-3*a^2*b^4*(8*A
*c^2-6*A*d^2-16*B*c*d-8*C*c^2+5*C*d^2)+2*a^3*b^3*(20*c*(A-C)*d+B*(4*c^2-13*
d^2))-3*a*b^5*(8*c*(A-C)*d+B*(8*c^2-d^2))-b^6*(4*c*(B*d+2*C*c)-A*(8*c^2+d^2
)))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-a*d+b*c)^(1/2))/b^(3/2)/(a^2+b
^2)^3/(-a*d+b*c)^(3/2)/f-(A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(
1/2))*(c-I*d)^(1/2)/(I*a+b)^3/f+(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c
+I*d)^(1/2))*(c+I*d)^(1/2)/(I*a-b)^3/f-1/2*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x
+e))^(1/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^2-1/4*(3*a^3*b*B*d+a^4*C*d+b^4*(A
*d+4*B*c)+a*b^3*(8*A*c-5*B*d-8*C*c)-a^2*b^2*(7*A*d+4*B*c-9*C*d))*(c+d*tan(f
*x+e))^(1/2)/b/(a^2+b^2)^2/(-a*d+b*c)/f/(a+b*tan(f*x+e))
```

Rubi [A] time = 4.04, antiderivative size = 543, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}} = 0.170$, Rules used = {3645, 3649, 3653, 3539, 3537, 63, 208, 3634}

$$\frac{\left(2a^3b^3\left(20cd(A-C)+B\left(4c^2-13d^2\right)\right)-3a^2b^4\left(8Ac^2-6Ad^2-16Bcd-8c^2C+5Cd^2\right)-3a^4b^2d(5Ad+4Bc-\right.}{4b^{3/2}f\left(\right.}$$

Antiderivative was successfully verified.

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x])^3,x]
```

```
[Out] -(((A - I*B - C)*Sqrt[c - I*d]*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/((I*a + b)^3*f)) + ((A + I*B - C)*Sqrt[c + I*d]*ArcTanh[Sqrt[c + d*Tan
[e + f*x]]/Sqrt[c + I*d]])/((I*a - b)^3*f) + ((3*a^5*b*B*d^2 + a^6*C*d^2 -
3*a^4*b^2*d*(4*B*c + 5*A*d - 6*C*d) - 3*a^2*b^4*(8*A*c^2 - 8*c^2*C - 16*B*c
*d - 6*A*d^2 + 5*C*d^2) + 2*a^3*b^3*(20*c*(A - C)*d + B*(4*c^2 - 13*d^2)) -
 3*a*b^5*(8*c*(A - C)*d + B*(8*c^2 - d^2)) - b^6*(4*c*(2*c*C + B*d) - A*(8*
c^2 + d^2)))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(
4*b^(3/2)*(a^2 + b^2)^3*(b*c - a*d)^(3/2)*f) - ((A*b^2 - a*(b*B - a*C))*Sqr
t[c + d*Tan[e + f*x]])/(2*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^2) - ((3*a^3
*b*B*d + a^4*C*d + b^4*(4*B*c + A*d) + a*b^3*(8*A*c - 8*c*C - 5*B*d) - a^2*
```

```
b^2*(4*B*c + 7*A*d - 9*C*d))*Sqrt[c + d*Tan[e + f*x]])/(4*b*(a^2 + b^2)^2*(
b*c - a*d)*f*(a + b*Tan[e + f*x]))
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
```

```
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2} + \frac{\int}{}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2} - \frac{(3}{}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2} - \frac{(3}{}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2} - \frac{(3}{}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2} - \frac{(3}{}$$

$$= \frac{\left(3a^5bBd^2+a^6Cd^2-3a^4b^2d(4Bc+5Ad-6C\right.}{}$$

$$= -\frac{(A-iB-C)\sqrt{c-id}\,\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)^3f}$$

<u>**Mathematica**</u> **[B]** time = 6.38, size = 2819, normalized size = 5.19

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x])^3,x]
```

```
[Out] (-2*C*Sqrt[c + d*Tan[e + f*x]])/(3*b*f*(a + b*Tan[e + f*x])^2) - (2*(-1/2*(
((b^2*(-3*A*b*c + 4*b*c*C - a*C*d))/2 - a*((-3*b^2*(B*c + (A - C)*d))/2 - (
a*(b*c*C - 3*b*B*d - a*C*d))/2))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*
c - a*d)*f*(a + b*Tan[e + f*x])^2) - (-((((I*Sqrt[c - I*d]*(b*(b*c - a*d)*(
```

```
(3*b*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 + 3*a*b*(b*c -
a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d) + (3*b*(b*c - a*d)*(a^
2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4) + a*((3*(b*c - a
*d)*((b^2*d)/2 - a*(b*c - a*d))*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c -
 4*c*C - B*d)))/4 + (-(b*c) + (a*d)/2)*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4
*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d
 - b*B*d + a*C*d)) - (d*((3*b^2*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) +
a*b*(4*A*c - 4*c*C - B*d)))/4 - a*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*
C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*
B*d + a*C*d))))/2) - I*(a*(b*c - a*d)*((3*b*(3*A*b^2 - 3*a*b*B - a^2*C - 4*
b^2*C)*d*(b*c - a*d))/4 + 3*a*b*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d
- b*B*d + a*C*d) + (3*b*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A
*c - 4*c*C - B*d)))/4) - b*((3*(b*c - a*d)*((b^2*d)/2 - a*(b*c - a*d))*(a^2
*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 + (-(b*c) + (a*d)/
2)*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*
c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d)) - (d*((3*b^2*(b*c
 - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 - a*((
3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a
*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d))))/2)))*ArcTanh[Sqrt[c
+ d*Tan[e + f*x]]/Sqrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*(b*(b*c
 - a*d)*((3*b*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 + 3*a*
b*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d) + (3*b*(b*c -
 a*d)*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4) + a*((3
*(b*c - a*d)*((b^2*d)/2 - a*(b*c - a*d))*(a^2*C*d + b^2*(4*B*c + A*d) + a*b
*(4*A*c - 4*c*C - B*d)))/4 + (-(b*c) + (a*d)/2)*((3*a*(3*A*b^2 - 3*a*b*B -
a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*
C - a*A*d - b*B*d + a*C*d)) - (d*((3*b^2*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c
+ A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 - a*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C
 - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a
*A*d - b*B*d + a*C*d))))/2) + I*(a*(b*c - a*d)*((3*b*(3*A*b^2 - 3*a*b*B - a
^2*C - 4*b^2*C)*d*(b*c - a*d))/4 + 3*a*b*(b*c - a*d)*(A*b*c - a*B*c - b*c*C
 - a*A*d - b*B*d + a*C*d) + (3*b*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) +
 a*b*(4*A*c - 4*c*C - B*d)))/4) - b*((3*(b*c - a*d)*((b^2*d)/2 - a*(b*c - a
*d))*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 + (-(b*c)
 + (a*d)/2)*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 -
3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d)) - (d*((3
*b^2*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))
/4 - a*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2
*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d))))/2)))*ArcTan
h[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((-c - I*d)*f))/(a^2 + b^2) + (2
*Sqrt[b*c - a*d]*(-(a*b*(b*c - a*d)*((3*b*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^
2*C)*d*(b*c - a*d))/4 + 3*a*b*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d -
b*B*d + a*C*d) + (3*b*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c
 - 4*c*C - B*d)))/4)) + (a^2*d*((3*b^2*(b*c - a*d)*(a^2*C*d + b^2*(4*B*c +
A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 - a*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C -
```

```
 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A
*d - b*B*d + a*C*d))))/2 + b^2*((3*(b*c - a*d)*((b^2*d)/2 - a*(b*c - a*d))*
(a^2*C*d + b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 + (-(b*c) + (a
*d)/2)*((3*a*(3*A*b^2 - 3*a*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2
*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d))))*ArcTanh[(Sq
rt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(Sqrt[b]*(a^2 + b^2)*(-(b
*c) + a*d)*f))/((a^2 + b^2)*(b*c - a*d))) - (((3*b^2*(b*c - a*d)*(a^2*C*d +
 b^2*(4*B*c + A*d) + a*b*(4*A*c - 4*c*C - B*d)))/4 - a*((3*a*(3*A*b^2 - 3*a
*b*B - a^2*C - 4*b^2*C)*d*(b*c - a*d))/4 - 3*b^2*(b*c - a*d)*(A*b*c - a*B*c
 - b*c*C - a*A*d - b*B*d + a*C*d)))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*
(b*c - a*d)*f*(a + b*Tan[e + f*x]))))/(2*(a^2 + b^2)*(b*c - a*d))))/(3*b)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.88, size = 9797, normalized size = 18.04

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^3,x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c + d \tan \left(e + f x\right)} \left(A + B \tan \left(e + f x\right) + C \tan^{2}\left(e + f x\right)\right)}{\left(a + b \tan \left(e + f x\right)\right)^{3}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**3,x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x))**3, x)
```

3.97 $\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+f$

Optimal. Leaf size=550

$$\frac{2(c+d\tan(e+fx))^{5/2}\left(168a^3Cd^3-2a^2bd^2(192cC-847Bd)+33ab^2d\left(63d^2(A-C)-18Bcd+8c^2C\right)-\left(b^3\left(198\right.\right.\right.}{3465d^4f}$$

```
[Out] (I*a+b)^3*(A-I*B-C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1
/2))/f+(a+I*b)^3*(I*A-B-I*C)*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(
c+I*d)^(1/2))/f+2*(3*a^2*b*(A*c-B*d-C*c)-b^3*(A*c-B*d-C*c)+a^3*(B*c+(A-C)*d
)-3*a*b^2*(B*c+(A-C)*d))*(c+d*tan(f*x+e))^(1/2)/f+2/3*(a^3*B-3*a*b^2*B+3*a^
2*b*(A-C)-b^3*(A-C))*(c+d*tan(f*x+e))^(3/2)/f+2/3465*(168*a^3*C*d^3-2*a^2*b
*d^2*(-847*B*d+192*C*c)+33*a*b^2*d*(8*c^2*C-18*B*c*d+63*(A-C)*d^2)-b^3*(48*
c^3*C-88*B*c^2*d+198*c*(A-C)*d^2+693*B*d^3))*(c+d*tan(f*x+e))^(5/2)/d^4/f+2
/693*b*(99*b*(A*b+B*a-C*b)*d^2+4*(-a*d+b*c)*(-11*B*b*d-6*C*a*d+6*C*b*c))*ta
n(f*x+e)*(c+d*tan(f*x+e))^(5/2)/d^3/f-2/99*(-11*B*b*d-6*C*a*d+6*C*b*c)*(a+b
*tan(f*x+e))^2*(c+d*tan(f*x+e))^(5/2)/d^2/f+2/11*C*(a+b*tan(f*x+e))^3*(c+d*
tan(f*x+e))^(5/2)/d/f
```

Rubi [A] time = 2.73, antiderivative size = 550, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3647, 3637, 3630, 3528, 3539, 3537, 63, 208}

$$\frac{2(c+d\tan(e+fx))^{5/2}\left(-2a^2bd^2(192cC-847Bd)+168a^3Cd^3+33ab^2d\left(63d^2(A-C)-18Bcd+8c^2C\right)+b^3\left(-\left(\right.\right.\right.}{3465d^4f}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] ((I*a + b)^3*(A - I*B - C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c - I*d]])/f + ((a + I*b)^3*(I*A - B - I*C)*(c + I*d)^(3/2)*ArcTanh[S
qrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(3*a^2*b*(A*c - c*C - B*d) -
 b^3*(A*c - c*C - B*d) + a^3*(B*c + (A - C)*d) - 3*a*b^2*(B*c + (A - C)*d))
*Sqrt[c + d*Tan[e + f*x]])/f + (2*(a^3*B - 3*a*b^2*B + 3*a^2*b*(A - C) - b^
3*(A - C))*(c + d*Tan[e + f*x])^(3/2))/(3*f) + (2*(168*a^3*C*d^3 - 2*a^2*b*
d^2*(192*c*C - 847*B*d) + 33*a*b^2*d*(8*c^2*C - 18*B*c*d + 63*(A - C)*d^2)
- b^3*(48*c^3*C - 88*B*c^2*d + 198*c*(A - C)*d^2 + 693*B*d^3))*(c + d*Tan[e
 + f*x])^(5/2))/(3465*d^4*f) + (2*b*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c -
a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d))*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2
))/(693*d^3*f) - (2*(6*b*c*C - 11*b*B*d - 6*a*C*d)*(a + b*Tan[e + f*x])^2*(
```

```
c + d*Tan[e + f*x])^(5/2))/(99*d^2*f) + (2*C*(a + b*Tan[e + f*x])^3*(c + d*
Tan[e + f*x])^(5/2))/(11*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
```

```
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int (a+b\tan(e+fx))^3(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{2C(a+b\tan(e+fx))^3(\ldots}{11d\ldots}$$

$$= -\frac{2(6bcC-11bBd-6aC\ldots}{\ldots}$$

$$= \frac{2b\left(99b(Ab+aB-bC)\ldots\right.}{\ldots}$$

$$= \frac{2\left(168a^3Cd^3-2a^2bd^2(\ldots\right.}{\ldots}$$

$$= \frac{2\left(a^3B-3ab^2B+3a^2b(\ldots\right.}{\ldots}$$

$$= \frac{2\left(3a^2b(Ac-cC-Bd)\ldots\right.}{\ldots}$$

$$= \frac{2\left(3a^2b(Ac-cC-Bd)\ldots\right.}{\ldots}$$

$$= \frac{2\left(3a^2b(Ac-cC-Bd)\ldots\right.}{\ldots}$$

$$= \frac{2\left(3a^2b(Ac-cC-Bd)\ldots\right.}{\ldots}$$

$$= -\frac{(a-ib)^3(iA+B-iC)\ldots}{\ldots}$$

Mathematica [B] time = 6.41, size = 1290, normalized size = 2.35

$$\frac{2C(c+d\tan(e+fx))^{5/2}(a+b\tan(e+fx))^3}{11df}+\frac{2\left(\frac{(-6bcC+6adC+11bBd)(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}{9df}+\frac{2\left(\frac{b(99b(Ab-Cb+aB)d^2+\cdots}{\cdots}\right)}{\cdots}\right)}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e +
f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^3*(c + d*Tan[e + f*x])^(5/2))/(11*d*f) + (2*(((-6
*b*c*C + 11*b*B*d + 6*a*C*d)*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(5
/2))/(9*d*f) + (2*((b*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C
- 11*b*B*d - 6*a*C*d))*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2))/(14*d*f) -
(2*((2*((-7*a*d*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b
*B*d - 6*a*C*d)))/8 + b*((-693*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (c*
(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d))
)/4))*(c + d*Tan[e + f*x])^(5/2))/(5*d*f) + ((I/2)*((-7*a*d*(3*a^2*(33*A -
25*C)*d^2 + 4*b^2*c*(6*c*C - 11*B*d) - a*b*d*(48*c*C + 55*B*d)))/8 + (b*c*(
99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d)))
/4 + (7*a*d*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d
 - 6*a*C*d)))/8 + ((7*I)/2)*d*((99*a*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2)/4
 + (b*(3*a^2*(33*A - 25*C)*d^2 + 4*b^2*c*(6*c*C - 11*B*d) - a*b*d*(48*c*C +
```

```
55*B*d)))/4 - (b*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11
*b*B*d - 6*a*C*d)))/4) - b*((-693*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 +
(c*(99*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*
d)))/4))*((2*(c + d*Tan[e + f*x])^(3/2))/3 + (c - I*d)*((2*(c - I*d)^(3/2)*
ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/(-c + I*d) + 2*Sqrt[c + d*
Tan[e + f*x]])))/f - ((I/2)*((-7*a*d*(3*a^2*(33*A - 25*C)*d^2 + 4*b^2*c*(6*
c*C - 11*B*d) - a*b*d*(48*c*C + 55*B*d)))/8 + (b*c*(99*b*(A*b + a*B - b*C)*
d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d)))/4 + (7*a*d*(99*b*(A*b
+ a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d)))/8 - ((7*I
)/2)*d*((99*a*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2)/4 + (b*(3*a^2*(33*A - 25
*C)*d^2 + 4*b^2*c*(6*c*C - 11*B*d) - a*b*d*(48*c*C + 55*B*d)))/4 - (b*(99*b
*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d)))/4)
- b*((-693*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (c*(99*b*(A*b + a*B - b
*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 11*b*B*d - 6*a*C*d)))/4))*((2*(c + d*Tan
[e + f*x])^(3/2))/3 + (c + I*d)*((2*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[
e + f*x]]/Sqrt[c + I*d]])/(-c - I*d) + 2*Sqrt[c + d*Tan[e + f*x]])))/f))/(7
*d)))/(9*d)))/(11*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.64, size = 11056, normalized size = 20.10

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^3*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) +
 C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a + b \tan \left(e + f x\right)\right)^{3} \left(c + d \tan \left(e + f x\right)\right)^{\frac{3}{2}} \left(A + B \tan \left(e + f x\right) + C \tan^{2} \left(e + f x\right)\right)\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**3*(c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan
(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**3*(c + d*tan(e + f*x))**(3/2)*(A + B*tan(e +
 f*x) + C*tan(e + f*x)**2), x)
```

3.98 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+f\right.$

Optimal. Leaf size=396

$$\frac{2(c+d\tan(e+fx))^{5/2}\left(28a^2Cd^2-18abd(2cC-7Bd)+b^2\left(63d^2(A-C)-18Bcd+8c^2C\right)\right)}{315d^3f}+\frac{2\left(a^2B+2ab(A-\right.}{}$$

```
[Out] -(a-I*b)^2*(B+I*(A-C))*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)
^(1/2))/f+(a+I*b)^2*(I*A-B-I*C)*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2
)/(c+I*d)^(1/2))/f+2*(2*a*b*(A*c-B*d-C*c)+a^2*(B*c+(A-C)*d)-b^2*(B*c+(A-C)*
d))*(c+d*tan(f*x+e))^(1/2)/f+2/3*(a^2*B-b^2*B+2*a*b*(A-C))*(c+d*tan(f*x+e))
^(3/2)/f+2/315*(28*a^2*C*d^2-18*a*b*d*(-7*B*d+2*C*c)+b^2*(8*c^2*C-18*B*c*d+
63*(A-C)*d^2))*(c+d*tan(f*x+e))^(5/2)/d^3/f-2/63*b*(-9*B*b*d-4*C*a*d+4*C*b*
c)*tan(f*x+e)*(c+d*tan(f*x+e))^(5/2)/d^2/f+2/9*C*(a+b*tan(f*x+e))^2*(c+d*ta
n(f*x+e))^(5/2)/d/f
```

Rubi [A] time = 1.73, antiderivative size = 396, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 8, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3647, 3637, 3630, 3528, 3539, 3537, 63, 208}

$$\frac{2(c+d\tan(e+fx))^{5/2}\left(28a^2Cd^2-18abd(2cC-7Bd)+b^2\left(63d^2(A-C)-18Bcd+8c^2C\right)\right)}{315d^3f}+\frac{2\left(a^2B+2ab(A-\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^2*(B + I*(A - C))*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c - I*d]])/f) + ((a + I*b)^2*(I*A - B - I*C)*(c + I*d)^(3/2)*ArcT
anh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(2*a*b*(A*c - c*C - B*d
) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A - C)*d))*Sqrt[c + d*Tan[e + f*x]]
)/f + (2*(a^2*B - b^2*B + 2*a*b*(A - C))*(c + d*Tan[e + f*x])^(3/2))/(3*f)
+ (2*(28*a^2*C*d^2 - 18*a*b*d*(2*c*C - 7*B*d) + b^2*(8*c^2*C - 18*B*c*d + 6
3*(A - C)*d^2))*(c + d*Tan[e + f*x])^(5/2))/(315*d^3*f) - (2*b*(4*b*c*C - 9
*b*B*d - 4*a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2))/(63*d^2*f) + (2*
C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(5/2))/(9*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
```

```
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
```

```
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{2C(a+b\tan(e+fx))^2(c}{9df}$$

$$= -\frac{2b(4bcC-9bBd-4aC}{}$$

$$= \frac{2\left(28a^2Cd^2-18abd(2cC\right.}{}$$

$$= \frac{2\left(a^2B-b^2B+2ab(A-\right.}{3}$$

$$= \frac{2\left(2ab(Ac-cC-Bd)+\right.}{}$$

$$= \frac{2\left(2ab(Ac-cC-Bd)+\right.}{}$$

$$= \frac{2\left(2ab(Ac-cC-Bd)+\right.}{}$$

$$= \frac{2\left(2ab(Ac-cC-Bd)+\right.}{}$$

$$= -\frac{(a-ib)^2(iA+B-iC)(}{}$$

Mathematica [A] time = 6.20, size = 350, normalized size = 0.88

$$\frac{2\left((c+d\tan(e+fx))^{5/2}\left(28a^2Cd^2+18abd(7Bd-2cC)+b^2\left(63d^2(A-C)-18Bcd+8c^2C\right)\right)+\frac{105}{2}d^3(a-ib)^2(iA\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e +
f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*((28*a^2*C*d^2 + 18*a*b*d*(-2*c*C + 7*B*d) + b^2*(8*c^2*C - 18*B*c*d + 6
3*(A - C)*d^2))*(c + d*Tan[e + f*x])^(5/2) + 5*b*d*(-4*b*c*C + 9*b*B*d + 4*
```

```
a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2) + 35*C*d^2*(a + b*Tan[e + f*
x])^2*(c + d*Tan[e + f*x])^(5/2) + (105*(a - I*b)^2*(I*A + B - I*C)*d^3*(-3
*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]] + Sqrt[c +
 d*Tan[e + f*x]]*(4*c - (3*I)*d + d*Tan[e + f*x])))/2 + (105*(a + I*b)^2*((
-I)*A + B + I*C)*d^3*(-3*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/S
qrt[c + I*d]] + Sqrt[c + d*Tan[e + f*x]]*(4*c + (3*I)*d + d*Tan[e + f*x])))
/2))/(315*d^3*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.60, size = 8031, normalized size = 20.28

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) +
 C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a + b \tan \left(e + f x\right)\right)^{2} \left(c + d \tan \left(e + f x\right)\right)^{\frac{3}{2}} \left(A + B \tan \left(e + f x\right) + C \tan^{2} \left(e + f x\right)\right)\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan
(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*(c + d*tan(e + f*x))**(3/2)*(A + B*tan(e +
 f*x) + C*tan(e + f*x)**2), x)
```

3.99 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx\right.$

Optimal. Leaf size=273

$$\frac{2(aB+Ab-bC)(c+d\tan(e+fx))^{3/2}}{3f}+\frac{2\sqrt{c+d\tan(e+fx)}\,(aAd+aBc-aCd+Abc-bBd-bcC)}{f}-\frac{(b+ia)(c}{}$$

```
[Out] -(I*a+b)*(A-I*B-C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/
2))/f+(I*a-b)*(A+I*B-C)*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d
)^(1/2))/f+2*(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b*c)*(c+d*tan(f*x+e))^(1/2)/f
+2/3*(A*b+B*a-C*b)*(c+d*tan(f*x+e))^(3/2)/f-2/35*(-7*B*b*d-7*C*a*d+2*C*b*c)
*(c+d*tan(f*x+e))^(5/2)/d^2/f+2/7*b*C*tan(f*x+e)*(c+d*tan(f*x+e))^(5/2)/d/f
```

Rubi [A] time = 0.88, antiderivative size = 273, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 7, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.156, Rules used = {3637, 3630, 3528, 3539, 3537, 63, 208}

$$\frac{2(aB+Ab-bC)(c+d\tan(e+fx))^{3/2}}{3f}+\frac{2\sqrt{c+d\tan(e+fx)}\,(aAd+aBc-aCd+Abc-bBd-bcC)}{f}-\frac{(b+ia)(c}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C
*Tan[e + f*x]^2),x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c - I*d]])/f) + ((I*a - b)*(A + I*B - C)*(c + I*d)^(3/2)*ArcTanh[Sqrt
[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(A*b*c + a*B*c - b*c*C + a*A*d
- b*B*d - a*C*d)*Sqrt[c + d*Tan[e + f*x]])/f + (2*(A*b + a*B - b*C)*(c + d*
Tan[e + f*x])^(3/2))/(3*f) - (2*(2*b*c*C - 7*b*B*d - 7*a*C*d)*(c + d*Tan[e
+ f*x])^(5/2))/(35*d^2*f) + (2*b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(5/2))
/(7*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx=\frac{2bC\tan(e+fx)(c+d\,\mathrm{ta}}{7df}$$

$$=-\frac{2(2bcC-7bBd-7aCd}{35d}$$

$$=\frac{2(Ab+aB-bC)(c+d\,\mathrm{t}}{3f}$$

$$=\frac{2(Abc+aBc-bcC+aA}{}$$

$$=\frac{2(Abc+aBc-bcC+aA}{}$$

$$=\frac{2(Abc+aBc-bcC+aA}{}$$

$$=\frac{2(Abc+aBc-bcC+aA}{}$$

$$=-\frac{(a-ib)(iA+B-iC)(c}{}$$

Mathematica [A] time = 4.44, size = 260, normalized size = 0.95

$$\frac{\frac{35}{3}d(b+ia)(A-iB-C)\left(\sqrt{c+d\tan(e+fx)}\,(4c+d\tan(e+fx)-3id)-3(c-id)^{3/2}\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)\right)+}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*
x] + C*Tan[e + f*x]^2),x]
```

```
[Out] ((2*(-2*b*c*C + 7*b*B*d + 7*a*C*d)*(c + d*Tan[e + f*x])^(5/2))/d + 10*b*C*T
an[e + f*x]*(c + d*Tan[e + f*x])^(5/2) + (35*(I*a + b)*(A - I*B - C)*d*(-3*
(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]] + Sqrt[c +
```

```
d*Tan[e + f*x]]*(4*c - (3*I)*d + d*Tan[e + f*x])))/3 + (35*((-I)*a + b)*(A
+ I*B - C)*d*(-3*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c +
I*d]] + Sqrt[c + d*Tan[e + f*x]]*(4*c + (3*I)*d + d*Tan[e + f*x])))/3)/(35*
d*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.55, size = 5149, normalized size = 18.86

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [**F(-1)**] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C
*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

sympy [**F**] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a + b \tan \left(e + fx\right)\right) \left(c + d \tan \left(e + fx\right)\right)^{\frac{3}{2}} \left(A + B \tan \left(e + fx\right) + C \tan^{2} \left(e + fx\right)\right) dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*
x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*(c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)**2), x)
```

3.100 $\int (c+d\tan(e+fx))^{3/2} \left(A + B\tan(e+fx) + C\tan^2(e+$

Optimal. Leaf size=187

$$\frac{2(d(A-C)+Bc)\sqrt{c+d\tan(e+fx)}}{f}-\frac{(c-id)^{3/2}(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}-\frac{(c+id)^{3/2}(B-i(A-C))\,\mathrm{t}}{f}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f-
(B-I*(A-C))*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/f+2
*(B*c+(A-C)*d)*(c+d*tan(f*x+e))^(1/2)/f+2/3*B*(c+d*tan(f*x+e))^(3/2)/f+2/5*
C*(c+d*tan(f*x+e))^(5/2)/d/f
```

Rubi [A] time = 0.46, antiderivative size = 187, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 6, integrand size = 35, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.171, Rules used = {3630, 3528, 3539, 3537, 63, 208}

$$\frac{2(d(A-C)+Bc)\sqrt{c+d\tan(e+fx)}}{f}-\frac{(c-id)^{3/2}(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f}-\frac{(c+id)^{3/2}(B-i(A-C))\,\mathrm{t}}{f}$$

Antiderivative was successfully verified.

```
[In] Int[(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/f) - ((B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c + I*d]])/f + (2*(B*c + (A - C)*d)*Sqrt[c + d*Tan[e + f*x]])/f +
 (2*B*(c + d*Tan[e + f*x])^(3/2))/(3*f) + (2*C*(c + d*Tan[e + f*x])^(5/2))/
(5*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int (c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{2C(c+d\tan(e+fx))^{5/2}}{5df}+\int (A-C+B\tan$$

$$= \frac{2B(c+d\tan(e+fx))^{3/2}}{3f}+\frac{2C(c+d\tan(e+}{5df}$$

$$= \frac{2(Bc+(A-C)d)\sqrt{c+d\tan(e+fx)}}{f}+\frac{2B(c}{}$$

$$= \frac{2(Bc+(A-C)d)\sqrt{c+d\tan(e+fx)}}{f}+\frac{2B(c}{}$$

$$= \frac{2(Bc+(A-C)d)\sqrt{c+d\tan(e+fx)}}{f}+\frac{2B(c}{}$$

$$= \frac{2(Bc+(A-C)d)\sqrt{c+d\tan(e+fx)}}{f}+\frac{2B(c}{}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+f}}{\sqrt{c-id}}\right.}{f}$$

Mathematica [A] time = 1.24, size = 202, normalized size = 1.08

$$\frac{5(iA+B-iC)\left(\sqrt{c+d\tan(e+fx)}\,(4c+d\tan(e+fx)-3id)-3(c-id)^{3/2}\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)\right)+5(-iA+B}{15f}$$

Antiderivative was successfully verified.

```
[In] Integrate[(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2
),x]
```

```
[Out] ((6*C*(c + d*Tan[e + f*x])^(5/2))/d + 5*(I*A + B - I*C)*(-3*(c - I*d)^(3/2)
*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]] + Sqrt[c + d*Tan[e + f*x]]
*(4*c - (3*I)*d + d*Tan[e + f*x])) + 5*((-I)*A + B + I*C)*(-3*(c + I*d)^(3/
2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]] + Sqrt[c + d*Tan[e + f*x
]]*(4*c + (3*I)*d + d*Tan[e + f*x])))/(15*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.41, size = 2517, normalized size = 13.46

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] 1/4/d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1
/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2-1/4/d/f*ln(d*tan(f
*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2
))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2-2*d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*
arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2
)^(1/2)-2*c)^(1/2))*A*c+d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d
^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2
))*A*(c^2+d^2)^(1/2)-d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)
^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*
C*(c^2+d^2)^(1/2)+2*d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^
(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C
*c+d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c
^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*(c^2+d^2)^(1/2)-
1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+
d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)*c+1
/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c
+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)*c-1/
4/d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-
c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2-1/2/f*ln(d*tan(f*x+e
)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B
*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c-1/4/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^
2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*
c)^(1/2)*(c^2+d^2)^(1/2)+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*t
```

```
an(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/
2))*B*c^2-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)
^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*c^2-1/4/d
/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(
c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*c+1/4/d/f*l
n((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+
d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*c-1/4/d/f*ln((c
+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)
^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*c+1/4/f*ln(d*tan(f*
x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2)
)*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)+2*d/f*A*(c+d*tan(f*x+e))^
(1/2)-2*d/f*C*(c+d*tan(f*x+e))^(1/2)+2/f*B*c*(c+d*tan(f*x+e))^(1/2)+1/2/f*l
n((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+
d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c-1/4*d/f*ln(d*tan(f*x+e)+c+(c+
d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^
2+d^2)^(1/2)+2*c)^(1/2)+1/4*d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2
*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2
)-d^2/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(
c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B+d^2/f/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e
))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B-1/4*d/f*ln((c+d*tan(f*x+e))^(1/2
)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d
^2)^(1/2)+2*c)^(1/2)+1/4*d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2
*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+1
/4/d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/
2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*c+2/5*C
*(c+d*tan(f*x+e))^(5/2)/d/f+1/4/d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^
(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^
(1/2)*c^2-d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2
)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*(c^2+d^2)
^(1/2)-2*d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)
+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*c+2*d/f/(2
*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(
1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*c+2/3*B*(c+d*tan(f*x+e))^
(3/2)/f
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^2 + B\tan\left(fx+e\right) + A\right)\left(d\tan\left(fx+e\right) + c\right)^{\frac{3}{2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)^(3/2
), x)
```

mupad [B] time = 44.87, size = 4260, normalized size = 22.78

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] ((2*C*c^2)/(d*f) - (2*C*(d^3*f + c^2*d*f))/(d^2*f^2))*(c + d*tan(e + f*x))^
(1/2) - log((((16*c*d^2*(((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + B^2*c^3*f^
2 - 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(B*c^2 + B*d^2 + f*(((-B^4*d^2*f^4*(3*c^2 -
 d^2)^2)^(1/2) + B^2*c^3*f^2 - 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f
*x))^(1/2)))/f - (16*B^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*
d^2))/f^2)*(((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + B^2*c^3*f^2 - 3*B^2*c*d
^2*f^2)/f^4)^(1/2))/2 - (8*B^3*d^2*(c^2 - d^2)*(c^2 + d^2)^2)/f^3)*(((6*B^4
*c^2*d^4*f^4 - B^4*d^6*f^4 - 9*B^4*c^4*d^2*f^4)^(1/2) + B^2*c^3*f^2 - 3*B^2
*c*d^2*f^2)/(4*f^4))^(1/2) - log((((16*c*d^2*(-((-B^4*d^2*f^4*(3*c^2 - d^2)
^2)^(1/2) - B^2*c^3*f^2 + 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(B*c^2 + B*d^2 + f*(-
((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - B^2*c^3*f^2 + 3*B^2*c*d^2*f^2)/f^4)
^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f - (16*B^2*d^2*(c + d*tan(e + f*x))^(1
/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(-((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) -
 B^2*c^3*f^2 + 3*B^2*c*d^2*f^2)/f^4)^(1/2))/2 - (8*B^3*d^2*(c^2 - d^2)*(c^2
 + d^2)^2)/f^3)*(-((6*B^4*c^2*d^4*f^4 - B^4*d^6*f^4 - 9*B^4*c^4*d^2*f^4)^(1
/2) - B^2*c^3*f^2 + 3*B^2*c*d^2*f^2)/(4*f^4))^(1/2) + log((((16*c*d^2*(((-B
^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + B^2*c^3*f^2 - 3*B^2*c*d^2*f^2)/f^4)^(1/
2)*(B*c^2 + B*d^2 - f*(((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + B^2*c^3*f^2
- 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f + (16*B^2*d^2*
(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(((-B^4*d^2*f^4*(3
*c^2 - d^2)^2)^(1/2) + B^2*c^3*f^2 - 3*B^2*c*d^2*f^2)/f^4)^(1/2))/2 - (8*B^
3*d^2*(c^2 - d^2)*(c^2 + d^2)^2)/f^3)*((6*B^4*c^2*d^4*f^4 - B^4*d^6*f^4 - 9
*B^4*c^4*d^2*f^4)^(1/2)/(4*f^4) + (B^2*c^3)/(4*f^2) - (3*B^2*c*d^2)/(4*f^2)
)^(1/2) + log((((16*c*d^2*(-((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - B^2*c^3
*f^2 + 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(B*c^2 + B*d^2 - f*(-((-B^4*d^2*f^4*(3*c
^2 - d^2)^2)^(1/2) - B^2*c^3*f^2 + 3*B^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e
 + f*x))^(1/2)))/f + (16*B^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*
c^2*d^2))/f^2)*(-((-B^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - B^2*c^3*f^2 + 3*B^
2*c*d^2*f^2)/f^4)^(1/2))/2 - (8*B^3*d^2*(c^2 - d^2)*(c^2 + d^2)^2)/f^3)*((B
^2*c^3)/(4*f^2) - (6*B^4*c^2*d^4*f^4 - B^4*d^6*f^4 - 9*B^4*c^4*d^2*f^4)^(1/
2)/(4*f^4) - (3*B^2*c*d^2)/(4*f^2))^(1/2) - log((((16*d^2*(-((-A^4*d^2*f^4*
(3*c^2 - d^2)^2)^(1/2) + A^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/f^4)^(1/2)*(A*d^3 +
 A*c^2*d + c*f*(-((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + A^2*c^3*f^2 - 3*A^
2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f + (16*A^2*d^2*(c + d
*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(-((-A^4*d^2*f^4*(3*c^2
```

```
- d^2)^2)^(1/2) + A^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/f^4)^(1/2))/2 - (16*A^3*c*
d^3*(c^2 + d^2)^2)/f^3)*(-((6*A^4*c^2*d^4*f^4 - A^4*d^6*f^4 - 9*A^4*c^4*d^2
*f^4)^(1/2) + A^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/(4*f^4))^(1/2) - log((((16*d^2
*(((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - A^2*c^3*f^2 + 3*A^2*c*d^2*f^2)/f^
4)^(1/2)*(A*d^3 + A*c^2*d + c*f*(((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - A^
2*c^3*f^2 + 3*A^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f + (1
6*A^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(((-A^4*
d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - A^2*c^3*f^2 + 3*A^2*c*d^2*f^2)/f^4)^(1/2))
/2 - (16*A^3*c*d^3*(c^2 + d^2)^2)/f^3)*(((6*A^4*c^2*d^4*f^4 - A^4*d^6*f^4 -
 9*A^4*c^4*d^2*f^4)^(1/2) - A^2*c^3*f^2 + 3*A^2*c*d^2*f^2)/(4*f^4))^(1/2) +
 log((((16*d^2*(((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - A^2*c^3*f^2 + 3*A^2
*c*d^2*f^2)/f^4)^(1/2)*(A*d^3 + A*c^2*d - c*f*(((-A^4*d^2*f^4*(3*c^2 - d^2)
^2)^(1/2) - A^2*c^3*f^2 + 3*A^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^
(1/2)))/f - (16*A^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))
/f^2)*(((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - A^2*c^3*f^2 + 3*A^2*c*d^2*f^
2)/f^4)^(1/2))/2 - (16*A^3*c*d^3*(c^2 + d^2)^2)/f^3)*((6*A^4*c^2*d^4*f^4 -
A^4*d^6*f^4 - 9*A^4*c^4*d^2*f^4)^(1/2)/(4*f^4) - (A^2*c^3)/(4*f^2) + (3*A^2
*c*d^2)/(4*f^2))^(1/2) + log((((16*d^2*(-((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1
/2) + A^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/f^4)^(1/2)*(A*d^3 + A*c^2*d - c*f*(-((
-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + A^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/f^4)^(
1/2)*(c + d*tan(e + f*x))^(1/2)))/f - (16*A^2*d^2*(c + d*tan(e + f*x))^(1/2
)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(-((-A^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + A
^2*c^3*f^2 - 3*A^2*c*d^2*f^2)/f^4)^(1/2))/2 - (16*A^3*c*d^3*(c^2 + d^2)^2)/
f^3)*((3*A^2*c*d^2)/(4*f^2) - (A^2*c^3)/(4*f^2) - (6*A^4*c^2*d^4*f^4 - A^4*
d^6*f^4 - 9*A^4*c^4*d^2*f^4)^(1/2)/(4*f^4))^(1/2) - log((16*C^3*c*d^3*(c^2
+ d^2)^2)/f^3 - (((16*d^2*(-((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + C^2*c^3
*f^2 - 3*C^2*c*d^2*f^2)/f^4)^(1/2)*(C*d^3 + C*c^2*d - c*f*(-((-C^4*d^2*f^4*
(3*c^2 - d^2)^2)^(1/2) + C^2*c^3*f^2 - 3*C^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*t
an(e + f*x))^(1/2)))/f - (16*C^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4
- 6*c^2*d^2))/f^2)*(-((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + C^2*c^3*f^2 -
3*C^2*c*d^2*f^2)/f^4)^(1/2))/2)*(-((6*C^4*c^2*d^4*f^4 - C^4*d^6*f^4 - 9*C^4
*c^4*d^2*f^4)^(1/2) + C^2*c^3*f^2 - 3*C^2*c*d^2*f^2)/(4*f^4))^(1/2) - log((
16*C^3*c*d^3*(c^2 + d^2)^2)/f^3 - (((16*d^2*(((-C^4*d^2*f^4*(3*c^2 - d^2)^2
)^(1/2) - C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/f^4)^(1/2)*(C*d^3 + C*c^2*d - c*f*
(((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/f^4
)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f - (16*C^2*d^2*(c + d*tan(e + f*x))^(
1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) -
 C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/f^4)^(1/2))/2)*(((6*C^4*c^2*d^4*f^4 - C^4*d
^6*f^4 - 9*C^4*c^4*d^2*f^4)^(1/2) - C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/(4*f^4))
^(1/2) + log((16*C^3*c*d^3*(c^2 + d^2)^2)/f^3 - (((16*d^2*(((-C^4*d^2*f^4*(
3*c^2 - d^2)^2)^(1/2) - C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/f^4)^(1/2)*(C*d^3 +
C*c^2*d + c*f*(((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) - C^2*c^3*f^2 + 3*C^2*
c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/f + (16*C^2*d^2*(c + d*t
an(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*(((-C^4*d^2*f^4*(3*c^2 - d
^2)^2)^(1/2) - C^2*c^3*f^2 + 3*C^2*c*d^2*f^2)/f^4)^(1/2))/2)*((6*C^4*c^2*d^
```

```
4*f^4 - C^4*d^6*f^4 - 9*C^4*c^4*d^2*f^4)^(1/2)/(4*f^4) - (C^2*c^3)/(4*f^2)
+ (3*C^2*c*d^2)/(4*f^2))^(1/2) + log((16*C^3*c*d^3*(c^2 + d^2)^2)/f^3 - (((
16*d^2*(-((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + C^2*c^3*f^2 - 3*C^2*c*d^2*
f^2)/f^4)^(1/2)*(C*d^3 + C*c^2*d + c*f*(-((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1
/2) + C^2*c^3*f^2 - 3*C^2*c*d^2*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2))
)/f + (16*C^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^4 + d^4 - 6*c^2*d^2))/f^2)*
(-((-C^4*d^2*f^4*(3*c^2 - d^2)^2)^(1/2) + C^2*c^3*f^2 - 3*C^2*c*d^2*f^2)/f^
4)^(1/2))/2)*((3*C^2*c*d^2)/(4*f^2) - (C^2*c^3)/(4*f^2) - (6*C^4*c^2*d^4*f^
4 - C^4*d^6*f^4 - 9*C^4*c^4*d^2*f^4)^(1/2)/(4*f^4))^(1/2) + (2*B*(c + d*tan
(e + f*x))^(3/2))/(3*f) + (2*A*d*(c + d*tan(e + f*x))^(1/2))/f + (2*B*c*(c
+ d*tan(e + f*x))^(1/2))/f + (2*C*(c + d*tan(e + f*x))^(5/2))/(5*d*f)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(c + d \tan \left(e + f x\right)\right)^{\frac{3}{2}} \left(A + B \tan \left(e + f x\right) + C \tan^{2} \left(e + f x\right)\right)\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2), x)
```

3.101 $\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$

Optimal. Leaf size=271

$$-\frac{2(bc-ad)^{3/2}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{5/2}f\left(a^2+b^2\right)}-\frac{(c-id)^{3/2}(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)}-\frac{(c$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a
-I*b)/f-(A+I*B-C)*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/(I*a-b)/f-2*(A*b^2-a*(B*b-C*a))*(-a*d+b*c)^(3/2)*arctanh(b^(1/2)*(c+d*ta
n(f*x+e))^(1/2)/(-a*d+b*c)^(1/2))/b^(5/2)/(a^2+b^2)/f+2*(B*b*d-C*a*d+C*b*c)
*(c+d*tan(f*x+e))^(1/2)/b^2/f+2/3*C*(c+d*tan(f*x+e))^(3/2)/b/f
```

Rubi [A] time = 1.81, antiderivative size = 271, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3647, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{2(bc-ad)^{3/2}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{5/2}f\left(a^2+b^2\right)}-\frac{(c-id)^{3/2}(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)}-\frac{(c$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x]),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/((a - I*b)*f)) - ((A + I*B - C)*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((I*a - b)*f) - (2*(A*b^2 - a*(b*B - a*C))*(
b*c - a*d)^(3/2)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]
])/(b^(5/2)*(a^2 + b^2)*f) + (2*(b*c*C + b*B*d - a*C*d)*Sqrt[c + d*Tan[e +
f*x]])/(b^2*f) + (2*C*(c + d*Tan[e + f*x])^(3/2))/(3*b*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
```

```
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \frac{2C(c+d\tan(e+fx))^{3/2}}{3bf} + \frac{2\int \frac{\sqrt{c+d\tan(e+fx)}}{\cdots}}{\cdots}$$

$$= \frac{2(bcC+bBd-aCd)\sqrt{c+d\tan(e+fx)}}{b^2f} + \frac{2\cdots}{\cdots}$$

$$= \frac{2(bcC+bBd-aCd)\sqrt{c+d\tan(e+fx)}}{b^2f} + \frac{2\cdots}{\cdots}$$

$$= \frac{2(bcC+bBd-aCd)\sqrt{c+d\tan(e+fx)}}{b^2f} + \frac{2\cdots}{\cdots}$$

$$= \frac{2(bcC+bBd-aCd)\sqrt{c+d\tan(e+fx)}}{b^2f} + \frac{2\cdots}{\cdots}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)(bc-ad)^{3/2}\tanh^{-1}\left(\cdots\right.}{b^{5/2}\left(a^2+b^2\right)f}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+\cdots}}{\sqrt{c-id}}\right.}{(a-ib)f}$$

Mathematica [A] time = 2.55, size = 266, normalized size = 0.98

$$\frac{\frac{3ib\left((a-ib)(c+id)^{3/2}(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)-(a+ib)(c-id)^{3/2}(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)\right)}{a^2+b^2} - \frac{6(bc-ad)^{3/2}\left(a(aC-bB)+Ab^2\right)\tanh^{-1}\left(\frac{\sqrt{b\cdots}}{\cdots}\right.}{b^{3/2}\left(a^2+b^2\right)}\cdots}{3bf}$$

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x]),x]
```

```
[Out] (((3*I)*b*(-((a + I*b)*(A - I*B - C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan
[e + f*x]]/Sqrt[c - I*d]]) + (a - I*b)*(A + I*B - C)*(c + I*d)^(3/2)*ArcTan
h[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]))/(a^2 + b^2) - (6*(A*b^2 + a*(-(
b*B) + a*C))*(b*c - a*d)^(3/2)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/S
qrt[b*c - a*d]])/(b^(3/2)*(a^2 + b^2)) + (6*(b*c*C + b*B*d - a*C*d)*Sqrt[c
+ d*Tan[e + f*x]])/b + 2*C*(c + d*Tan[e + f*x])^(3/2))/(3*b*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.83, size = 6055, normalized size = 22.34

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [B] time = 58.88, size = 106783, normalized size = 394.03

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x)),x)
```

```
[Out] atan((((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f
^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 +
4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 -
 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*
f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*
d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (3
2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 1
6*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*
b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^
4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b
^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^
2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(
16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d
^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*
f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24
*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2
- 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^
2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2
*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^
2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^
2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2
*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*
f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8
*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2
 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9
*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^
2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*
B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*
d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*
```

a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^

3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)
/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2
*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8
*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f
^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^
7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*
B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*
c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*
d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2
*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2
- 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (
32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*
f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*
f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2
+ 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d
^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7
*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 -
24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c
^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34
*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*(-(((8*B^
2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f
^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^
4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2)
)^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^
2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4
+ b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^
2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^
7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*
c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^
6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B
^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^
2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b
^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2
- 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b
^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2
- 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2
*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2
+ 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(
B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f
^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*
b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^
4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16
+ 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6
*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 +
12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 -

48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14
+ 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 -
48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*(-(((8*B^2*a^2*c^3*
f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^
2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32
*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) -
4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^
2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4
+ 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11
*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 +
8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 -
12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4
+ 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d
^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c
^3*d^9*f^4))/(b*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c^3*f^
2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*
b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a
^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*
B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*
f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 +
2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^
6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8
*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 +
24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4)
)*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*
a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*
f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B
^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*
f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(
16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(
1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10
*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f
^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f
^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c
*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a
^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*
f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*
b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2
- 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2
*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2
+ 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(
B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f
^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*
b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^
4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^

7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^

2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/

(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (64*(B^5*a^3*b^2*d^18 - B^5*a^5*d^18 - B^5*a^5*c^2*d^16 + B^5*a^5*c^4*d^14 + B^5*a^5*c^6*d^12 - 8*B^5*a^2*b^3*c^3*d^15 - 14*B^5*a^2*b^3*c^5*d^13 - 12*B^5*a^2*b^3*c^7*d^11 - 4*B^5*a^2*b^3*c^9*d^9 + 3*B^5*a^3*b^2*c^2*d^16 + 9*B^5*a^3*b^2*c^4*d^14 + 13*B^5*a^3*b^2*c^6*d^12 + 6*B^5*a^3*b^2*c^8*d^10 + 2*B^5*a^4*b*c*d^17 + B^5*a*b^4*c^2*d^16 + 4*B^5*a*b^4*c^4*d^14 + 6*B^5*a*b^4*c^6*d^12 + 4*B^5*a*b^4*c^8*d^10 + B^5*a*b^4*c^10*d^8 - 2*B^5*a^2*b^3*c*d^17 - 6*B^5*a^4*b*c^5*d^13 - 4*B^5*a^4*b*c^7*d^11))/(b*f^5)))*(-(((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) - 4*B^2*a^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 + 8*B^2*a*b*d^3*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 - 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i - atan((((((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 +

4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*
c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))
^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^
6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d
^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4
*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a
^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2
- 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^
2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3
*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*((((8*A^2*a^2*c^3
*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A
^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 3
2*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) -
4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d
^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^
4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 +
4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c
^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^
12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*
d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 -
8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)
*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^
2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^
4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4
*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^
2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16
*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(4*A*a^2*b^6*d^12
*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4
+ 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 -
16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8
*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*
c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*
b^7*c*d^11*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2
- 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b
^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^
2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A
^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f
^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 +
2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*
d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^
4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*
a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8
*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^
2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16
*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d

^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12 *A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4* f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4* A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8* A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c ^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28 *A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5* d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c *d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^ 4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a ^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a* b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 5 3*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13 *f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2* b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^ 5*d^10*f^2))/f^5)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^ 3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2) ^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^ 4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a *b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d* tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b ^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^ 3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b ^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2 *d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f ^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48* A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^ 6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4 *A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c* d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^ 2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3* d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6* c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a ^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 +

4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 +

12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2* d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (64*(A^5*a^2*b^2*d^18 + A^5*b^4*c^2*d^16 + 5*A^5*b^4*c^4*d^14 + 7*A^5*b^4*c^6*d^12 + 3*A^5*b^4*c^8*d^10 + 9*A^5*a^2*b^2*c^2*d^16 + 15*A^5*a^2*b^2*c^4*d^14 + 7*A^5*a^2*b^2*c^6*d^12 - 2*A^5*a*b^3*c*d^17 - 2*A^5*a^3*b*c*d^17 - 12*A^5*a*b^3*c^3*d^15 - 18*A^5*a*b^3*c^5*d^13 - 8*A^5*a*b^3*c^7*d^11 - 4*A^5*a^3*b*c^3*d^15 - 2*A^5*a^3*b*c^5*d^13))/f^5))*((((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*A^2*b^2*c^3*f^2 + 8*A^2*a*b*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*A^2*b^2*c*d^2*f^2 - 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i - atan((((((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d

^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f
^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*
a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A
^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d
^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*
f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2
*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 1
2*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*
b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 -
14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2
- 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 +
20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^
12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a
^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f
^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b
^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c^3*f^2 -
8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2
*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*
b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2
*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2
+ 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*
a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f
^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 +
4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25
*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*
f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4
*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*
A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^
9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*(-(
((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c
*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 +
16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^
4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 -
12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a
^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*
(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^1
0 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4
*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A
^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*
a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*
c^4*d^12))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3
*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^
2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4
*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 -
8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*

b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(
4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b
^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*
b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A
*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4
 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^
10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(-(((
8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d
^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 1
6*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*
d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 1
2*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4
*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*
f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a
^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^
8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9
f^4))/f^4)(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2
 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4
- (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2
*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^
2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^
2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e
 + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b
^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b
^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2
*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2
+ 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4
*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 -
2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4
*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3
*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^
2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4
*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 -
8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*
b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*
b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^
3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d
^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*
b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4
*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*
d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a
^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2
 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3
*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 4
8*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*

```
c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 -
 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*
c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))
^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14
+ 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b
*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*
c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c
^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^1
5 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)*(-(((8*A^2*a^2*c^3*
f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^
2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32
*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) +
4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^
2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4
 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*
d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*
f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^1
1*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5
*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*
a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^
5 - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f
^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*
A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^
6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4
*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*
d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(
1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*
b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*
d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4
+ 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/f^4)*(-(((8*A^2*a^2*c^3*f^2
- 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^
2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2
*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^
2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^
2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2
*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*
f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^
8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*
f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5
*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*
A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*
d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*
a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c^3*
f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^
2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32
```

*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) +
4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^
2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4
 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d
^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*
f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2
 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*
d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^
2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2
+ 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c
^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5
)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*
a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*
f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A
^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*
f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(
16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(
1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^
6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 7
2*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13
- 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8
*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a
^4*b*c^4*d^12))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*
b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*
f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 +
3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f
^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A
^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32
*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A
*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^
3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24
*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f
^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*
d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 + (32*(c + d*tan(e + f*x))^(1/2)*(-(
((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c
*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 +
 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^
4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 -
 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a
^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^1
0*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40
*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*
b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d
^9*f^4))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f
^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/

4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c ^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8* A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b* c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan (e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2 *b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2 *b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A ^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^ 2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b ^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c ^4*d^9*f^2))/f^4)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d ^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A ^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2* a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(23*A^ 3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3* A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c *d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3* a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^ 5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3 *a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f ^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c ^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^ 4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^ 2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^1 4 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4 *b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^ 3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4 *c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d ^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4)*(-(((8*A^2*a^2*c^ 3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24* A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*f^2 - 12*A^2*a^2*c* d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f ^4 + 2*a^2*b^2*f^4)))^(1/2) + (64*(A^5*a^2*b^2*d^18 + A^5*b^4*c^2*d^16 + 5* A^5*b^4*c^4*d^14 + 7*A^5*b^4*c^6*d^12 + 3*A^5*b^4*c^8*d^10 + 9*A^5*a^2*b^2* c^2*d^16 + 15*A^5*a^2*b^2*c^4*d^14 + 7*A^5*a^2*b^2*c^6*d^12 - 2*A^5*a*b^3*c *d^17 - 2*A^5*a^3*b*c*d^17 - 12*A^5*a*b^3*c^3*d^15 - 18*A^5*a*b^3*c^5*d^13

```
- 8*A^5*a*b^3*c^7*d^11 - 4*A^5*a^3*b*c^3*d^15 - 2*A^5*a^3*b*c^5*d^13))/f^5)
)*(-(((8*A^2*a^2*c^3*f^2 - 8*A^2*b^2*c^3*f^2 - 16*A^2*a*b*d^3*f^2 - 24*A^2*
a^2*c*d^2*f^2 + 24*A^2*b^2*c*d^2*f^2 + 48*A^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*
f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(A^4*c^6 + A^4*d^6 + 3*A^4*c^2*d^4 + 3*A
^4*c^4*d^2))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*A^2*b^2*c^3*f^2 - 8*A^2*a*b*d^3*
f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*A^2*b^2*c*d^2*f^2 + 24*A^2*a*b*c^2*d*f^2)/(
16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i - ((4*C*c)/(b*f) + (2*C*(
a*d*f - 3*b*c*f))/(b^2*f^2))*(c + d*tan(e + f*x))^(1/2) + atan((((((32*(4*
B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*
d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8
*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^
11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6
*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^
5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (32*(c + d*tan(e +
f*x))^(1/2)*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2
- 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 -
 (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*
d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2
*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2
*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 +
 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*
c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^
2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4
 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2
 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^
2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6
+ B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B
^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^
2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/
2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3
*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7
*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^1
2*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b
^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B
^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8
*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4
*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^
2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*(((
(8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*
d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 +
16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4
*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 -
12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^
4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B
^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B
```

^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*

B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3
*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/
(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^
(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^1
0*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*
f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*
f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*
c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*
a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11
*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4
*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2
 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2
*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2
 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(
B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f
^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*
b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^
4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^
7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^
7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*
d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*
b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4
*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^
13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a
^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*
f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*
a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10
*f^2))/(b*f^5))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*
f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2
/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*
c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8
*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b
*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*ta
n(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*
B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*
d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 -
 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 +
 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12
- 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*
B^4*a^5*b*c^5*d^11))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 1
6*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*
b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^
4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b
^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^
2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1

i)/((((((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4
 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*
B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4
*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^
4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^
10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (32*
(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B
^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c
^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d
^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*
c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 +
 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*
b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10
*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4
 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^
5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*
B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*
d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2
f^4)(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^
2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 +
12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2
*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2
+ 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2
 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 2
2*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2
 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*
d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*
a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*
f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*
b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2
))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2
- 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 -
 (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*
d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2
*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2
*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b
^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2
*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d
^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^
12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b
^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B
^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^1
1*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*
a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13
*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a

```
^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5))*((((8*B^2*a^2*
c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 2
4*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4
+ 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2
) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*
c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4
*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^1
6 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c
^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4
*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4
*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^
4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*
B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*
((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2
*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4
 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*
c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2
 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*
(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(4*B*a*b^8*d^12*f^4 -
 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9
*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^
7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*
b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*
B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4
- 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*((((8*
B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2
*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*
b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^
2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*
B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f
^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f
^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a
^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^
9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9
*f^4))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*
f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2
/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*
c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8
*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b
*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*ta
n(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2
*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2
*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b
^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*
B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*
```

f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*
b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2
- 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3
*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4))*((((8*B^2*a^2*c^3*f^2
- 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b
^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^
2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B
^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f
^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 +
2*a^2*b^2*f^4)))^(1/2) + (32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^
2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^
2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 -
B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^
2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*
d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^
2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^
2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*
b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f
^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*
b^2*c^5*d^10*f^2))/(b*f^5))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*
B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*
c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*
d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2
*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2
+ 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (
32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c
^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*
B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^
4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^
3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b
^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c
^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4))*((((8*B^2*a^2*c^3*f^2 - 8*B^2*b^2
*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2
+ 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(
B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4*d^2))^(1/2) + 4*B^2*a^2*c^3*f
^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*B^2*
b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^
4)))^(1/2) + (64*(B^5*a^3*b^2*d^18 - B^5*a^5*d^18 - B^5*a^5*c^2*d^16 + B^5*
a^5*c^4*d^14 + B^5*a^5*c^6*d^12 - 8*B^5*a^2*b^3*c^3*d^15 - 14*B^5*a^2*b^3*c
^5*d^13 - 12*B^5*a^2*b^3*c^7*d^11 - 4*B^5*a^2*b^3*c^9*d^9 + 3*B^5*a^3*b^2*c
^2*d^16 + 9*B^5*a^3*b^2*c^4*d^14 + 13*B^5*a^3*b^2*c^6*d^12 + 6*B^5*a^3*b^2*
c^8*d^10 + 2*B^5*a^4*b*c*d^17 + B^5*a*b^4*c^2*d^16 + 4*B^5*a*b^4*c^4*d^14 +
6*B^5*a*b^4*c^6*d^12 + 4*B^5*a*b^4*c^8*d^10 + B^5*a*b^4*c^10*d^8 - 2*B^5*a
^2*b^3*c*d^17 - 6*B^5*a^4*b*c^5*d^13 - 4*B^5*a^4*b*c^7*d^11))/(b*f^5)))*(((
(8*B^2*a^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 - 16*B^2*a*b*d^3*f^2 - 24*B^2*a^2*c*

d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 + 48*B^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 +
16*b^4*f^4 + 32*a^2*b^2*f^4)*(B^4*c^6 + B^4*d^6 + 3*B^4*c^2*d^4 + 3*B^4*c^4
*d^2))^(1/2) + 4*B^2*a^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 - 8*B^2*a*b*d^3*f^2 -
12*B^2*a^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 + 24*B^2*a*b*c^2*d*f^2)/(16*(a^
4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i - atan(((((((32*(12*C*a^2*b^9*d
^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10
*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11
*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^
4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*
b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16
*C*a*b^10*c*d^11*f^4))/(b^3*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*((((8*C^2
*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^
2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4
*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))
^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2
*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4
+ b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4
 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2
*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^1
1*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9
*f^4))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^
3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)
^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^
4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 +
 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a
*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*
tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C
^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16
*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C
^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^
2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2
*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2
*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10
*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^
5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2
 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^
2*c^3*d^10*f^2))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*
C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*
c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*
d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2
*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2
 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (
32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 1
6*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^
3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d

^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b

```
^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2
))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C
^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^
4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*C^
2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10
*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C
^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 5
0*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*
f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8
*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4
*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5
*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^
2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^1
1*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4
))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*
a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*
f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C
^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*
f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(
16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 +
C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 +
4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*
b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*
d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*
c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3
*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f
^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b
^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 +
 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^
5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2
 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3
*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))
/(b^3*f^5))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2
- 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 -
 (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*
d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2
*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2
*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e
+ f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*
a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10
 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C
^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C
^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12
*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*
a^7*b*c^5*d^11))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*
```

C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*
c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*
d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2
*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2
- 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)
/((((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^1
2*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*
f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*
d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^
7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*
a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) - (32*(c + d*tan(
e + f*x))^(1/2)*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*
f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2
/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*
c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8
*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b
*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f
^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*
b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*
b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*
d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^
2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^
2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*
(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*
f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2
*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f
^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^
2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 1
4*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C
^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 -
28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^
12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2
*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2
+ 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5
*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 +
8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^
4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2
- 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^
2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2
*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^
2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^
2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2
*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16
*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 -
C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b

^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/

4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4))*((((8*C^2*a^2*c^3*f^2 - 8

*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (64*(C^5*a^4*b^3*d^18 + 4*C^5*a^7*c^3*d^15 + 2*C^5*a^7*c^5*d^13 - C^5*a^6*b*d^18 + 2*C^5*a^7*c*d^17 + C^5*a^2*b^5*c^2*d^16 + 4*C^5*a^2*b^5*c^4*d^14 + 6*C^5*a^2*b^5*c^6*d^12 + 4*C^5*a^2*b^5*c^8*d^10 + C^5*a^2*b^5*c^10*d^8 - 8*C^5*a^3*b^4*c^3*d^15 - 12*C^5*a^3*b^4*c^5*d^13 - 8*C^5*a^3*b^4*c^7*d^11 - 2*C^5*a^3*b^4*c^9*d^9 + 3*C^5*a^4*b^3*c^2*d^16 + C^5*a^4*b^3*c^4*d^14 - 3*C^5*a^4*b^3*c^6*d^12 - 2*C^5*a^4*b^3*c^8*d^10 + 12*C^5*a^5*b^2*c^3*d^15 + 18*C^5*a^5*b^2*c^5*d^13 + 8*C^5*a^5*b^2*c^7*d^11 - 2*C^5*a^3*b^4*c*d^17 + 2*C^5*a^5*b^2*c*d^17 - 9*C^5*a^6*b*c^2*d^16 - 15*C^5*a^6*b*c^4*d^14 - 7*C^5*a^6*b*c^6*d^12))/(b^3*f^5)))*((((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) - 4*C^2*a^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 + 8*C^2*a*b*d^3*f^2 + 12*C^2*a^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 - 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*2i - atan((((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) - (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^1

2*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i - (((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9

```
*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a
^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 +
12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) + (32*(c + d
*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*
b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*
f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 +
3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f
^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C
^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*
d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4
 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 -
8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*
b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8
*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c
*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^
2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a
^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 +
 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^
2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2
 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*
f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2
 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^
9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b
^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*
C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d
^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*
a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10
*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7
*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4))*(-(((8*C^2*a^2*
c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 2
4*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4
+ 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2
) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*
c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4
*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*
f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^1
3*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 +
 C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*
a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C
^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f
^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b
^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 +
 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c
^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 1
44*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4
```

*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 5
6*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*(-(((
8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d
^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 1
6*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*
d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 1
2*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4
*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x))^(1/2)*(2
*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4
*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d
^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d
^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d
^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6
*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))
/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2
 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4
- (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2
*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^
2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^
2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*1i)/((((((32*(12*C
*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^1
1*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*
b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*
a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4
+ 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^1
0*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) - (32*(c + d*tan(e + f*x))^(1/2)
*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a
^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f
^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^
4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f
^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(1
6*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^
10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f
^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^
4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^
7*b^5*c*d^9*f^4))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 1
6*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*
b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^
4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b
^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^
2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) -
 (32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^1
3*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d
^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^1
2*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b

^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (((((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a

^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 3
2*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f
^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d
^11*f^4))/(b^3*f^5) + (32*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^2*c^3*f^2
 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b
^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^
2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C
^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f
^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 +
2*a^2*b^2*f^4)))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^
8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^
8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4
+ 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^3*
f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*
C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*
a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 +
 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*
d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^
2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (32*(c + d*tan(e + f*x
))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3
*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*
d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^
2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a
^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 +
 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^
2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C
^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d
^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a
^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*
f^2))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*C^2*a*b*d^
3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*c^2*d*f^2)
^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*d^6 + 3*C^
4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 -
 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 + 24*C^2*a
*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) - (32*(4*C^3*
a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*
b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*
d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 -
28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f
^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^
14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2
*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2
 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^
6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2
- 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b

```
^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 -
 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2
*c^4*d^11*f^2))/(b^3*f^5))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^2*c^3*f^2 - 16*
C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^2 + 48*C^2*a*b*
c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*(C^4*c^6 + C^4*
d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*f^2 - 4*C^2*b^2
*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2
+ 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f^4)))^(1/2) + (
32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c
^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*
C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^
4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^
3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b
^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c
^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*
b^2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*
f^2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4
)*(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^
3*f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C
^2*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2
*f^4)))^(1/2) + (64*(C^5*a^4*b^3*d^18 + 4*C^5*a^7*c^3*d^15 + 2*C^5*a^7*c^5*
d^13 - C^5*a^6*b*d^18 + 2*C^5*a^7*c*d^17 + C^5*a^2*b^5*c^2*d^16 + 4*C^5*a^2
*b^5*c^4*d^14 + 6*C^5*a^2*b^5*c^6*d^12 + 4*C^5*a^2*b^5*c^8*d^10 + C^5*a^2*b
^5*c^10*d^8 - 8*C^5*a^3*b^4*c^3*d^15 - 12*C^5*a^3*b^4*c^5*d^13 - 8*C^5*a^3*
b^4*c^7*d^11 - 2*C^5*a^3*b^4*c^9*d^9 + 3*C^5*a^4*b^3*c^2*d^16 + C^5*a^4*b^3
*c^4*d^14 - 3*C^5*a^4*b^3*c^6*d^12 - 2*C^5*a^4*b^3*c^8*d^10 + 12*C^5*a^5*b^
2*c^3*d^15 + 18*C^5*a^5*b^2*c^5*d^13 + 8*C^5*a^5*b^2*c^7*d^11 - 2*C^5*a^3*b
^4*c*d^17 + 2*C^5*a^5*b^2*c*d^17 - 9*C^5*a^6*b*c^2*d^16 - 15*C^5*a^6*b*c^4*
d^14 - 7*C^5*a^6*b*c^6*d^12))/(b^3*f^5)))*(-(((8*C^2*a^2*c^3*f^2 - 8*C^2*b^
2*c^3*f^2 - 16*C^2*a*b*d^3*f^2 - 24*C^2*a^2*c*d^2*f^2 + 24*C^2*b^2*c*d^2*f^
2 + 48*C^2*a*b*c^2*d*f^2)^2/4 - (16*a^4*f^4 + 16*b^4*f^4 + 32*a^2*b^2*f^4)*
(C^4*c^6 + C^4*d^6 + 3*C^4*c^2*d^4 + 3*C^4*c^4*d^2))^(1/2) + 4*C^2*a^2*c^3*
f^2 - 4*C^2*b^2*c^3*f^2 - 8*C^2*a*b*d^3*f^2 - 12*C^2*a^2*c*d^2*f^2 + 12*C^2
*b^2*c*d^2*f^2 + 24*C^2*a*b*c^2*d*f^2)/(16*(a^4*f^4 + b^4*f^4 + 2*a^2*b^2*f
^4)))^(1/2)*2i + (2*C*(c + d*tan(e + f*x))^(3/2))/(3*b*f) - (atan((((-(b^7*
f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a
^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(B
^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4
*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d
^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d
^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d
^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6
*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))
/(b*f^4) - ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^
2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(15*B^3*a^
```

```
4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*
c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*
b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3
*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^
3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 1
6*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*
d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B
^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d
^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^
3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5) + ((-(b^7*f^
2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4
*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*B
^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10
*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^
2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B
^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2
 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^
3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*
B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d
^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*
a^6*b^2*c^3*d^10*f^2))/(b*f^4) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)
*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))
^(1/2)*((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4
 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*
B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4
*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^
4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^
10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (32*
(-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 -
3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*
(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*
d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8
*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 2
4*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b^4*f^6*(a^2 + b^2)^2)))/(b^3*
f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2))*1i)
/(b^3*f^2*(a^2 + b^2)^2) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*
a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)
*((32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^
6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 +
 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2
*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3
*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^
4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*
b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 +
 a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^
```

3*b^2*c^2*d))^(1/2)*((32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5) - ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4) - ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) + (32*(-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b^4*f^6*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2))*1i)/(b^3*f^2*(a^2 + b^2)^2))/((64*(B^5*a^3*b^2*d^18 - B^5*a^5*d^18 - B^5*a^5*c^2*d^16 + B^5*a^5*c^4*d^14 + B^5*a^5*c^6*d^12 - 8*B^5*a^2*b^3*c^3*d^15 - 14*B^5*a^2*b^3*c^5*d^13 - 12*B^5*a^2*b^3*c^7*d^11 - 4*B^5*a^2*b^3*c^9*d^9 + 3*B^5*a^3*b^2*c^2*d^16 + 9*B^5*a^3*b^2*c^4*d^14 + 13*B^5*a^3*b^2*c^6*d^12 + 6*B^5*a^3*b^2*c^8*d^10 + 2*B^5*a^4*b*c*d^17 + B^5*a*b^4*c^2*d^16 + 4*B^5*a*b^4*c^4*d^14 + 6*B^5*a*b^4*c^6*d^12 + 4*B^5*a*b^4*c^8*d^10 + B^5*a*b^4*c^10*d^8 - 2*B^5*a^2*b^3*c*d^17 - 6*B^5*a^4*b*c^5*d^13 - 4*B^5*a^4*b*c^7*d^11))/(b*f^5) - ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a

^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6
*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^
4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^
3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c
^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c
*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b*c^5*d^11))/(b*f^4) - ((-(b^7*f^
2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4
*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(15*B^3*a^4*b^3*d^15*f^2 - B^3*
a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 + 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*
b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 - 12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^
7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*
b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2 - 57*B^3*a^3*b^4*c*d^14*f^2 + 64
*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d^13*f^2 + 16*B^3*a^6*b*c^4*d^11*
f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2
*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*f^2 + 129*B^3*a^3*b^4*c^5*d^10*f
^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^4*b^3*c^2*d^13*f^2 - 119*B^3*a^4
*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^
2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a
^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*
b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 +
2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^10*f^2 - 10*B^2*b^8*c^5*d^8*f^2
- 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22
*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2
- 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d
^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a
^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^11*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f
^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b
^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^2 - 48*B^2*a^6*b^2*c^3*d^10*f^2)
)/(b*f^4) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a
^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(4*B*a*b^
8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f
^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 -
12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4
- 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d
^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*
c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*f^5) - (32*(-(b^7*f^2 + 2*a^2*b^
5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3
*B^2*a^3*b^2*c^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^10*d^10*f^4 + 1
6*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 - 16*a^6*b^4*d^10*f^4 + 24*b^10*c^
2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*
d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 +
8*a^7*b^3*c*d^9*f^4))/(b^4*f^6*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/
(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)
+ ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^
3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x

```
))^(1/2)*(B^4*b^6*d^16 - 2*B^4*a^6*d^16 + 12*B^4*a^6*c^2*d^14 - 2*B^4*a^6*c
^4*d^12 + 4*B^4*b^6*c^2*d^14 + 6*B^4*b^6*c^4*d^12 + 4*B^4*b^6*c^6*d^10 + B^
4*b^6*c^8*d^8 - 2*B^4*a^2*b^4*c^4*d^12 + 12*B^4*a^2*b^4*c^6*d^10 - 2*B^4*a^
2*b^4*c^8*d^8 + 8*B^4*a^3*b^3*c^3*d^13 - 48*B^4*a^3*b^3*c^5*d^11 + 8*B^4*a^
3*b^3*c^7*d^9 - 12*B^4*a^4*b^2*c^2*d^14 + 72*B^4*a^4*b^2*c^4*d^12 - 12*B^4*
a^4*b^2*c^6*d^10 + 8*B^4*a^5*b*c*d^15 - 48*B^4*a^5*b*c^3*d^13 + 8*B^4*a^5*b
*c^5*d^11))/(b*f^4) + ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d
^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32
*(15*B^3*a^4*b^3*d^15*f^2 - B^3*a^2*b^5*d^15*f^2 - 4*B^3*a^7*c^3*d^12*f^2 +
 2*B^3*b^7*c^2*d^13*f^2 + 4*B^3*b^7*c^4*d^11*f^2 + 2*B^3*b^7*c^6*d^9*f^2 -
12*B^3*a^6*b*d^15*f^2 - 4*B^3*a^7*c*d^14*f^2 - B^3*a*b^6*c*d^14*f^2 - 27*B^
3*a*b^6*c^3*d^12*f^2 - 19*B^3*a*b^6*c^5*d^10*f^2 + 7*B^3*a*b^6*c^7*d^8*f^2
- 57*B^3*a^3*b^4*c*d^14*f^2 + 64*B^3*a^5*b^2*c*d^14*f^2 + 4*B^3*a^6*b*c^2*d
^13*f^2 + 16*B^3*a^6*b*c^4*d^11*f^2 + 65*B^3*a^2*b^5*c^2*d^13*f^2 + 9*B^3*a
^2*b^5*c^4*d^11*f^2 - 57*B^3*a^2*b^5*c^6*d^9*f^2 + 77*B^3*a^3*b^4*c^3*d^12*
f^2 + 129*B^3*a^3*b^4*c^5*d^10*f^2 - 5*B^3*a^3*b^4*c^7*d^8*f^2 - 121*B^3*a^
4*b^3*c^2*d^13*f^2 - 119*B^3*a^4*b^3*c^4*d^11*f^2 + 17*B^3*a^4*b^3*c^6*d^9*
f^2 + 40*B^3*a^5*b^2*c^3*d^12*f^2 - 24*B^3*a^5*b^2*c^5*d^10*f^2))/(b*f^5) -
 ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3
- 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))
^(1/2)*(4*B^2*a^3*b^5*d^13*f^2 + 2*B^2*a^5*b^3*d^13*f^2 + 28*B^2*b^8*c^3*d^
10*f^2 - 10*B^2*b^8*c^5*d^8*f^2 - 14*B^2*a*b^7*d^13*f^2 + 16*B^2*a^7*b*d^13
*f^2 - 8*B^2*a^8*c*d^12*f^2 + 22*B^2*b^8*c*d^12*f^2 + 20*B^2*a*b^7*c^2*d^11
*f^2 + 50*B^2*a*b^7*c^4*d^9*f^2 - 28*B^2*a^2*b^6*c*d^12*f^2 - 2*B^2*a^4*b^4
*c*d^12*f^2 - 56*B^2*a^6*b^2*c*d^12*f^2 + 32*B^2*a^7*b*c^2*d^11*f^2 + 8*B^2
*a^2*b^6*c^3*d^10*f^2 + 12*B^2*a^2*b^6*c^5*d^8*f^2 - 24*B^2*a^3*b^5*c^2*d^1
1*f^2 - 12*B^2*a^3*b^5*c^4*d^9*f^2 - 4*B^2*a^4*b^4*c^3*d^10*f^2 - 10*B^2*a^
4*b^4*c^5*d^8*f^2 + 52*B^2*a^5*b^3*c^2*d^11*f^2 + 34*B^2*a^5*b^3*c^4*d^9*f^
2 - 48*B^2*a^6*b^2*c^3*d^10*f^2))/(b*f^4) - ((-(b^7*f^2 + 2*a^2*b^5*f^2 + a
^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*
b^2*c^2*d))^(1/2)*((32*(4*B*a*b^8*d^12*f^4 - 4*B*b^9*c*d^11*f^4 + 8*B*a^3*b
^6*d^12*f^4 + 4*B*a^5*b^4*d^12*f^4 - 4*B*b^9*c^3*d^9*f^4 + 8*B*a*b^8*c^2*d^
10*f^4 + 4*B*a*b^8*c^4*d^8*f^4 - 12*B*a^2*b^7*c*d^11*f^4 - 12*B*a^4*b^5*c*d
^11*f^4 - 4*B*a^6*b^3*c*d^11*f^4 - 12*B*a^2*b^7*c^3*d^9*f^4 + 16*B*a^3*b^6*
c^2*d^10*f^4 + 8*B*a^3*b^6*c^4*d^8*f^4 - 12*B*a^4*b^5*c^3*d^9*f^4 + 8*B*a^5
*b^4*c^2*d^10*f^4 + 4*B*a^5*b^4*c^4*d^8*f^4 - 4*B*a^6*b^3*c^3*d^9*f^4))/(b*
f^5) + (32*(-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2)*(B^2*a^5*d^3 - B^2*a^2
*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d))^(1/2)*(c + d*tan(e + f
*x))^(1/2)*(16*b^10*d^10*f^4 + 16*a^2*b^8*d^10*f^4 - 16*a^4*b^6*d^10*f^4 -
16*a^6*b^4*d^10*f^4 + 24*b^10*c^2*d^8*f^4 + 40*a^2*b^8*c^2*d^8*f^4 + 8*a^4*
b^6*c^2*d^8*f^4 - 8*a^6*b^4*c^2*d^8*f^4 + 8*a*b^9*c*d^9*f^4 + 24*a^3*b^7*c*
d^9*f^4 + 24*a^5*b^5*c*d^9*f^4 + 8*a^7*b^3*c*d^9*f^4))/(b^4*f^6*(a^2 + b^2)
^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))/(b^3*f^2*(a^2 + b
^2)^2)))/(b^3*f^2*(a^2 + b^2)^2)))*(-(b^7*f^2 + 2*a^2*b^5*f^2 + a^4*b^3*f^2
)*(B^2*a^5*d^3 - B^2*a^2*b^3*c^3 - 3*B^2*a^4*b*c*d^2 + 3*B^2*a^3*b^2*c^2*d)
```

)^(1/2)*2i)/(b^3*f^2*(a^2 + b^2)^2) + (atan((((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4 + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5 + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4 + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 + (32*(-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*1i)/(b*f^2*(a^2 + b^2)^2) + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*

```
a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a
*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c
^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4 - ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*
f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(
1/2)*((32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c
^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15
*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^
11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*
b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 8
1*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d
^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^
4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5 - ((-(b^5*f^2 + a^4*b
*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^
2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f
^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8
*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f
^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*
c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A
^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d
^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a
^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4 - ((-(b^5*f^2 + a^4*b
*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^
2*a^2*b*c*d^2))^(1/2)*((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4
*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a
*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A
*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4
 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*
d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 - (32*(-(b
^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^
2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10
*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24
*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b
^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^
9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)
))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2))*1i)/(b*f^2*(a^2 + b^2)^2)
)/((64*(A^5*a^2*b^2*d^18 + A^5*b^4*c^2*d^16 + 5*A^5*b^4*c^4*d^14 + 7*A^5*b^
4*c^6*d^12 + 3*A^5*b^4*c^8*d^10 + 9*A^5*a^2*b^2*c^2*d^16 + 15*A^5*a^2*b^2*c
^4*d^14 + 7*A^5*a^2*b^2*c^6*d^12 - 2*A^5*a*b^3*c*d^17 - 2*A^5*a^3*b*c*d^17
- 12*A^5*a*b^3*c^3*d^15 - 18*A^5*a*b^3*c^5*d^13 - 8*A^5*a*b^3*c^7*d^11 - 4*
A^5*a^3*b*c^3*d^15 - 2*A^5*a^3*b*c^5*d^13))/f^5 + ((-(b^5*f^2 + a^4*b*f^2 +
 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*
b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c
^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A
^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*
a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4
```

*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4 + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5 + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4 + ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 + (32*(-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2) - ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(A^4*b^5*d^16 + 4*A^4*b^5*c^2*d^14 + 8*A^4*b^5*c^4*d^12 - 8*A^4*b^5*c^6*d^10 + 3*A^4*b^5*c^8*d^8 + 2*A^4*a^4*b*d^16 + 12*A^4*a^2*b^3*c^2*d^14 - 72*A^4*a^2*b^3*c^4*d^12 + 12*A^4*a^2*b^3*c^6*d^10 + 48*A^4*a^3*b^2*c^3*d^13 - 8*A^4*a^3*b^2*c^5*d^11 - 8*A^4*a*b^4*c^3*d^13 + 48*A^4*a*b^4*c^5*d^11 - 8*A^4*a*b^4*c^7*d^9 - 8*A^4*a^3*b^2*c*d^15 - 12*A^4*a^4*b*c^2*d^14 + 2*A^4*a^4*b*c^4*d^12))/f^4 - ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(23*A^3*b^6*c^3*d^12*f^2 - 15*A^3*a^3*b^3*d^15*f^2 + 21*A^3*b^6*c^5*d^10*f^2 - 3*A^3*b^6*c^7*d^8*f^2 + A^3*a*b^5*d^15*f^2 + 4*A^3*a^5*b*d^

15*f^2 - A^3*b^6*c*d^14*f^2 - 61*A^3*a*b^5*c^2*d^13*f^2 - 25*A^3*a*b^5*c^4*d^11*f^2 + 37*A^3*a*b^5*c^6*d^9*f^2 + 53*A^3*a^2*b^4*c*d^14*f^2 - 30*A^3*a^4*b^2*c*d^14*f^2 + 4*A^3*a^5*b*c^2*d^13*f^2 - 29*A^3*a^2*b^4*c^3*d^12*f^2 - 81*A^3*a^2*b^4*c^5*d^10*f^2 + A^3*a^2*b^4*c^7*d^8*f^2 + 59*A^3*a^3*b^3*c^2*d^13*f^2 + 75*A^3*a^3*b^3*c^4*d^11*f^2 + A^3*a^3*b^3*c^6*d^9*f^2 - 32*A^3*a^4*b^2*c^3*d^12*f^2 - 2*A^3*a^4*b^2*c^5*d^10*f^2))/f^5 - ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^13*f^2 - 14*A^2*a^5*b^2*d^13*f^2 + 28*A^2*b^7*c^3*d^10*f^2 - 18*A^2*b^7*c^5*d^8*f^2 - 14*A^2*a*b^6*d^13*f^2 + 22*A^2*b^7*c*d^12*f^2 + 8*A^2*a^6*b*c*d^12*f^2 + 20*A^2*a*b^6*c^2*d^11*f^2 + 66*A^2*a*b^6*c^4*d^9*f^2 - 28*A^2*a^2*b^5*c*d^12*f^2 + 54*A^2*a^4*b^3*c*d^12*f^2 + 24*A^2*a^2*b^5*c^3*d^10*f^2 + 12*A^2*a^2*b^5*c^5*d^8*f^2 - 88*A^2*a^3*b^4*c^2*d^11*f^2 - 28*A^2*a^3*b^4*c^4*d^9*f^2 + 60*A^2*a^4*b^3*c^3*d^10*f^2 - 2*A^2*a^4*b^3*c^5*d^8*f^2 - 44*A^2*a^5*b^2*c^2*d^11*f^2 + 2*A^2*a^5*b^2*c^4*d^9*f^2))/f^4 - ((-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*((32*(4*A*a^2*b^6*d^12*f^4 + 8*A*a^4*b^4*d^12*f^4 + 4*A*a^6*b^2*d^12*f^4 + 12*A*b^8*c^2*d^10*f^4 + 12*A*b^8*c^4*d^8*f^4 - 16*A*a*b^7*c^3*d^9*f^4 - 32*A*a^3*b^5*c*d^11*f^4 - 16*A*a^5*b^3*c*d^11*f^4 + 28*A*a^2*b^6*c^2*d^10*f^4 + 24*A*a^2*b^6*c^4*d^8*f^4 - 32*A*a^3*b^5*c^3*d^9*f^4 + 20*A*a^4*b^4*c^2*d^10*f^4 + 12*A*a^4*b^4*c^4*d^8*f^4 - 16*A*a^5*b^3*c^3*d^9*f^4 + 4*A*a^6*b^2*c^2*d^10*f^4 - 16*A*a*b^7*c*d^11*f^4))/f^5 - (32*(-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(b*f^6*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))/(b*f^2*(a^2 + b^2)^2)))*(-(b^5*f^2 + a^4*b*f^2 + 2*a^2*b^3*f^2)*(A^2*a^3*d^3 - A^2*b^3*c^3 + 3*A^2*a*b^2*c^2*d - 3*A^2*a^2*b*c*d^2))^(1/2)*2i)/(b*f^2*(a^2 + b^2)^2) + (2*B*d*(c + d*tan(e + f*x))^(1/2))/(b*f) + (atan((((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2

+ 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*
f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3
*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*
C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d
^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*
a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^
8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^
3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^1
0*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^
5) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*
c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f
x))^(1/2)(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c
^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*
b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*
c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2
*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2
+ 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*
c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10
*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4
*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2
*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^1
0*f^2))/(b^3*f^4) - ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3
- C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(
12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C
*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*
a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 +
8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*
f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2
*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) - (32*(-(b^9*f^2 + 2*a^2*b^7
*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*
C^2*a^5*b^2*c^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^12*d^10*f^4 + 16
*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^
2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2
*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4
+ 8*a^7*b^5*c*d^9*f^4))/(b^8*f^6*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2))
)/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2))*1i)/(b^5*f^2*(a^2 + b^
2)^2) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b
^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e
+ f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*
a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10
+ C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C
^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C
^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12
*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*
a^7*b*c^5*d^11))/(b^3*f^4) - ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^

2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5) - ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) + (32*(-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^8*f^6*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2))*1i)/(b^5*f^2*(a^2 + b^2)^2))/((64*(C^5*a^4*b^3*d^18 + 4*C^5*a^7*c^3*d^15 + 2*C^5*a^7*c^5*d^13 - C^5*a^6*b*d^18 + 2*C^5*a^7*c*d^17 + C^5*a^2*b^5*c^2*d^16 + 4*C^5*a^2*b^5*c^4*d^14 + 6*C^5*a^2*b^5*c^6*d^12 + 4*C^5*a^2*b^5*c^8*d^10 + C^5*a^2*b^5*c^10*d^8 - 8*C^5*a^3*b^4*c^3*d^15 - 12*C^5*a^3*b^4*c^5*d^13 - 8*C^5*a^3*b^4*c^7*d^11 - 2*C^5*a^3*b^4*c^9*d^9 + 3*C^5* a^4*b^3*c^2*d^16 + C^5*a^4*b^3*c^4*d^14 - 3*C^5*a^4*b^3*c^6*d^12 - 2*C^5*a^4*b^3*c^8*d^10 + 12*C^5*a^5*b^2*c^3*d^15 + 18*C^5*a^5*b^2*c^5*d^13 + 8*C^5*a^5*b^2*c^7*d^11 - 2*C^5*a^3*b^4*c*d^17 + 2*C^5*a^5*b^2*c*d^17 - 9*C^5*a^6*b*c^2*d^16 - 15*C^5*a^6*b*c^4*d^14 - 7*C^5*a^6*b*c^6*d^12))/(b^3*f^5) - ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4) - ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(12*C*a^2*b^9*d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^10*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^11*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3 *d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 16*C*a*b^10*c*d^11*f^4))/(b^3*f^5) - (32*(-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^10*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 40*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*c*d^9*f^4))/(b^8*f^6*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^8*d^16 + C^4*b^8*d^16 - 12*C^4*a^8*c^2*d^14 + 2*C^4*a^8*c^4*d^12 + 4*C^4*b^8*c^2*d^14 + 6*C^4*b^8*c^4*d^12 + 4*C^4*b^8*c^6*d^10 + C^4*b^8*c^8*d^8 + 2*C^4*a^4*b^4*c^4*d^12 - 12*C^4*a^4*b^4*c^6*d^10 + 2*C^4*a^4*b^4*c^8*d^8 - 8*C^4*a^5*b^3*c^3*d^13 + 48*C^4*a^5*b^3*c^5*d^11 - 8*C^4*a^5*b^3*c^7*d^9 + 12*C^4*a^6*b^2*c^2*d^14 - 72*C^4*a^6*b^2*c^4*d^12 + 12*C^4*a^6*b^2*c^6*d^10 - 8*C^4*a^7*b*c*d^15 + 48*C^4*a^7*b*c^3*d^13 - 8*C^4*a^7*b*c^5*d^11))/(b^3*f^4) - ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(4*C^3*a^9*d^15*f^2 + C^3*a^3*b^6*d^15*f^2 + 16*C^3*a^5*b^4*d^15*f^2 - 16*C^3*a^7*b^2*d^15*f^2 + 4*C^3*a^9*c^2*d^13*f^2 - C^3*b^9*c^3*d^12*f^2 + C^3*b^9*c^5*d^10*f^2 + C^3*b^9*c^7*d^8*f^2 + C^3*a*b^8*d^15*f^2 - C^3*b^9*c*d^14*f^2 - 28*C^3*a^8*b*c*d^14*f^2 + 3*C^3*a*b^8*c^2*d^13*f^2 + 3*C^3*a*b^8*c^4*d^11*f^2 + C^3*a*b^8*c^6*d^9*f^2 - 3*C^3*a^2*b^7*c*d^14*f^2 - 58*C^3*a^4*b^5*c*d^14*f^2 + 80*C^3*a^6*b^3*c*d^14*f^2 - 28*C^3*a^8*b*c^3*d^12*f^2 - 29*C^3*a^2*b^7*c^3*d^12*f^2 - 17*C^3*a^2*b^7*c^5*d^10*f^2 + 9*C^3*a^2*b^7*c^7*d^8*f^2 + 67*C^3*a^3*b^6*c^2*d^13*f^2 + 3*C^3*a^3*b^6*c^4*d^11*f^2 - 63*C^3*a^3*b^6*c^6*d^9*f^2 + 92*C^3*a^4*b^5*c^3*d^12*f^2 + 138*C^3*a^4*b^5*c^5*d^10*f^2 - 12*C^3*a^4*b^5*c^7*d^8*f^2 - 144*C^3*a^5*b^4*c^2*d^13*f^2 - 108*C^3*a^5*b^4*c^4*d^11*f^2 + 52*C^3*a^5*b^4*c^6*d^9*f^2 - 8*C^3*a^6*b^3*c^3*d^12*f^2 - 88*C^3*a^6*b^3*c^5*d^10*f^2 + 56*C^3*a^7*b^2*c^2*d^13*f^2 + 72*C^3*a^7*b^2*c^4*d^11*f^2))/(b^3*f^5) - ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(c + d*tan(e + f*x))^(1/2)*(4*C^2*a^3*b^7*d^13*f^2 + 2*C^2*a^5*b^5*d^13*f^2 + 28*C^2*b^10*c^3*d^10*f^2 - 10*C^2*b^10*c^5*d^8*f^2 - 14*C^2*a*b^9*d^13*f^2 - 16*C^2*a^9*b*d^13*f^2 + 8*C^2*a^10*c*d^12*f^2 + 22*C^2*b^10*c*d^12*f^2 + 20*C^2*a*b^9*c^2*d^11*f^2 + 50*C^2*a*b^9*c^4*d^9*f^2 - 28*C^2*a^2*b^8*c*d^12*f^2 - 2*C^2*a^4*b^6*c*d^12*f^2 + 56*C^2*a^8*b^2*c*d^12*f^2 - 32*C^2*a^9*b*c^2*d^11*f^2 + 8*C^2*a^2*b^8*c^3*d^10*f^2 + 4*C^2*a^2*b^8*c^5*d^8*f^2 - 24*C^2*a^3*b^7*c^2*d^11*f^2 + 4*C^2*a^3*b^7*c^4*d^9*f^2 + 12*C^2*a^4*b^6*c^3*d^10*f^2 - 10*C^2*a^4*b^6*c^5*d^8*f^2 - 12*C^2*a^5*b^5*c^2*d^11*f^2 + 18*C^2*a^5*b^5*c^4*d^9*f^2 + 16*C^2*a^6*b^4*c^3*d^10*f^2 + 8*C^2*a^6*b^4*c^5*d^8*f^2 - 64*C^2*a^7*b^3*c^2*d^11*f^2 - 32*C^2*a^7*b^3*c^4*d^9*f^2 + 48*C^2*a^8*b^2*c^3*d^10*f^2))/(b^3*f^4) + ((-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*((32*(12*C*a^2*b^9*

```
d^12*f^4 + 24*C*a^4*b^7*d^12*f^4 + 12*C*a^6*b^5*d^12*f^4 + 4*C*b^11*c^2*d^1
0*f^4 + 4*C*b^11*c^4*d^8*f^4 - 16*C*a*b^10*c^3*d^9*f^4 - 32*C*a^3*b^8*c*d^1
1*f^4 - 16*C*a^5*b^6*c*d^11*f^4 + 20*C*a^2*b^9*c^2*d^10*f^4 + 8*C*a^2*b^9*c
^4*d^8*f^4 - 32*C*a^3*b^8*c^3*d^9*f^4 + 28*C*a^4*b^7*c^2*d^10*f^4 + 4*C*a^4
*b^7*c^4*d^8*f^4 - 16*C*a^5*b^6*c^3*d^9*f^4 + 12*C*a^6*b^5*c^2*d^10*f^4 - 1
6*C*a*b^10*c*d^11*f^4))/(b^3*f^5) + (32*(-(b^9*f^2 + 2*a^2*b^7*f^2 + a^4*b^
5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^6*b*c*d^2 + 3*C^2*a^5*b^2*c
^2*d))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(16*b^12*d^10*f^4 + 16*a^2*b^10*d^1
0*f^4 - 16*a^4*b^8*d^10*f^4 - 16*a^6*b^6*d^10*f^4 + 24*b^12*c^2*d^8*f^4 + 4
0*a^2*b^10*c^2*d^8*f^4 + 8*a^4*b^8*c^2*d^8*f^4 - 8*a^6*b^6*c^2*d^8*f^4 + 8*
a*b^11*c*d^9*f^4 + 24*a^3*b^9*c*d^9*f^4 + 24*a^5*b^7*c*d^9*f^4 + 8*a^7*b^5*
c*d^9*f^4))/(b^8*f^6*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a
^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))/(b^5*f^2*(a^2 + b^2)^2)))*(-(b^9*f
^2 + 2*a^2*b^7*f^2 + a^4*b^5*f^2)*(C^2*a^7*d^3 - C^2*a^4*b^3*c^3 - 3*C^2*a^
6*b*c*d^2 + 3*C^2*a^5*b^2*c^2*d))^(1/2)*2i)/(b^5*f^2*(a^2 + b^2)^2)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(e+fx\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{a+b\tan\left(e+fx\right)}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e)),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x)), x)
```

3.102 $\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$

Optimal. Leaf size=372

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{bf\left(a^2+b^2\right)(a+b\tan(e+fx))}+\frac{d\left(3a^2C-abB+Ab^2+2b^2C\right)\sqrt{c+d\tan(e+fx)}}{b^2f\left(a^2+b^2\right)}+\frac{\sqrt{bc-ad}\,\left(-3a\right.}{}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a
-I*b)^2/f-(B-I*(A-C))*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^
(1/2))/(a+I*b)^2/f+(a^3*b*B*d-3*a^4*C*d-b^4*(3*A*d+2*B*c)-a*b^3*(4*A*c-5*B*
d-4*C*c)+a^2*b^2*(2*B*c+(A-7*C)*d))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/
(-a*d+b*c)^(1/2))*(-a*d+b*c)^(1/2)/b^(5/2)/(a^2+b^2)^2/f+(A*b^2-B*a*b+3*C*a
^2+2*C*b^2)*d*(c+d*tan(f*x+e))^(1/2)/b^2/(a^2+b^2)/f-(A*b^2-a*(B*b-C*a))*(c
+d*tan(f*x+e))^(3/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] time = 2.55, antiderivative size = 372, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3645, 3647, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{bf\left(a^2+b^2\right)(a+b\tan(e+fx))}+\frac{d\left(3a^2C-abB+Ab^2+2b^2C\right)\sqrt{c+d\tan(e+fx)}}{b^2f\left(a^2+b^2\right)}+\frac{\sqrt{bc-ad}\,\left(a^2b\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^2,x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/((a - I*b)^2*f)) - ((B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[Sqrt[c
 + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((a + I*b)^2*f) + (Sqrt[b*c - a*d]*(a^3*
b*B*d - 3*a^4*C*d - b^4*(2*B*c + 3*A*d) - a*b^3*(4*A*c - 4*c*C - 5*B*d) + a
^2*b^2*(2*B*c + (A - 7*C)*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sq
rt[b*c - a*d]])/(b^(5/2)*(a^2 + b^2)^2*f) + ((A*b^2 - a*b*B + 3*a^2*C + 2*b
^2*C)*d*Sqrt[c + d*Tan[e + f*x]])/(b^2*(a^2 + b^2)*f) - ((A*b^2 - a*(b*B -
a*C))*(c + d*Tan[e + f*x])^(3/2))/(b*(a^2 + b^2)*f*(a + b*Tan[e + f*x]))
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
```

```
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}$$

$$= \frac{\left(Ab^2-abB+3a^2C+2b^2C\right)d\sqrt{c+d\tan(e}}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{\left(Ab^2-abB+3a^2C+2b^2C\right)d\sqrt{c+d\tan(e}}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{\left(Ab^2-abB+3a^2C+2b^2C\right)d\sqrt{c+d\tan(e}}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{\left(Ab^2-abB+3a^2C+2b^2C\right)d\sqrt{c+d\tan(e}}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{\sqrt{bc-ad}\left(a^3bBd-3a^4Cd-b^4(2Bc+3Ad)\right.}{}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+}}{\sqrt{c-id}}\right.}{(a-ib)^2f}$$

Mathematica [B] time = 6.31, size = 1732, normalized size = 4.66

result too large to display

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^2,x]
```

```
[Out] (-4*a^2*A*b^3*c*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/
Sqrt[b*c - a*d]] + 2*a^3*b^2*B*c*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c +
d*Tan[e + f*x]])/Sqrt[b*c - a*d]] - 2*a*b^4*B*c*Sqrt[b*c - a*d]*ArcTanh[(Sq
rt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]] + 4*a^2*b^3*c*C*Sqrt[b*c -
```

```
 a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]] + a^3*A*b
^2*d*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c -
a*d]] - 3*a*A*b^4*d*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x
]])/Sqrt[b*c - a*d]] + a^4*b*B*d*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c +
d*Tan[e + f*x]])/Sqrt[b*c - a*d]] + 5*a^2*b^3*B*d*Sqrt[b*c - a*d]*ArcTanh[(
Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]] - 3*a^5*C*d*Sqrt[b*c - a
*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]] - 7*a^3*b^2
*C*d*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c -
a*d]] - 4*a*A*b^4*c*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x
]])/Sqrt[b*c - a*d]]*Tan[e + f*x] + 2*a^2*b^3*B*c*Sqrt[b*c - a*d]*ArcTanh[(
Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]]*Tan[e + f*x] - 2*b^5*B*c
*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]
]*Tan[e + f*x] + 4*a*b^4*c*C*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Ta
n[e + f*x]])/Sqrt[b*c - a*d]]*Tan[e + f*x] + a^2*A*b^3*d*Sqrt[b*c - a*d]*Ar
cTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]]*Tan[e + f*x] - 3*
A*b^5*d*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c
 - a*d]]*Tan[e + f*x] + a^3*b^2*B*d*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c
 + d*Tan[e + f*x]])/Sqrt[b*c - a*d]]*Tan[e + f*x] + 5*a*b^4*B*d*Sqrt[b*c -
a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]]*Tan[e + f*
x] - 3*a^4*b*C*d*Sqrt[b*c - a*d]*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])
/Sqrt[b*c - a*d]]*Tan[e + f*x] - 7*a^2*b^3*C*d*Sqrt[b*c - a*d]*ArcTanh[(Sqr
t[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]]*Tan[e + f*x] + b^(5/2)*((-I
)*a + b)^2*(I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c - I*d]]*(a + b*Tan[e + f*x]) + b^(5/2)*(I*a + b)^2*((-I)*A + B + I*
C)*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]*(a + b*T
an[e + f*x]) - a^2*A*b^(7/2)*c*Sqrt[c + d*Tan[e + f*x]] - A*b^(11/2)*c*Sqrt
[c + d*Tan[e + f*x]] + a^3*b^(5/2)*B*c*Sqrt[c + d*Tan[e + f*x]] + a*b^(9/2)
*B*c*Sqrt[c + d*Tan[e + f*x]] - a^4*b^(3/2)*c*C*Sqrt[c + d*Tan[e + f*x]] -
a^2*b^(7/2)*c*C*Sqrt[c + d*Tan[e + f*x]] + a^3*A*b^(5/2)*d*Sqrt[c + d*Tan[e
 + f*x]] + a*A*b^(9/2)*d*Sqrt[c + d*Tan[e + f*x]] - a^4*b^(3/2)*B*d*Sqrt[c
+ d*Tan[e + f*x]] - a^2*b^(7/2)*B*d*Sqrt[c + d*Tan[e + f*x]] + 3*a^5*Sqrt[b
]*C*d*Sqrt[c + d*Tan[e + f*x]] + 5*a^3*b^(5/2)*C*d*Sqrt[c + d*Tan[e + f*x]]
 + 2*a*b^(9/2)*C*d*Sqrt[c + d*Tan[e + f*x]] + 2*a^4*b^(3/2)*C*d*Tan[e + f*x
]*Sqrt[c + d*Tan[e + f*x]] + 4*a^2*b^(7/2)*C*d*Tan[e + f*x]*Sqrt[c + d*Tan[
e + f*x]] + 2*b^(11/2)*C*d*Tan[e + f*x]*Sqrt[c + d*Tan[e + f*x]])/(b^(5/2)*
(a^2 + b^2)^2*f*(a + b*Tan[e + f*x]))
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.83, size = 9865, normalized size = 26.52

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^2,x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(e+fx\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\left(a+b\tan\left(e+fx\right)\right)^{2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**2,x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**2, x)
```

3.103 $\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$

Optimal. Leaf size=532

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{2bf\left(a^2+b^2\right)(a+b\tan(e+fx))^2}-\frac{\sqrt{c+d\tan(e+fx)}\left(3a^4Cd+a^3bBd-a^2b^2(5Ad+4Bc-11Cd)+a\right.}{4b^2f\left(a^2+b^2\right)^2(a+b\tan(e+f}$$

```
[Out] -(A-I*B-C)*(c-I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(I*a
+b)^3/f+(A+I*B-C)*(c+I*d)^(3/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/(I*a-b)^3/f-1/4*(a^5*b*B*d^2+3*a^6*C*d^2+a^4*b^2*d*(4*B*c+3*(A+2*C)*d)-b
^6*(8*A*c^2-3*A*d^2-12*B*c*d-8*C*c^2)+a^2*b^4*(24*A*c^2-26*A*d^2-48*B*c*d-2
4*C*c^2+35*C*d^2)-2*a^3*b^3*(12*c*(A-C)*d+B*(4*c^2-9*d^2))+a*b^5*(40*c*(A-C
)*d+3*B*(8*c^2-5*d^2)))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-a*d+b*c)^(
1/2))/b^(5/2)/(a^2+b^2)^3/f/(-a*d+b*c)^(1/2)-1/4*(a^3*b*B*d+3*a^4*C*d+b^4*(
3*A*d+4*B*c)+a*b^3*(8*A*c-7*B*d-8*C*c)-a^2*b^2*(5*A*d+4*B*c-11*C*d))*(c+d*t
an(f*x+e))^(1/2)/b^2/(a^2+b^2)^2/f/(a+b*tan(f*x+e))-1/2*(A*b^2-a*(B*b-C*a))
*(c+d*tan(f*x+e))^(3/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^2
```

Rubi [A] time = 4.09, antiderivative size = 532, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3645, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{\left(-2a^3b^3\left(12cd(A-C)+B\left(4c^2-9d^2\right)\right)+a^2b^4\left(24Ac^2-26Ad^2-48Bcd-24c^2C+35Cd^2\right)+a^4b^2d(3d(A+2C\right.}{4b^{5/}}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^3,x]
```

```
[Out] -(((A - I*B - C)*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c -
I*d]])/((I*a + b)^3*f)) + ((A + I*B - C)*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((I*a - b)^3*f) - ((a^5*b*B*d^2 + 3*a^6*C*d^
2 + a^4*b^2*d*(4*B*c + 3*(A + 2*C)*d) - b^6*(8*A*c^2 - 8*c^2*C - 12*B*c*d -
 3*A*d^2) + a^2*b^4*(24*A*c^2 - 24*c^2*C - 48*B*c*d - 26*A*d^2 + 35*C*d^2)
- 2*a^3*b^3*(12*c*(A - C)*d + B*(4*c^2 - 9*d^2)) + a*b^5*(40*c*(A - C)*d +
3*B*(8*c^2 - 5*d^2)))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c -
 a*d]])/(4*b^(5/2)*(a^2 + b^2)^3*Sqrt[b*c - a*d]*f) - ((a^3*b*B*d + 3*a^4*C
*d + b^4*(4*B*c + 3*A*d) + a*b^3*(8*A*c - 8*c*C - 7*B*d) - a^2*b^2*(4*B*c +
 5*A*d - 11*C*d))*Sqrt[c + d*Tan[e + f*x]])/(4*b^2*(a^2 + b^2)^2*f*(a + b*T
```

```
an[e + f*x])) - ((A*b^2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^(3/2))/(2*b*(
a^2 + b^2)*f*(a + b*Tan[e + f*x])^2)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
```

```
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}$$

$$= -\frac{\left(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(\right.}{4b^2\left(\right.}$$

$$= -\frac{\left(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(\right.}{4b^2\left(\right.}$$

$$= -\frac{\left(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(\right.}{4b^2\left(\right.}$$

$$= -\frac{\left(a^3bBd+3a^4Cd+b^4(4Bc+3Ad)+ab^3(\right.}{4b^2\left(\right.}$$

$$= -\frac{\left(a^5bBd^2+3a^6Cd^2+a^4b^2d(4Bc+3(A+\right.}{}$$

$$= -\frac{(A-iB-C)(c-id)^{3/2}\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+}}{\sqrt{c-id}}\right.}{(ia+b)^3f}$$

Mathematica [B] time = 6.56, size = 7678, normalized size = 14.43

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^3,x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.83, size = 14441, normalized size = 27.14

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^3,x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(e+fx\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\left(a+b\tan\left(e+fx\right)\right)^{3}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**3,x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**3, x)
```

3.104 $\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+\right.$

Optimal. Leaf size=503

$$-\frac{2\sqrt{c+d\tan(e+fx)}\left(-\left(a^2\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\right)+2ab\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)+b^2\left(2cd(A\right.\right.}{f}$$

```
[Out] -(a-I*b)^2*(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)
^(1/2))/f+(a+I*b)^2*(I*A-B-I*C)*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2
)/(c+I*d)^(1/2))/f-2*(2*a*b*(c^2*C+2*B*c*d-C*d^2-A*(c^2-d^2))-a^2*(2*c*(A-C
)*d+B*(c^2-d^2))+b^2*(2*c*(A-C)*d+B*(c^2-d^2)))*(c+d*tan(f*x+e))^(1/2)/f+2/
3*(2*a*b*(A*c-B*d-C*c)+a^2*(B*c+(A-C)*d)-b^2*(B*c+(A-C)*d))*(c+d*tan(f*x+e)
)^(3/2)/f+2/5*(a^2*B-b^2*B+2*a*b*(A-C))*(c+d*tan(f*x+e))^(5/2)/f+2/693*(36*
a^2*C*d^2-22*a*b*d*(-9*B*d+2*C*c)+b^2*(8*c^2*C-22*B*c*d+99*(A-C)*d^2))*(c+d
*tan(f*x+e))^(7/2)/d^3/f-2/99*b*(-11*B*b*d-4*C*a*d+4*C*b*c)*tan(f*x+e)*(c+d
*tan(f*x+e))^(7/2)/d^2/f+2/11*C*(a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(7/2)/d
/f
```

Rubi [A] time = 2.31, antiderivative size = 503, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}} = 0.170$, Rules used = {3647, 3637, 3630, 3528, 3539, 3537, 63, 208}

$$\frac{2(c+d\tan(e+fx))^{7/2}\left(36a^2Cd^2-22abd(2cC-9Bd)+b^2\left(99d^2(A-C)-22Bcd+8c^2C\right)\right)}{693d^3f}-\frac{2\sqrt{c+d\tan(e+fx}}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^2*(I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c - I*d]])/f) + ((a + I*b)^2*(I*A - B - I*C)*(c + I*d)^(5/2)*ArcT
anh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f - (2*(2*a*b*(c^2*C + 2*B*c*d
 - C*d^2 - A*(c^2 - d^2)) - a^2*(2*c*(A - C)*d + B*(c^2 - d^2)) + b^2*(2*c*
(A - C)*d + B*(c^2 - d^2)))*Sqrt[c + d*Tan[e + f*x]])/f + (2*(2*a*b*(A*c -
c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A - C)*d))*(c + d*Tan[e +
f*x]])^(3/2))/(3*f) + (2*(a^2*B - b^2*B + 2*a*b*(A - C))*(c + d*Tan[e + f*x]
)^(5/2))/(5*f) + (2*(36*a^2*C*d^2 - 22*a*b*d*(2*c*C - 9*B*d) + b^2*(8*c^2*C
 - 22*B*c*d + 99*(A - C)*d^2))*(c + d*Tan[e + f*x]])^(7/2))/(693*d^3*f) - (2
*b*(4*b*c*C - 11*b*B*d - 4*a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x]])^(7/2))/
(99*d^2*f) + (2*C*(a + b*Tan[e + f*x]])^2*(c + d*Tan[e + f*x]])^(7/2))/(11*d*
f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int (a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{2C(a+b\tan(e+fx))^2(}{11d}$$

$$= -\frac{2b(4bcC-11bBd-4a}{}$$

$$= \frac{2\left(36a^2Cd^2-22abd(2c\right.}{}$$

$$= \frac{2\left(a^2B-b^2B+2ab(A-\right.}{}$$

$$= \frac{2\left(2ab(Ac-cC-Bd)+\right.}{}$$

$$= -\frac{2\left(2ab\left(c^2C+2Bcd-\right.\right.}{}$$

$$= -\frac{2\left(2ab\left(c^2C+2Bcd-\right.\right.}{}$$

$$= -\frac{2\left(2ab\left(c^2C+2Bcd-\right.\right.}{}$$

$$= -\frac{2\left(2ab\left(c^2C+2Bcd-\right.\right.}{}$$

$$= -\frac{(a-ib)^2(iA+B-iC)}{}$$

Mathematica [A] time = 6.45, size = 564, normalized size = 1.12

$$\frac{2C(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{7/2}}{11df}+\frac{2\left(\frac{b\tan(e+fx)(4aCd+11bBd-4bcC)(c+d\tan(e+fx))^{7/2}}{9df}-\frac{2\left(\frac{(c+d\tan(e+fx))^{7/2}\left(-36a^2Cd^2+\cdots\right.}{\cdots}\cdots\right)}{\cdots}\right)}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e +
f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^2*(c + d*Tan[e + f*x])^(7/2))/(11*d*f) + (2*((b*(
-4*b*c*C + 11*b*B*d + 4*a*C*d)*Tan[e + f*x]*(c + d*Tan[e + f*x])^(7/2))/(9*
d*f) - (2*(((-36*a^2*C*d^2 + 22*a*b*d*(2*c*C - 9*B*d) - b^2*(8*c^2*C - 22*B
*c*d + 99*(A - C)*d^2))*(c + d*Tan[e + f*x])^(7/2))/(14*d*f) + ((I/2)*(((99
*I)/4)*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2 + (99*(2*a*b*B - a^2*(A - C) + b
^2*(A - C))*d^2)/4)*((2*(c + d*Tan[e + f*x])^(5/2))/5 + (c - I*d)*((2*(c +
d*Tan[e + f*x])^(3/2))/3 + (c - I*d)*((2*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c - I*d]])/(-c + I*d) + 2*Sqrt[c + d*Tan[e + f*x]]))))/
f - ((I/2)*(((-99*I)/4)*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2 + (99*(2*a*b*B
- a^2*(A - C) + b^2*(A - C))*d^2)/4)*((2*(c + d*Tan[e + f*x])^(5/2))/5 + (c
 + I*d)*((2*(c + d*Tan[e + f*x])^(3/2))/3 + (c + I*d)*((2*(c + I*d)^(3/2)*A
rcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/(-c - I*d) + 2*Sqrt[c + d*T
an[e + f*x]]))))/f))/(9*d)))/(11*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.60, size = 11478, normalized size = 22.82

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f
*x+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^2*(c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) +
 C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a + b \tan\left(e + fx\right)\right)^{2} \left(c + d \tan\left(e + fx\right)\right)^{\frac{5}{2}} \left(A + B \tan\left(e + fx\right) + C \tan^{2}\left(e + fx\right)\right)\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan
(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*(c + d*tan(e + f*x))**(5/2)*(A + B*tan(e +
 f*x) + C*tan(e + f*x)**2), x)
```

3.105 $\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+\right.$

Optimal. Leaf size=353

$$\frac{2\sqrt{c+d\tan(e+fx)}\left(A\left(2acd+b\left(c^2-d^2\right)\right)+a\left(Bc^2-Bd^2-2cCd\right)-b\left(2Bcd+c^2C-Cd^2\right)\right)}{f}+\frac{2(aB+Ab-bC}{}$$

```
[Out] -(I*a+b)*(A-I*B-C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/
2))/f+(I*a-b)*(A+I*B-C)*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d
)^(1/2))/f+2*(a*(B*c^2-B*d^2-2*C*c*d)-b*(2*B*c*d+C*c^2-C*d^2)+A*(2*a*c*d+b*
(c^2-d^2)))*(c+d*tan(f*x+e))^(1/2)/f+2/3*(A*a*d+A*b*c+B*a*c-B*b*d-C*a*d-C*b
*c)*(c+d*tan(f*x+e))^(3/2)/f+2/5*(A*b+B*a-C*b)*(c+d*tan(f*x+e))^(5/2)/f-2/6
3*(-9*B*b*d-9*C*a*d+2*C*b*c)*(c+d*tan(f*x+e))^(7/2)/d^2/f+2/9*b*C*tan(f*x+e
)*(c+d*tan(f*x+e))^(7/2)/d/f
```

Rubi [A] time = 1.21, antiderivative size = 351, normalized size of antiderivative = 0.99, number of steps used = 12, number of rules used = 7, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.156, Rules used = {3637, 3630, 3528, 3539, 3537, 63, 208}

$$\frac{2\sqrt{c+d\tan(e+fx)}\left(2aAcd+aB\left(c^2-d^2\right)-2acCd+Ab\left(c^2-d^2\right)-b\left(2Bcd+c^2C-Cd^2\right)\right)}{f}+\frac{2(aB+Ab-bC}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C
*Tan[e + f*x]^2),x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c - I*d]])/f) + ((I*a - b)*(A + I*B - C)*(c + I*d)^(5/2)*ArcTanh[Sqrt
[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/f + (2*(2*a*A*c*d - 2*a*c*C*d + A*b*(c
^2 - d^2) + a*B*(c^2 - d^2) - b*(c^2*C + 2*B*c*d - C*d^2))*Sqrt[c + d*Tan[e
 + f*x]])/f + (2*(A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d)*(c + d*Tan
[e + f*x])^(3/2))/(3*f) + (2*(A*b + a*B - b*C)*(c + d*Tan[e + f*x])^(5/2))/
(5*f) - (2*(2*b*c*C - 9*b*B*d - 9*a*C*d)*(c + d*Tan[e + f*x])^(7/2))/(63*d^
2*f) + (2*b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(7/2))/(9*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
```

```
!LtQ[n, -1]
```

Rubi steps

$$\int (a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{2bC\tan(e+fx)(c+d\,\text{ta}}{9df}$$

$$= -\frac{2(2bcC-9bBd-9aC\text{d}}{63\text{d}}$$

$$= \frac{2(Ab+aB-bC)(c+d\,\text{t}}{5f}$$

$$= \frac{2(Abc+aBc-bcC+aA}{}$$

$$= \frac{2\left(2aAcd-2acCd+Ab\right.}{}$$

$$= \frac{2\left(2aAcd-2acCd+Ab\right.}{}$$

$$= \frac{2\left(2aAcd-2acCd+Ab\right.}{}$$

$$= \frac{2\left(2aAcd-2acCd+Ab\right.}{}$$

$$= -\frac{(ia+b)(A-iB-C)(c}{}$$

Mathematica [A] time = 5.36, size = 324, normalized size = 0.92

$$\frac{\frac{63}{2}id(a-ib)(A-iB-C)\left(\frac{2}{5}(c+d\tan(e+fx))^{5/2}+\frac{2}{3}(c-id)\left(\sqrt{c+d\tan(e+fx)}\,(4c+d\tan(e+fx)-3id)-3(\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*
x] + C*Tan[e + f*x]^2),x]
```

```
[Out] ((2*(-2*b*c*C + 9*b*B*d + 9*a*C*d)*(c + d*Tan[e + f*x])^(7/2))/d + 14*b*C*T
an[e + f*x]*(c + d*Tan[e + f*x])^(7/2) + ((63*I)/2)*(a - I*b)*(A - I*B - C)
*d*((2*(c + d*Tan[e + f*x])^(5/2))/5 + (2*(c - I*d)*(-3*(c - I*d)^(3/2)*Arc
Tanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]] + Sqrt[c + d*Tan[e + f*x]]*(4*
c - (3*I)*d + d*Tan[e + f*x])))/3) - ((63*I)/2)*(a + I*b)*(A + I*B - C)*d*(
(2*(c + d*Tan[e + f*x])^(5/2))/5 + (2*(c + I*d)*(-3*(c + I*d)^(3/2)*ArcTanh
[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]] + Sqrt[c + d*Tan[e + f*x]]*(4*c +
(3*I)*d + d*Tan[e + f*x])))/3))/(63*d*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.52, size = 7402, normalized size = 20.97

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)
,x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad **[F(-1)]** time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))*(c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C
*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

sympy **[F]** time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a + b \tan \left(e + f x\right)\right) \left(c + d \tan \left(e + f x\right)\right)^{\frac{5}{2}} \left(A + B \tan \left(e + f x\right) + C \tan^{2} \left(e + f x\right)\right) dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*
x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*(c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)**2), x)
```

3.106 $\int (c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+\right.$

Optimal. Leaf size=229

$$\frac{2\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\sqrt{c+d\tan(e+fx)}}{f}+\frac{2(d(A-C)+Bc)(c+d\tan(e+fx))^{3/2}}{3f}-\frac{(c-id)^{5/2}(iA+B-i}{}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f-
(B-I*(A-C))*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/f+2
*(2*c*(A-C)*d+B*(c^2-d^2))*(c+d*tan(f*x+e))^(1/2)/f+2/3*(B*c+(A-C)*d)*(c+d*
tan(f*x+e))^(3/2)/f+2/5*B*(c+d*tan(f*x+e))^(5/2)/f+2/7*C*(c+d*tan(f*x+e))^(
7/2)/d/f
```

Rubi [A] time = 0.63, antiderivative size = 229, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 6, integrand size = 35, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.171, Rules used = {3630, 3528, 3539, 3537, 63, 208}

$$\frac{2\left(2cd(A-C)+B\left(c^2-d^2\right)\right)\sqrt{c+d\tan(e+fx)}}{f}+\frac{2(d(A-C)+Bc)(c+d\tan(e+fx))^{3/2}}{3f}-\frac{(c-id)^{5/2}(iA+B-i}{}$$

Antiderivative was successfully verified.

```
[In] Int[(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/f) - ((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c + I*d]])/f + (2*(2*c*(A - C)*d + B*(c^2 - d^2))*Sqrt[c + d*Tan[
e + f*x]])/f + (2*(B*c + (A - C)*d)*(c + d*Tan[e + f*x])^(3/2))/(3*f) + (2*
B*(c + d*Tan[e + f*x])^(5/2))/(5*f) + (2*C*(c + d*Tan[e + f*x])^(7/2))/(7*d
*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3528

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[(d*(a + b*Tan[e + f*x])^m)/(f*m), x] + Int
[(a + b*Tan[e + f*x])^(m - 1)*Simp[a*c - b*d + (b*c + a*d)*Tan[e + f*x], x]
, x] /; FreeQ[{a, b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2,
 0] && GtQ[m, 0]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int (c + d\tan(e + fx))^{5/2} \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right) dx = \frac{2C(c + d\tan(e + fx))^{7/2}}{7df} + \int (A - C + B\tan$$

$$= \frac{2B(c + d\tan(e + fx))^{5/2}}{5f} + \frac{2C(c + d\tan(e +}{7df}$$

$$= \frac{2(Bc + (A - C)d)(c + d\tan(e + fx))^{3/2}}{3f} + \frac{2B}{}$$

$$= \frac{2\left(2c(A - C)d + B\left(c^2 - d^2\right)\right)\sqrt{c + d\tan(e +}}{f}$$

$$= \frac{2\left(2c(A - C)d + B\left(c^2 - d^2\right)\right)\sqrt{c + d\tan(e +}}{f}$$

$$= \frac{2\left(2c(A - C)d + B\left(c^2 - d^2\right)\right)\sqrt{c + d\tan(e +}}{f}$$

$$= \frac{2\left(2c(A - C)d + B\left(c^2 - d^2\right)\right)\sqrt{c + d\tan(e +}}{f}$$

$$= -\frac{(iA + B - iC)(c - id)^{5/2}\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+f}}{\sqrt{c-id}}\right.}{f}$$

Mathematica [A] time = 2.07, size = 262, normalized size = 1.14

$$\frac{7i(A - iB - C)\left(\frac{2}{5}(c + d\tan(e + fx))^{5/2} + \frac{2}{3}(c - id)\left(\sqrt{c + d\tan(e + fx)}\,(4c + d\tan(e + fx) - 3id) - 3(c - id)^{3/2}\,\text{ta}\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2
),x]
```

```
[Out] ((4*C*(c + d*Tan[e + f*x])^(7/2))/d + (7*I)*(A - I*B - C)*((2*(c + d*Tan[e
+ f*x])^(5/2))/5 + (2*(c - I*d)*(-3*(c - I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[
e + f*x]]/Sqrt[c - I*d]] + Sqrt[c + d*Tan[e + f*x]]*(4*c - (3*I)*d + d*Tan[
e + f*x])))/3) - (7*I)*(A + I*B - C)*((2*(c + d*Tan[e + f*x])^(5/2))/5 + (2
*(c + I*d)*(-3*(c + I*d)^(3/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*
d]] + Sqrt[c + d*Tan[e + f*x]]*(4*c + (3*I)*d + d*Tan[e + f*x])))/3))/(14*f
)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.41, size = 3614, normalized size = 15.78

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] 1/4*d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1
/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)-3/4*d/
f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c
^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c+d^2/f/(2*(c^2+d^2)^(1/2)-2
*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(
2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)+3*d/f/(2*(c^2+d^2)^(1/2)-2*
c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2
*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*c^2-3*d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arct
an((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1
/2)-2*c)^(1/2))*C*c^2+1/2/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^
2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c
^2+d^2)^(1/2)*c-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e)
)^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c^
2+d^2)^(1/2)*c^2+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/
2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*(c
^2+d^2)^(1/2)*c^2-1/2/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(
1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d
```

```
^2)^(1/2)*c-1/4*d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(
1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^
(1/2)-1/4/d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan
(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^3+2/5*B*(c+d*t
an(f*x+e))^(5/2)/f-2*d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)
^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*
C*(c^2+d^2)^(1/2)*c+2*d/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2
)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))
*A*(c^2+d^2)^(1/2)*c+1/4/d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2
*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(
c^2+d^2)^(1/2)*c^2-1/4/d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c
)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^
2+d^2)^(1/2)*c^2-1/4/d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d
^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+
d^2)^(1/2)*c^2-2*d^2/f*B*(c+d*tan(f*x+e))^(1/2)-2*d/f/(2*(c^2+d^2)^(1/2)-2*
c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2
*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*(c^2+d^2)^(1/2)*c+2*d/f/(2*(c^2+d^2)^(1/2)-2
*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(
2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*(c^2+d^2)^(1/2)*c+1/4/d/f*ln(d*tan(f*x+e)+c
+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2
*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)*c^2+2/3*d/f*A*(c+d*tan(f*x+e))^
(3/2)-2/3*d/f*C*(c+d*tan(f*x+e))^(3/2)+2/3/f*B*(c+d*tan(f*x+e))^(3/2)*c+2/f
*B*c^2*(c+d*tan(f*x+e))^(1/2)+2/7*C*(c+d*tan(f*x+e))^(7/2)/d/f+d^3/f/(2*(c^
2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)
+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C-d^3/f/(2*(c^2+d^2)^(1/2)-2*c)
^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(
c^2+d^2)^(1/2)-2*c)^(1/2))*A-d^3/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2
*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*
c)^(1/2))*C+1/4*d^2/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+1/4/d/f*
ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2
+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^3+4*d/f*A*c*(c+d*tan(f*x+e))
^(1/2)-3/4/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*
c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2+1/f/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*
c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*c^3+d^3/f/(2*(c^2+d^2)^(1/2)-2*c
)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*
(c^2+d^2)^(1/2)-2*c)^(1/2))*A+3/4*d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1
/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c
)^(1/2)*c-3*d^2/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^
(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*c+3*d
/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c
+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*c^2+3*d^2/f/(2*(c^2+
d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+
e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*c-1/4*d/f*ln((c+d*tan(f*x+e))^(
```

```
1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^
2+d^2)^(1/2)+2*c)^(1/2)*(c^2+d^2)^(1/2)+3/4*d/f*ln((c+d*tan(f*x+e))^(1/2)*(
2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)*c+1/4*d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*
c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*(c
^2+d^2)^(1/2)-3/4*d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/
2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c-3*d/f/
(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*
tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*c^2-4*d/f*c*C*(c+d*tan(
f*x+e))^(1/2)-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+
2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*c^3+3
/4/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c
-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^2-1/4*d^2/f*ln((c+d*tan
(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2)
)*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d^2/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arcta
n(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/
2)-2*c)^(1/2))*B*(c^2+d^2)^(1/2)+1/4/d/f*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+
2*c)^(1/2)*c^3-1/4/d/f*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2
)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*c^3
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorit
hm="maxima")
```

```
[Out] Timed out
```

mupad [B] time = 117.31, size = 5863, normalized size = 25.60

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] ((2*C*c^2)/(3*d*f) - (2*C*(d^3*f + c^2*d*f))/(3*d^2*f^2))*(c + d*tan(e + f*
x))^(3/2) - log(((((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*
c^5*f^2 - 10*B^2*c^3*d^2*f^2 + 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(((((-B^4*d^2*f^
4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*c^5*f^2 - 10*B^2*c^3*d^2*f^2 +
5*B^2*c*d^4*f^2)/f^4)^(1/2)*(32*B*c^4*d^2 - 32*B*d^6 + 32*c*d^2*f*(((-B^4*d
^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*c^5*f^2 - 10*B^2*c^3*d^2*f
^2 + 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (16*B
```

^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^
2))/2 - (8*B^3*c*d^2*(c^2 - 3*d^2)*(c^2 + d^2)^3)/f^3)*(((20*B^4*c^2*d^8*f^
4 - B^4*d^10*f^4 - 110*B^4*c^4*d^6*f^4 + 100*B^4*c^6*d^4*f^4 - 25*B^4*c^8*d
^2*f^4)^(1/2) + B^2*c^5*f^2 - 10*B^2*c^3*d^2*f^2 + 5*B^2*c*d^4*f^2)/(4*f^4)
)^(1/2) + log(- ((((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*
c^5*f^2 - 10*B^2*c^3*d^2*f^2 + 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(((((-B^4*d^2*f^
4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*c^5*f^2 - 10*B^2*c^3*d^2*f^2 +
5*B^2*c*d^4*f^2)/f^4)^(1/2)*(32*B*d^6 - 32*B*c^4*d^2 + 32*c*d^2*f*(((-B^4*d
^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + B^2*c^5*f^2 - 10*B^2*c^3*d^2*f
^2 + 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (16*B
^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^
2))/2 - (8*B^3*c*d^2*(c^2 - 3*d^2)*(c^2 + d^2)^3)/f^3)*((20*B^4*c^2*d^8*f^4
 - B^4*d^10*f^4 - 110*B^4*c^4*d^6*f^4 + 100*B^4*c^6*d^4*f^4 - 25*B^4*c^8*d^
2*f^4)^(1/2)/(4*f^4) + (B^2*c^5)/(4*f^2) - (5*B^2*c^3*d^2)/(2*f^2) + (5*B^2
*c*d^4)/(4*f^2))^(1/2) - log(((-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2
)^(1/2) - B^2*c^5*f^2 + 10*B^2*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2)*((
(-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - B^2*c^5*f^2 + 10*B^2
*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(32*B*c^4*d^2 - 32*B*d^6 + 32*c*
d^2*f*(-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - B^2*c^5*f^2 +
10*B^2*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)
))/(2*f) - (16*B^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 -
 15*c^4*d^2))/f^2))/2 - (8*B^3*c*d^2*(c^2 - 3*d^2)*(c^2 + d^2)^3)/f^3)*(-((
20*B^4*c^2*d^8*f^4 - B^4*d^10*f^4 - 110*B^4*c^4*d^6*f^4 + 100*B^4*c^6*d^4*f
^4 - 25*B^4*c^8*d^2*f^4)^(1/2) - B^2*c^5*f^2 + 10*B^2*c^3*d^2*f^2 - 5*B^2*c
*d^4*f^2)/(4*f^4))^(1/2) + log(- ((-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^
2)^2)^(1/2) - B^2*c^5*f^2 + 10*B^2*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2
)*(((-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - B^2*c^5*f^2 + 10
*B^2*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(32*B*d^6 - 32*B*c^4*d^2 + 3
2*c*d^2*f*(-((-B^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - B^2*c^5*f^
2 + 10*B^2*c^3*d^2*f^2 - 5*B^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(
1/2)))/(2*f) - (16*B^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d
^4 - 15*c^4*d^2))/f^2))/2 - (8*B^3*c*d^2*(c^2 - 3*d^2)*(c^2 + d^2)^3)/f^3)*
((B^2*c^5)/(4*f^2) - (20*B^4*c^2*d^8*f^4 - B^4*d^10*f^4 - 110*B^4*c^4*d^6*f
^4 + 100*B^4*c^6*d^4*f^4 - 25*B^4*c^8*d^2*f^4)^(1/2)/(4*f^4) - (5*B^2*c^3*d
^2)/(2*f^2) + (5*B^2*c*d^4)/(4*f^2))^(1/2) + ((4*B*c^2)/f - (2*B*(c^2*f + d
^2*f))/f^2)*(c + d*tan(e + f*x))^(1/2) - log(((((-((-A^4*d^2*f^4*(5*c^4 + d
^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d^2*f^2 + 5*A^2*c*d^4*
f^2)/f^4)^(1/2)*(64*A*c^3*d^3 + 64*A*c*d^5 + 32*c*d^2*f*(-((-A^4*d^2*f^4*(5
*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d^2*f^2 + 5*A^
2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) + (16*A^2*d^2*(c
 + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(-((-A
^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d
^2*f^2 + 5*A^2*c*d^4*f^2)/f^4)^(1/2))/2 - (8*A^3*d^3*(3*c^2 - d^2)*(c^2 + d
^2)^3)/f^3)*(-((20*A^4*c^2*d^8*f^4 - A^4*d^10*f^4 - 110*A^4*c^4*d^6*f^4 + 1
00*A^4*c^6*d^4*f^4 - 25*A^4*c^8*d^2*f^4)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d

```
^2*f^2 + 5*A^2*c*d^4*f^2)/(4*f^4))^(1/2) - log(((((((-A^4*d^2*f^4*(5*c^4 +
d^4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2*f^2 - 5*A^2*c*d^4
*f^2)/f^4)^(1/2)*(64*A*c^3*d^3 + 64*A*c*d^5 + 32*c*d^2*f*(((-A^4*d^2*f^4*(5
*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2*f^2 - 5*A^
2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) + (16*A^2*d^2*(c
 + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(((-A^
4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^
2*f^2 - 5*A^2*c*d^4*f^2)/f^4)^(1/2))/2 - (8*A^3*d^3*(3*c^2 - d^2)*(c^2 + d^
2)^3)/f^3)*(((20*A^4*c^2*d^8*f^4 - A^4*d^10*f^4 - 110*A^4*c^4*d^6*f^4 + 100
*A^4*c^6*d^4*f^4 - 25*A^4*c^8*d^2*f^4)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2
*f^2 - 5*A^2*c*d^4*f^2)/(4*f^4))^(1/2) + log(((((((-A^4*d^2*f^4*(5*c^4 + d^
4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2*f^2 - 5*A^2*c*d^4*f
^2)/f^4)^(1/2)*(64*A*c^3*d^3 + 64*A*c*d^5 - 32*c*d^2*f*(((-A^4*d^2*f^4*(5*c
^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2*f^2 - 5*A^2*
c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (16*A^2*d^2*(c +
 d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(((-A^4*
d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - A^2*c^5*f^2 + 10*A^2*c^3*d^2*
f^2 - 5*A^2*c*d^4*f^2)/f^4)^(1/2))/2 - (8*A^3*d^3*(3*c^2 - d^2)*(c^2 + d^2)
^3)/f^3)*((20*A^4*c^2*d^8*f^4 - A^4*d^10*f^4 - 110*A^4*c^4*d^6*f^4 + 100*A^
4*c^6*d^4*f^4 - 25*A^4*c^8*d^2*f^4)^(1/2)/(4*f^4) - (A^2*c^5)/(4*f^2) + (5*
A^2*c^3*d^2)/(2*f^2) - (5*A^2*c*d^4)/(4*f^2))^(1/2) + log(((((-((-A^4*d^2*f
^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d^2*f^2 +
 5*A^2*c*d^4*f^2)/f^4)^(1/2)*(64*A*c^3*d^3 + 64*A*c*d^5 - 32*c*d^2*f*(-((-A
^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 - 10*A^2*c^3*d
^2*f^2 + 5*A^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (
16*A^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2)
)/f^2)*(-((-A^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + A^2*c^5*f^2 -
 10*A^2*c^3*d^2*f^2 + 5*A^2*c*d^4*f^2)/f^4)^(1/2))/2 - (8*A^3*d^3*(3*c^2 -
d^2)*(c^2 + d^2)^3)/f^3)*((5*A^2*c^3*d^2)/(2*f^2) - (A^2*c^5)/(4*f^2) - (20
*A^4*c^2*d^8*f^4 - A^4*d^10*f^4 - 110*A^4*c^4*d^6*f^4 + 100*A^4*c^6*d^4*f^4
 - 25*A^4*c^8*d^2*f^4)^(1/2)/(4*f^4) - (5*A^2*c*d^4)/(4*f^2))^(1/2) - log((
8*C^3*d^3*(3*c^2 - d^2)*(c^2 + d^2)^3)/f^3 - ((((-((-C^4*d^2*f^4*(5*c^4 + d
^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 - 10*C^2*c^3*d^2*f^2 + 5*C^2*c*d^4*
f^2)/f^4)^(1/2)*(64*C*c^3*d^3 + 64*C*c*d^5 - 32*c*d^2*f*(-((-C^4*d^2*f^4*(5
*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 - 10*C^2*c^3*d^2*f^2 + 5*C^
2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (16*C^2*d^2*(c
 + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(-((-C
^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 - 10*C^2*c^3*d
^2*f^2 + 5*C^2*c*d^4*f^2)/f^4)^(1/2))/2)*(-((20*C^4*c^2*d^8*f^4 - C^4*d^10*
f^4 - 110*C^4*c^4*d^6*f^4 + 100*C^4*c^6*d^4*f^4 - 25*C^4*c^8*d^2*f^4)^(1/2)
 + C^2*c^5*f^2 - 10*C^2*c^3*d^2*f^2 + 5*C^2*c*d^4*f^2)/(4*f^4))^(1/2) - log
((8*C^3*d^3*(3*c^2 - d^2)*(c^2 + d^2)^3)/f^3 - ((((((-C^4*d^2*f^4*(5*c^4 +
d^4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^2*f^2 - 5*C^2*c*d^4
*f^2)/f^4)^(1/2)*(64*C*c^3*d^3 + 64*C*c*d^5 - 32*c*d^2*f*(((-C^4*d^2*f^4*(5
*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^2*f^2 - 5*C^
```

```
2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) - (16*C^2*d^2*(c
 + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(((-C^
4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^
2*f^2 - 5*C^2*c*d^4*f^2)/f^4)^(1/2))/2)*(((20*C^4*c^2*d^8*f^4 - C^4*d^10*f^
4 - 110*C^4*c^4*d^6*f^4 + 100*C^4*c^6*d^4*f^4 - 25*C^4*c^8*d^2*f^4)^(1/2) -
 C^2*c^5*f^2 + 10*C^2*c^3*d^2*f^2 - 5*C^2*c*d^4*f^2)/(4*f^4))^(1/2) + log((
8*C^3*d^3*(3*c^2 - d^2)*(c^2 + d^2)^3)/f^3 - ((((((-C^4*d^2*f^4*(5*c^4 + d^
4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^2*f^2 - 5*C^2*c*d^4*f
^2)/f^4)^(1/2)*(64*C*c^3*d^3 + 64*C*c*d^5 + 32*c*d^2*f*(((-C^4*d^2*f^4*(5*c
^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^2*f^2 - 5*C^2*
c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) + (16*C^2*d^2*(c +
 d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2))/f^2)*(((-C^4*
d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) - C^2*c^5*f^2 + 10*C^2*c^3*d^2*
f^2 - 5*C^2*c*d^4*f^2)/f^4)^(1/2))/2)*((20*C^4*c^2*d^8*f^4 - C^4*d^10*f^4 -
 110*C^4*c^4*d^6*f^4 + 100*C^4*c^6*d^4*f^4 - 25*C^4*c^8*d^2*f^4)^(1/2)/(4*f
^4) - (C^2*c^5)/(4*f^2) + (5*C^2*c^3*d^2)/(2*f^2) - (5*C^2*c*d^4)/(4*f^2))^
(1/2) + log((8*C^3*d^3*(3*c^2 - d^2)*(c^2 + d^2)^3)/f^3 - ((((-((-C^4*d^2*f
^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 - 10*C^2*c^3*d^2*f^2 +
 5*C^2*c*d^4*f^2)/f^4)^(1/2)*(64*C*c^3*d^3 + 64*C*c*d^5 + 32*c*d^2*f*(-((-C
^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 - 10*C^2*c^3*d
^2*f^2 + 5*C^2*c*d^4*f^2)/f^4)^(1/2)*(c + d*tan(e + f*x))^(1/2)))/(2*f) + (
16*C^2*d^2*(c + d*tan(e + f*x))^(1/2)*(c^6 - d^6 + 15*c^2*d^4 - 15*c^4*d^2)
)/f^2)*(-((-C^4*d^2*f^4*(5*c^4 + d^4 - 10*c^2*d^2)^2)^(1/2) + C^2*c^5*f^2 -
 10*C^2*c^3*d^2*f^2 + 5*C^2*c*d^4*f^2)/f^4)^(1/2))/2)*((5*C^2*c^3*d^2)/(2*f
^2) - (C^2*c^5)/(4*f^2) - (20*C^4*c^2*d^8*f^4 - C^4*d^10*f^4 - 110*C^4*c^4*
d^6*f^4 + 100*C^4*c^6*d^4*f^4 - 25*C^4*c^8*d^2*f^4)^(1/2)/(4*f^4) - (5*C^2*
c*d^4)/(4*f^2))^(1/2) + 2*c*((2*C*c^2)/(d*f) - (2*C*(d^3*f + c^2*d*f))/(d^2
*f^2))*(c + d*tan(e + f*x))^(1/2) + (2*B*(c + d*tan(e + f*x))^(5/2))/(5*f)
+ (2*A*d*(c + d*tan(e + f*x))^(3/2))/(3*f) + (2*B*c*(c + d*tan(e + f*x))^(3
/2))/(3*f) + (2*C*(c + d*tan(e + f*x))^(7/2))/(7*d*f) + (4*A*c*d*(c + d*tan
(e + f*x))^(1/2))/f
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(c + d\tan\left(e + fx\right)\right)^{\frac{5}{2}} \left(A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2), x)
```

3.107 $\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx$

Optimal. Leaf size=336

$$-\frac{2(bc-ad)^{5/2}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{7/2}f\left(a^2+b^2\right)}+\frac{2\sqrt{c+d\tan(e+fx)}\left((bc-ad)(-aCd+bBd+bc\right.}{b^3f}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a
-I*b)/f+(I*A-B-I*C)*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1
/2))/(a+I*b)/f-2*(A*b^2-a*(B*b-C*a))*(-a*d+b*c)^(5/2)*arctanh(b^(1/2)*(c+d*
tan(f*x+e))^(1/2)/(-a*d+b*c)^(1/2))/b^(7/2)/(a^2+b^2)/f+2*(b^2*d*(B*c+(A-C)
*d)+(-a*d+b*c)*(B*b*d-C*a*d+C*b*c))*(c+d*tan(f*x+e))^(1/2)/b^3/f+2/3*(B*b*d
-C*a*d+C*b*c)*(c+d*tan(f*x+e))^(3/2)/b^2/f+2/5*C*(c+d*tan(f*x+e))^(5/2)/b/f
```

Rubi [A] time = 2.81, antiderivative size = 336, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3647, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{2(bc-ad)^{5/2}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{b^{7/2}f\left(a^2+b^2\right)}+\frac{2\sqrt{c+d\tan(e+fx)}\left((bc-ad)(-aCd+bBd+bc\right.}{b^3f}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x]),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/((a - I*b)*f)) + ((I*A - B - I*C)*(c + I*d)^(5/2)*ArcTanh[Sqrt[c +
 d*Tan[e + f*x]]/Sqrt[c + I*d]])/((a + I*b)*f) - (2*(A*b^2 - a*(b*B - a*C))
*(b*c - a*d)^(5/2)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*
d]])/(b^(7/2)*(a^2 + b^2)*f) + (2*(b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(b
*c*C + b*B*d - a*C*d))*Sqrt[c + d*Tan[e + f*x]])/(b^3*f) + (2*(b*c*C + b*B*
d - a*C*d)*(c + d*Tan[e + f*x])^(3/2))/(3*b^2*f) + (2*C*(c + d*Tan[e + f*x]
)^(5/2))/(5*b*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
```

```
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{a+b\tan(e+fx)}\,dx = \frac{2C(c+d\tan(e+fx))^{5/2}}{5bf} + \frac{2\int \frac{(c+d\tan(e+fx)}{}}{}$$

$$= \frac{2(bcC+bBd-aCd)(c+d\tan(e+fx))^{3/2}}{3b^2f}$$

$$= \frac{2\left(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+b\right.}{b^3f}$$

$$= \frac{2\left(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+b\right.}{b^3f}$$

$$= \frac{2\left(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+b\right.}{b^3f}$$

$$= \frac{2\left(b^2d(Bc+(A-C)d)+(bc-ad)(bcC+b\right.}{b^3f}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)(bc-ad)^{5/2}\tanh^{-1}}{b^{7/2}\left(a^2+b^2\right)f}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e}}{\sqrt{c-id}}\right.}{(a-ib)f}$$

Mathematica [A] time = 5.31, size = 322, normalized size = 0.96

$$\frac{15\left(b^{7/2}(b-ia)(c-id)^{5/2}(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)+b^{7/2}(b+ia)(c+id)^{5/2}(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)-2(bc-ad)^{5/2}\left(a(aC-bB)+Ab^2\right)\tanh^{-1}\left(\right.\right.}{b^{5/2}\left(a^2+b^2\right)}$$

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x]),x]
```

```
[Out] ((15*(b^(7/2)*((-I)*a + b)*(A - I*B - C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c - I*d]] + b^(7/2)*(I*a + b)*(A + I*B - C)*(c + I*d)^(
5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]] - 2*(A*b^2 + a*(-(b*B)
 + a*C))*(b*c - a*d)^(5/2)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[
b*c - a*d]]))/(b^(5/2)*(a^2 + b^2)) + (30*(b^2*d*(B*c + (A - C)*d) + (b*c -
 a*d)*(b*c*C + b*B*d - a*C*d))*Sqrt[c + d*Tan[e + f*x]])/b^2 + (10*(b*c*C +
 b*B*d - a*C*d)*(c + d*Tan[e + f*x])^(3/2))/b + 6*C*(c + d*Tan[e + f*x])^(5
/2))/(15*b*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.84, size = 8698, normalized size = 25.89

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
,x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e)),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x)),x)
```

```
[Out] \text{Hanged}
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e)),x)
```

```
[Out] Timed out
```

3.108 $\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx$

Optimal. Leaf size=473

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{bf\left(a^2+b^2\right)(a+b\tan(e+fx))}+\frac{d\left(5a^2C-3abB+3Ab^2+2b^2C\right)(c+d\tan(e+fx))^{3/2}}{3b^2f\left(a^2+b^2\right)}-\frac{d\sqrt{c+d\tan}}{}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a
-I*b)^2/f-(B-I*(A-C))*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^
(1/2))/(a+I*b)^2/f+(-a*d+b*c)^(3/2)*(3*a^3*b*B*d-5*a^4*C*d-b^4*(5*A*d+2*B*c
)-a*b^3*(4*A*c-7*B*d-4*C*c)+a^2*b^2*(2*B*c-(A+9*C)*d))*arctanh(b^(1/2)*(c+d
*tan(f*x+e))^(1/2)/(-a*d+b*c)^(1/2))/b^(7/2)/(a^2+b^2)^2/f-d*(5*a^3*C*d-A*b
^2*(-a*d+b*c)-2*b^3*(B*d+2*C*c)-a^2*b*(3*B*d+5*C*c)+a*b^2*(B*c+4*C*d))*(c+d
*tan(f*x+e))^(1/2)/b^3/(a^2+b^2)/f+1/3*(3*A*b^2-3*B*a*b+5*C*a^2+2*C*b^2)*d*
(c+d*tan(f*x+e))^(3/2)/b^2/(a^2+b^2)/f-(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))
^(5/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))
```

Rubi [A] time = 3.90, antiderivative size = 473, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}} = 0.170$, Rules used = {3645, 3647, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{bf\left(a^2+b^2\right)(a+b\tan(e+fx))}+\frac{d\left(5a^2C-3abB+3Ab^2+2b^2C\right)(c+d\tan(e+fx))^{3/2}}{3b^2f\left(a^2+b^2\right)}-\frac{d\sqrt{c+d\tan}}{}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^2,x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
- I*d]])/((a - I*b)^2*f)) - ((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[Sqrt[c
 + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((a + I*b)^2*f) + ((b*c - a*d)^(3/2)*(3*
a^3*b*B*d - 5*a^4*C*d - b^4*(2*B*c + 5*A*d) - a*b^3*(4*A*c - 4*c*C - 7*B*d)
 + a^2*b^2*(2*B*c - (A + 9*C)*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]]
)/Sqrt[b*c - a*d]])/(b^(7/2)*(a^2 + b^2)^2*f) - (d*(5*a^3*C*d - A*b^2*(b*c
- a*d) - 2*b^3*(2*c*C + B*d) - a^2*b*(5*c*C + 3*B*d) + a*b^2*(B*c + 4*C*d))
*Sqrt[c + d*Tan[e + f*x]])/(b^3*(a^2 + b^2)*f) + ((3*A*b^2 - 3*a*b*B + 5*a^
2*C + 2*b^2*C)*d*(c + d*Tan[e + f*x])^(3/2))/(3*b^2*(a^2 + b^2)*f) - ((A*b^
2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^(5/2))/(b*(a^2 + b^2)*f*(a + b*Tan[
e + f*x]))
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
```

```
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^2}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{b\left(a^2+b^2\right)f(a+b\tan(e+fx))}$$

$$= \frac{\left(3Ab^2-3abB+5a^2C+2b^2C\right)d(c+d\tan}{3b^2\left(a^2+b^2\right)f}$$

$$= -\frac{d\left(5a^3Cd-Ab^2(bc-ad)-2b^3(2cC+Bd\right.}{}$$

$$= -\frac{d\left(5a^3Cd-Ab^2(bc-ad)-2b^3(2cC+Bd\right.}{}$$

$$= -\frac{d\left(5a^3Cd-Ab^2(bc-ad)-2b^3(2cC+Bd\right.}{}$$

$$= -\frac{d\left(5a^3Cd-Ab^2(bc-ad)-2b^3(2cC+Bd\right.}{}$$

$$= \frac{(bc-ad)^{3/2}\left(3a^3bBd-5a^4Cd-b^4(2Bc+5\right.}{}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e}}{\sqrt{c-id}}\right.}{(a-ib)^2f}$$

Mathematica [B] time = 6.53, size = 6112, normalized size = 12.92

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^2,x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.87, size = 14119, normalized size = 29.85

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^2,x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^2,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^2,x)
```

```
[Out] \text{Hanged}
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**2,x)
```

```
[Out] Timed out
```

3.109 $\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx$

Optimal. Leaf size=643

$$-\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{2bf\left(a^2+b^2\right)(a+b\tan(e+fx))^2}+\frac{(c+d\tan(e+fx))^{3/2}\left(-5a^4Cd+a^3bBd+a^2b^2(3Ad+4Bc-13Cd)\right.}{4b^2f\left(a^2+b^2\right)^2(a+b\tan(e+}$$

```
[Out] -(A-I*B-C)*(c-I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(I*a
+b)^3/f+(A+I*B-C)*(c+I*d)^(5/2)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/(I*a-b)^3/f+1/4*(3*a^5*b*B*d^2-15*a^6*C*d^2+a^4*b^2*d*(4*B*c+(A-46*C)*d)
-3*a^2*b^4*(8*A*c^2-6*A*d^2-16*B*c*d-8*C*c^2+21*C*d^2)-a*b^5*(56*c*(A-C)*d+
B*(24*c^2-35*d^2))-b^6*(4*c*(5*B*d+2*C*c)-A*(8*c^2-15*d^2))+2*a^3*b^3*(4*c*
(A-C)*d+B*(4*c^2+3*d^2)))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-a*d+b*c)
^(1/2))*(-a*d+b*c)^(1/2)/b^(7/2)/(a^2+b^2)^3/f-1/4*d*(3*a^3*b*B*d-15*a^4*C*
d-a*b^3*(8*A*c-11*B*d-8*C*c)+a^2*b^2*(4*B*c+(A-31*C)*d)-b^4*(7*A*d+4*B*c+8*
C*d))*(c+d*tan(f*x+e))^(1/2)/b^3/(a^2+b^2)^2/f+1/4*(a^3*b*B*d-5*a^4*C*d-b^4
*(5*A*d+4*B*c)-a*b^3*(8*A*c-9*B*d-8*C*c)+a^2*b^2*(3*A*d+4*B*c-13*C*d))*(c+d
*tan(f*x+e))^(3/2)/b^2/(a^2+b^2)^2/f/(a+b*tan(f*x+e))-1/2*(A*b^2-a*(B*b-C*a
))*(c+d*tan(f*x+e))^(5/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^2
```

Rubi [A] time = 6.07, antiderivative size = 643, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3645, 3647, 3653, 3539, 3537, 63, 208, 3634}

$$\frac{\sqrt{bc-ad}\left(2a^3b^3\left(4cd(A-C)+B\left(4c^2+3d^2\right)\right)-3a^2b^4\left(8Ac^2-6Ad^2-16Bcd-8c^2C+21Cd^2\right)+a^4b^2d(d(A-4\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^3,x]
```

```
[Out] -(((A - I*B - C)*(c - I*d)^(5/2)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c -
I*d]])/((I*a + b)^3*f)) + ((A + I*B - C)*(c + I*d)^(5/2)*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((I*a - b)^3*f) + (Sqrt[b*c - a*d]*(3*a^5*b*
B*d^2 - 15*a^6*C*d^2 + a^4*b^2*d*(4*B*c + (A - 46*C)*d) - 3*a^2*b^4*(8*A*c^
2 - 8*c^2*C - 16*B*c*d - 6*A*d^2 + 21*C*d^2) - a*b^5*(56*c*(A - C)*d + B*(2
4*c^2 - 35*d^2)) - b^6*(4*c*(2*c*C + 5*B*d) - A*(8*c^2 - 15*d^2)) + 2*a^3*b
^3*(4*c*(A - C)*d + B*(4*c^2 + 3*d^2)))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e +
 f*x]])/Sqrt[b*c - a*d]])/(4*b^(7/2)*(a^2 + b^2)^3*f) - (d*(3*a^3*b*B*d - 1
```

```
5*a^4*C*d - a*b^3*(8*A*c - 8*c*C - 11*B*d) + a^2*b^2*(4*B*c + (A - 31*C)*d)
 - b^4*(4*B*c + 7*A*d + 8*C*d))*Sqrt[c + d*Tan[e + f*x]])/(4*b^3*(a^2 + b^2
)^2*f) + ((a^3*b*B*d - 5*a^4*C*d - b^4*(4*B*c + 5*A*d) - a*b^3*(8*A*c - 8*c
*C - 9*B*d) + a^2*b^2*(4*B*c + 3*A*d - 13*C*d))*(c + d*Tan[e + f*x])^(3/2))
/(4*b^2*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x])) - ((A*b^2 - a*(b*B - a*C))*(c
 + d*Tan[e + f*x])^(5/2))/(2*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^2)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^3}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{2b\left(a^2+b^2\right)f(a+b\tan(e+fx))^2}$$

$$= \frac{\left(a^3bBd-5a^4Cd-b^4(4Bc+5Ad)-ab^3(8A\right.}{4b^2\left(a\right.}$$

$$= -\frac{d\left(3a^3bBd-15a^4Cd-ab^3(8Ac-8cC-1\right.}{}$$

$$= -\frac{d\left(3a^3bBd-15a^4Cd-ab^3(8Ac-8cC-1\right.}{}$$

$$= -\frac{d\left(3a^3bBd-15a^4Cd-ab^3(8Ac-8cC-1\right.}{}$$

$$= -\frac{d\left(3a^3bBd-15a^4Cd-ab^3(8Ac-8cC-1\right.}{}$$

$$= \frac{\sqrt{bc-ad}\left(3a^5bBd^2-15a^6Cd^2+a^4b^2d(4B\right.}{}$$

$$= -\frac{(A-iB-C)(c-id)^{5/2}\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+}}{\sqrt{c-id}}\right.}{(ia+b)^3f}$$

Mathematica [B] time = 6.89, size = 18214, normalized size = 28.33

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^3,x]
```

```
[Out] Result too large to show
```

fricas [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="giac")
```

```
[Out] Timed out
```

maple [**B**] time = 0.84, size = 20663, normalized size = 32.14

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^3,x)
```

```
[Out] result too large to display
```

maxima [**F(-2)**] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^3,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [**F(-1)**] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^3,x)
```

```
[Out] \text{Hanged}
```

sympy [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**3,x)
```

```
[Out] Timed out
```

3.110 $\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=407

$$\frac{2\sqrt{c+d\tan(e+fx)}\,\left(72a^3Cd^3-6a^2bd^2(32cC-49Bd)+21ab^2d\left(15d^2(A-C)-10Bcd+8c^2C\right)-\left(b^3\left(70cd^2(A\right.\right.\right.}{105d^4f}$$

```
[Out] (I*a+b)^3*(A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f/(c-I*d)
^(1/2)-(I*a-b)^3*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/f/
(c+I*d)^(1/2)+2/105*(72*a^3*C*d^3-6*a^2*b*d^2*(-49*B*d+32*C*c)+21*a*b^2*d*(
8*c^2*C-10*B*c*d+15*(A-C)*d^2)-b^3*(48*c^3*C-56*B*c^2*d+70*c*(A-C)*d^2+105*
B*d^3))*(c+d*tan(f*x+e))^(1/2)/d^4/f+2/105*b*(35*b*(A*b+B*a-C*b)*d^2+4*(-a*
d+b*c)*(-7*B*b*d-6*C*a*d+6*C*b*c))*(c+d*tan(f*x+e))^(1/2)*tan(f*x+e)/d^3/f-
2/35*(-7*B*b*d-6*C*a*d+6*C*b*c)*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2/d
^2/f+2/7*C*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^3/d/f
```

Rubi [A] time = 1.70, antiderivative size = 407, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3647, 3637, 3630, 3539, 3537, 63, 208}

$$\frac{2\sqrt{c+d\tan(e+fx)}\,\left(-6a^2bd^2(32cC-49Bd)+72a^3Cd^3+21ab^2d\left(15d^2(A-C)-10Bcd+8c^2C\right)+b^3\left(-\left(70cd^2\right.\right.\right.}{105d^4f}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sqrt[c
 + d*Tan[e + f*x]],x]
```

```
[Out] ((I*a + b)^3*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])
/(Sqrt[c - I*d]*f) - ((I*a - b)^3*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e +
f*x]]/Sqrt[c + I*d]])/(Sqrt[c + I*d]*f) + (2*(72*a^3*C*d^3 - 6*a^2*b*d^2*(3
2*c*C - 49*B*d) + 21*a*b^2*d*(8*c^2*C - 10*B*c*d + 15*(A - C)*d^2) - b^3*(4
8*c^3*C - 56*B*c^2*d + 70*c*(A - C)*d^2 + 105*B*d^3))*Sqrt[c + d*Tan[e + f*
x]])/(105*d^4*f) + (2*b*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*
C - 7*b*B*d - 6*a*C*d))*Tan[e + f*x]*Sqrt[c + d*Tan[e + f*x]])/(105*d^3*f)
- (2*(6*b*c*C - 7*b*B*d - 6*a*C*d)*(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e
+ f*x]])/(35*d^2*f) + (2*C*(a + b*Tan[e + f*x])^3*Sqrt[c + d*Tan[e + f*x]])
/(7*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
```

```
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx=\frac{2C(a+b\tan(e+fx))^3\sqrt{c+d\tan(e+fx)}}{7df}+$$

$$=-\frac{2(6bcC-7bBd-6aCd)(a+b\tan(e+fx))^2}{35d^2f}$$

$$=\frac{2b\left(35b(Ab+aB-bC)d^2+4(bc-ad)(6bcC\right.}{10}$$

$$=\frac{2\left(72a^3Cd^3-6a^2bd^2(32cC-49Bd)+21ab^2d\right.}{}$$

$$=\frac{2\left(72a^3Cd^3-6a^2bd^2(32cC-49Bd)+21ab^2d\right.}{}$$

$$=\frac{2\left(72a^3Cd^3-6a^2bd^2(32cC-49Bd)+21ab^2d\right.}{}$$

$$=\frac{2\left(72a^3Cd^3-6a^2bd^2(32cC-49Bd)+21ab^2d\right.}{}$$

$$=-\frac{(a-ib)^3(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right.}{\sqrt{c-id}\,f}$$

Mathematica [B] time = 6.43, size = 1200, normalized size = 2.95

$$\frac{2C\sqrt{c+d\tan(e+fx)}\,(a+b\tan(e+fx))^3}{7df}+\frac{2\left(\frac{(-6bcC+6adC+7bBd)\sqrt{c+d\tan(e+fx)}\,(a+b\tan(e+fx))^2}{5df}+\frac{2\left(\frac{b\left(35b(Ab-Cb+aB)d^2+4(bc\cdots\right.}{\cdots}\right.}{\cdots}\right.}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^3*Sqrt[c + d*Tan[e + f*x]])/(7*d*f) + (2*(((-6*b*
c*C + 7*b*B*d + 6*a*C*d)*(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e + f*x]])/(
5*d*f) + (2*((b*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*
B*d - 6*a*C*d))*Tan[e + f*x]*Sqrt[c + d*Tan[e + f*x]])/(6*d*f) - (2*((I*Sqr
t[c - I*d]*((b*c*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b
*B*d - 6*a*C*d)))/4 + (3*a*d*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6
*b*c*C - 7*b*B*d - 6*a*C*d)))/8 - (3*a*d*(-5*a*d*(6*b*c*C - a*(7*A - C)*d)
+ (4*b*c + a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/8 - b*((-105*(a^2*B - b^2*B
 + 2*a*b*(A - C))*d^3)/8 + (c*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(
6*b*c*C - 7*b*B*d - 6*a*C*d)))/4) + ((3*I)/2)*d*((35*a*(a^2*B - b^2*B + 2*a
*b*(A - C))*d^2)/4 - (b*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*
C - 7*b*B*d - 6*a*C*d)))/4 + (b*(-5*a*d*(6*b*c*C - a*(7*A - C)*d) + (4*b*c
+ a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/4))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*((b*c*(35*b*(A*b + a*B -
 b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/4 + (3*a*d*(35*b*
(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/8 - (
3*a*d*(-5*a*d*(6*b*c*C - a*(7*A - C)*d) + (4*b*c + a*d)*(6*b*c*C - 7*b*B*d
- 6*a*C*d)))/8 - b*((-105*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (c*(35*b
*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/4) -
 ((3*I)/2)*d*((35*a*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2)/4 - (b*(35*b*(A*b
+ a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/4 + (b*(-5
*a*d*(6*b*c*C - a*(7*A - C)*d) + (4*b*c + a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d
)))/4))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((-c - I*d)*f) + (
2*((-3*a*d*(35*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d -
 6*a*C*d)))/8 + b*((-105*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/8 + (c*(35*b*
```

```
(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 7*b*B*d - 6*a*C*d)))/4))*S
qrt[c + d*Tan[e + f*x]])/(d*f)))/(3*d)))/(5*d)))/(7*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.46, size = 25426, normalized size = 62.47

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/
2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [B] time = 122.08, size = 28858, normalized size = 70.90

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(1/2),x)
```

```
[Out] atan(((((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A
*a^2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^6*
c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2
*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c
^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8
 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f
^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2
*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^
2*f^4)))^(1/2))*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2
+ 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*
A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6
*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^
10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 2
4*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^
3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/
2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f
^2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5
*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*
d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^1
0 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/
2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*
d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)
/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^
3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*ta
n(e + f*x))^(1/2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^
2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 16
0*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 +
 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*
a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 -
 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*
a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^6*c*f^2 - 8*A^2
*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^
2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d
^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6
*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^
6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 +
60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2
) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*
b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 +
 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*
```

a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4
*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^
4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24
*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b
^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((16*(
6*A^3*a^4*b^5*d^2 - A^3*b^9*d^2 + 8*A^3*a^6*b^3*d^2 + 3*A^3*a^8*b*d^2))/f^3
 + (((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^
2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^6*c*f
^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^
2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*
f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 +
20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2
+ 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^
4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f
^4)))^(1/2))*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 4
8*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2
*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^
4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*
b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A
^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b
^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*
(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)
*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*
d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f
^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 +
 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2)
- 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f
^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(1
6*(c^2*f^4 + d^2*f^4)))^(1/2) + (((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2
+ 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e +
f*x))^(1/2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48
*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*
a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4
*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b
^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^
2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^
3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c
*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 12
0*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4
)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 +
 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^
2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2
*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (1
6*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^
2 - 15*A^2*a^4*b^2*d^2))/f^2)*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^

2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) - 4*A^2*a^6*c*f^2 + 4*A^2*b^6*c*f^2 - 24*A^2*a*b^5*d*f^2 - 24*A^2*a^5*b*d*f^2 - 60*A^2*a^2*b^4*c*f^2 + 60*A^2*a^4*b^2*c*f^2 + 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i + atan(((((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)

^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15
*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4
*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2
+ 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 -
A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)*(-(((8*A^2*a^6
*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^
2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*
c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^
8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*
f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^
2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d
^2*f^4)))^(1/2)*1i)/((16*(6*A^3*a^4*b^5*d^2 - A^3*b^9*d^2 + 8*A^3*a^6*b^3*d
^2 + 3*A^3*a^8*b*d^2))/f^3 + (((8*(4*A*a^3*d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4
*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x
))^(1/2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A
^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^
3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a
^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2
))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*
a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*
d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*
f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120
*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)
*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 +
15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2
 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*
a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16
*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 - A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2
 - 15*A^2*a^4*b^2*d^2))/f^2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^
2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^
2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12
+ A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*
b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a
*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f
^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (((8*(4*A*a^3*
d^3*f^2 - 12*A*a*b^2*d^3*f^2 + 4*A*b^3*c*d^2*f^2 - 12*A*a^2*b*c*d^2*f^2))/f
^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c
*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 12
0*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4
)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 +
 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^
2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2
*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(
((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f

^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)
^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15
*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4
*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2
+ 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^6*d^2 -
A^2*b^6*d^2 + 15*A^2*a^2*b^4*d^2 - 15*A^2*a^4*b^2*d^2))/f^2)*(-(((8*A^2*a^6
*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f^2 + 48*A^2*a^5*b*d*f^2 + 120*A^
2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 160*A^2*a^3*b^3*d*f^2)^2/4 - (16*
c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12 + 6*A^4*a^2*b^10 + 15*A^4*a^4*b^
8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4*a^10*b^2))^(1/2) + 4*A^2*a^6*c*
f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2 + 24*A^2*a^5*b*d*f^2 + 60*A^2*a^
2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d
^2*f^4)))^(1/2)))*(-(((8*A^2*a^6*c*f^2 - 8*A^2*b^6*c*f^2 + 48*A^2*a*b^5*d*f
^2 + 48*A^2*a^5*b*d*f^2 + 120*A^2*a^2*b^4*c*f^2 - 120*A^2*a^4*b^2*c*f^2 - 1
60*A^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^12 + A^4*b^12
+ 6*A^4*a^2*b^10 + 15*A^4*a^4*b^8 + 20*A^4*a^6*b^6 + 15*A^4*a^8*b^4 + 6*A^4
*a^10*b^2))^(1/2) + 4*A^2*a^6*c*f^2 - 4*A^2*b^6*c*f^2 + 24*A^2*a*b^5*d*f^2
+ 24*A^2*a^5*b*d*f^2 + 60*A^2*a^2*b^4*c*f^2 - 60*A^2*a^4*b^2*c*f^2 - 80*A^2
*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - (c + d*tan(e + f*x))^(
1/2)*(2*c*((8*B*b^3*c - 6*B*a*b^2*d)/(d^3*f) - (4*B*b^3*c)/(d^3*f)) - (12*B
*b^3*c^2 + 6*B*a^2*b*d^2 - 18*B*a*b^2*c*d)/(d^3*f) + (2*B*b^3*(d^5*f + c^2*
d^3*f))/(d^6*f^2)) + atan(((((8*(4*B*b^3*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B
*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))
^(1/2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2
*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*
b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2
*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))
^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^
5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*
f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^
2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B
^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(
B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15
*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 -
24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^
4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(
c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 -
15*B^2*a^4*b^2*d^2))/f^2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*
a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*
c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 +
B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^
4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b
^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2
+ 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*B*b^3

*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/
f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*
c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 1
20*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^
4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6
+ 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f
^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^
2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-
(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*
f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2
)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 1
5*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) -
4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2
 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*
(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 -
 B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*b^2*d^2))/f^2)*(-(((8*B^2*a^
6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B
^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16
*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b
^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c
*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a
^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 +
d^2*f^4)))^(1/2)*1i)/((((8*(4*B*b^3*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*
c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2
)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*
b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d
*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10
 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2
) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d
*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/
(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 4
8*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^
4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a
^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*
a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B
^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2
*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d
*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B
^2*a^4*b^2*d^2))/f^2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5
*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2
 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b
^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6
*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*
f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80
*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(8*B^3*a^3*b^6*d^

2 - B^3*a^9*d^2 + 6*B^3*a^5*b^4*d^2 + 3*B^3*a*b^8*d^2))/f^3 + (((8*(4*B*b^3
*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/
f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*
c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 1
20*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^
4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6
+ 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f
^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^
2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-
(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*
f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2
)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 1
5*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) -
4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2
 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*
(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 -
 B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*b^2*d^2))/f^2)*(-(((8*B^2*a^
6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B
^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16
*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b
^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) - 4*B^2*a^6*c
*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a
^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 +
d^2*f^4)))^(1/2)))*(-(((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*
f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 -
160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12
 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^
4*a^10*b^2))^(1/2) - 4*B^2*a^6*c*f^2 + 4*B^2*b^6*c*f^2 - 24*B^2*a*b^5*d*f^2
 - 24*B^2*a^5*b*d*f^2 - 60*B^2*a^2*b^4*c*f^2 + 60*B^2*a^4*b^2*c*f^2 + 80*B^
2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i + atan(((((8*(4*B*b^3*d
^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/f^
3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f
^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*
B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*
(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 1
5*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2
+ 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a
^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8
*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2
+ 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/
4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^
4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^
2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 6
0*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2
*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 - B^2

```
*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*b^2*d^2))/f^2)*((((8*B^2*a^6*c*f
^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^
2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*
f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 +
20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2
- 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^
4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f
^4)))^(1/2)*1i - (((8*(4*B*b^3*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2
*f^2 + 12*B*a*b^2*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(((
(8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^
2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^
2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*
B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*
B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 +
 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c
^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a
*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c
*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B
^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4
 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^
5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2
- 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e
+ f*x))^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*
b^2*d^2))/f^2)*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 +
 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B
^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*
B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^1
0*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24
*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3
*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*B*b^3*d^3*f^2 - 12
*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 + 12*B*a*b^2*c*d^2*f^2))/f^3 - 64*c*d^
2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2
*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2
*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 +
 B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b
^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*
b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^
2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^6*c*
f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a
^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2
*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 +
 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2
 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b
^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*
f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 +
```

```
15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*b^2*d^2))/f^2)*((((8*B^2*a^6*c*f^2 - 8*B^2*
b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2
 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^
2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*
b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6
*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 6
0*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)
 - (16*(8*B^3*a^3*b^6*d^2 - B^3*a^9*d^2 + 6*B^3*a^5*b^4*d^2 + 3*B^3*a*b^8*d
^2))/f^3 + (((8*(4*B*b^3*d^3*f^2 - 12*B*a^2*b*d^3*f^2 - 4*B*a^3*c*d^2*f^2 +
 12*B*a*b^2*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2
*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 12
0*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 -
(16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^
4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^
6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^
2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4
 + d^2*f^4)))^(1/2))*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d
*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 -
 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^1
2 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B
^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^
2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B
^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x)
)^(1/2)*(B^2*a^6*d^2 - B^2*b^6*d^2 + 15*B^2*a^2*b^4*d^2 - 15*B^2*a^4*b^2*d^
2))/f^2)*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f^2 + 48*B^2*a*b^5*d*f^2 + 48*B^
2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*B^2*a^4*b^2*c*f^2 - 160*B^2*a^3
*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^12 + B^4*b^12 + 6*B^4*a^
2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 15*B^4*a^8*b^4 + 6*B^4*a^10*b^2)
)^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2 + 24*B^2*a*b^5*d*f^2 + 24*B^2*a
^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d
*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)))*((((8*B^2*a^6*c*f^2 - 8*B^2*b^6*c*f
^2 + 48*B^2*a*b^5*d*f^2 + 48*B^2*a^5*b*d*f^2 + 120*B^2*a^2*b^4*c*f^2 - 120*
B^2*a^4*b^2*c*f^2 - 160*B^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*
(B^4*a^12 + B^4*b^12 + 6*B^4*a^2*b^10 + 15*B^4*a^4*b^8 + 20*B^4*a^6*b^6 + 1
5*B^4*a^8*b^4 + 6*B^4*a^10*b^2))^(1/2) + 4*B^2*a^6*c*f^2 - 4*B^2*b^6*c*f^2
+ 24*B^2*a*b^5*d*f^2 + 24*B^2*a^5*b*d*f^2 + 60*B^2*a^2*b^4*c*f^2 - 60*B^2*a
^4*b^2*c*f^2 - 80*B^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - (
c + d*tan(e + f*x))^(3/2)*((2*c*((10*C*b^3*c - 6*C*a*b^2*d)/(d^4*f) - (4*C*
b^3*c)/(d^4*f)))/3 - (20*C*b^3*c^2 + 6*C*a^2*b*d^2 - 24*C*a*b^2*c*d)/(3*d^4
*f) + (2*C*b^3*(d^6*f + c^2*d^4*f))/(3*d^8*f^2)) - atan(((((8*(4*C*a^3*d^3*
f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*d^2*f^2))/f^3 -
 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2
+ 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2
*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^
4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C
```

^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4

+ 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5
*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 +
 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^6*c*f^2
 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*
b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^
4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20
*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 +
4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*
c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4
)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*
C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6
*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 -
120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f
^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6
 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*
f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C
^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) +
(((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b
*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^6*c*f^2
- 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b
^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4
 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*
C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4
*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c
*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)
))^(1/2))*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C
^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^
3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a
^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2
))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*
a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*
d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^
2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*((
((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f
^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)
^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15
*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4
*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 24*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2
- 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2) + (16*(6*C^3*a^4*b^5*d^2 - C^3*b^9*d^2 + 8*C^3*a
^6*b^3*d^2 + 3*C^3*a^8*b*d^2))/f^3))*((((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2
+ 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2
*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^
4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C
^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) - 4*C^2*a^6*c*f^2 + 4*C^2*b^6*c*f^2 - 2

```
4*C^2*a*b^5*d*f^2 - 24*C^2*a^5*b*d*f^2 - 60*C^2*a^2*b^4*c*f^2 + 60*C^2*a^4*
b^2*c*f^2 + 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - atan
(((((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2
*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^6*c*f
^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^
2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*
f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 +
20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2
- 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^
4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f
^4)))^(1/2))*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 +
48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^
2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C
^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10
*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*
C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*
b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)
*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2
)*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*
b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d
*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10
 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2
) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d
*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/
(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3
*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan
(e + f*x))^(1/2)*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^
2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 16
0*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 +
 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*
a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 +
 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*
a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*C^2*a^6*c*f^2 - 8*C^
2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f
^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*
d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^
6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b
^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 -
 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/
2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2
*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2
 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^
2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C
^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*
C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 +
```

```
24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4
*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((
8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*
d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^6*c*f^2 -
8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4
*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 +
 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^
4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C
^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f
^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))
^(1/2))*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^
2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3
*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^
2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2)
)^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a
^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d
*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(C^2
*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2 - 15*C^2*a^4*b^2*d^2))/f^2)*(-(
((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f
^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)
^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15
*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4
*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2
+ 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2) + (((8*(4*C*a^3*d^3*f^2 - 12*C*a*b^2*d^3*f^2 + 4
*C*b^3*c*d^2*f^2 - 12*C*a^2*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x
))^(1/2)*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C
^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^
3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a
^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2
))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*
a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*
d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*
f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120
*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)
*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 +
15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2
 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*
a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16
*(c + d*tan(e + f*x))^(1/2)*(C^2*a^6*d^2 - C^2*b^6*d^2 + 15*C^2*a^2*b^4*d^2
 - 15*C^2*a^4*b^2*d^2))/f^2)*(-(((8*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^
2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2 + 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^
2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12
+ C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*
b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a
```

```
*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 60*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f
^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(6*C^3*a^4
*b^5*d^2 - C^3*b^9*d^2 + 8*C^3*a^6*b^3*d^2 + 3*C^3*a^8*b*d^2))/f^3))*(-(((8
*C^2*a^6*c*f^2 - 8*C^2*b^6*c*f^2 + 48*C^2*a*b^5*d*f^2 + 48*C^2*a^5*b*d*f^2
+ 120*C^2*a^2*b^4*c*f^2 - 120*C^2*a^4*b^2*c*f^2 - 160*C^2*a^3*b^3*d*f^2)^2/
4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^12 + C^4*b^12 + 6*C^4*a^2*b^10 + 15*C^
4*a^4*b^8 + 20*C^4*a^6*b^6 + 15*C^4*a^8*b^4 + 6*C^4*a^10*b^2))^(1/2) + 4*C^
2*a^6*c*f^2 - 4*C^2*b^6*c*f^2 + 24*C^2*a*b^5*d*f^2 + 24*C^2*a^5*b*d*f^2 + 6
0*C^2*a^2*b^4*c*f^2 - 60*C^2*a^4*b^2*c*f^2 - 80*C^2*a^3*b^3*d*f^2)/(16*(c^2
*f^4 + d^2*f^4)))^(1/2)*2i - ((6*A*b^3*c - 6*A*a*b^2*d)/(d^2*f) - (4*A*b^3*
c)/(d^2*f))*(c + d*tan(e + f*x))^(1/2) - ((8*B*b^3*c - 6*B*a*b^2*d)/(3*d^3*
f) - (4*B*b^3*c)/(3*d^3*f))*(c + d*tan(e + f*x))^(3/2) - ((10*C*b^3*c - 6*C
*a*b^2*d)/(5*d^4*f) - (4*C*b^3*c)/(5*d^4*f))*(c + d*tan(e + f*x))^(5/2) + (
c + d*tan(e + f*x))^(1/2)*((2*C*a^3*d^3 - 20*C*b^3*c^3 + 36*C*a*b^2*c^2*d -
 18*C*a^2*b*c*d^2)/(d^4*f) - 2*c*(2*c*((10*C*b^3*c - 6*C*a*b^2*d)/(d^4*f) -
 (4*C*b^3*c)/(d^4*f)) - (20*C*b^3*c^2 + 6*C*a^2*b*d^2 - 24*C*a*b^2*c*d)/(d^
4*f) + (2*C*b^3*(d^6*f + c^2*d^4*f))/(d^8*f^2)) + ((d^6*f + c^2*d^4*f)*((10
*C*b^3*c - 6*C*a*b^2*d)/(d^4*f) - (4*C*b^3*c)/(d^4*f)))/(d^4*f)) + (2*A*b^3
*(c + d*tan(e + f*x))^(3/2))/(3*d^2*f) + (2*B*b^3*(c + d*tan(e + f*x))^(5/2
))/(5*d^3*f) + (2*C*b^3*(c + d*tan(e + f*x))^(7/2))/(7*d^4*f)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(e+fx\right)\right)^3\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)}{\sqrt{c+d\tan\left(e+fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(1/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**3*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/s
qrt(c + d*tan(e + f*x)), x)
```

3.111 $\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=287

$$\frac{2\sqrt{c+d\tan(e+fx)}\left(12a^2Cd^2-10abd(2cC-3Bd)+b^2\left(15d^2(A-C)-10Bcd+8c^2C\right)\right)}{15d^3f}-\frac{(a-ib)^2(B+i(A-C}{f}$$

```
[Out] -(a-I*b)^2*(B+I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f/(c-I
*d)^(1/2)+(a+I*b)^2*(I*A-B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/f/(c+I*d)^(1/2)+2/15*(12*a^2*C*d^2-10*a*b*d*(-3*B*d+2*C*c)+b^2*(8*c^2*C-
10*B*c*d+15*(A-C)*d^2))*(c+d*tan(f*x+e))^(1/2)/d^3/f-2/15*b*(-5*B*b*d-4*C*a
*d+4*C*b*c)*(c+d*tan(f*x+e))^(1/2)*tan(f*x+e)/d^2/f+2/5*C*(c+d*tan(f*x+e))^
(1/2)*(a+b*tan(f*x+e))^2/d/f
```

Rubi [A] time = 1.00, antiderivative size = 287, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3647, 3637, 3630, 3539, 3537, 63, 208}

$$\frac{2\sqrt{c+d\tan(e+fx)}\left(12a^2Cd^2-10abd(2cC-3Bd)+b^2\left(15d^2(A-C)-10Bcd+8c^2C\right)\right)}{15d^3f}-\frac{(a-ib)^2(B+i(A-C}{f}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sqrt[c
 + d*Tan[e + f*x]],x]
```

```
[Out] -(((a - I*b)^2*(B + I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/(Sqrt[c - I*d]*f)) + ((a + I*b)^2*(I*A - B - I*C)*ArcTanh[Sqrt[c + d*T
an[e + f*x]]/Sqrt[c + I*d]])/(Sqrt[c + I*d]*f) + (2*(12*a^2*C*d^2 - 10*a*b*
d*(2*c*C - 3*B*d) + b^2*(8*c^2*C - 10*B*c*d + 15*(A - C)*d^2))*Sqrt[c + d*T
an[e + f*x]])/(15*d^3*f) - (2*b*(4*b*c*C - 5*b*B*d - 4*a*C*d)*Tan[e + f*x]*
Sqrt[c + d*Tan[e + f*x]])/(15*d^2*f) + (2*C*(a + b*Tan[e + f*x])^2*Sqrt[c +
 d*Tan[e + f*x]])/(5*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
```

```
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx=\frac{2C(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}{5df}$$

$$=-\frac{2b(4bcC-5bBd-4aCd)\tan(e+fx)\sqrt{c+}}{15d^2f}$$

$$=\frac{2\left(12a^2Cd^2-10abd(2cC-3Bd)+b^2\left(8c^2C\right.\right.}{15}$$

$$=\frac{2\left(12a^2Cd^2-10abd(2cC-3Bd)+b^2\left(8c^2C\right.\right.}{15}$$

$$=\frac{2\left(12a^2Cd^2-10abd(2cC-3Bd)+b^2\left(8c^2C\right.\right.}{15}$$

$$=\frac{2\left(12a^2Cd^2-10abd(2cC-3Bd)+b^2\left(8c^2C\right.\right.}{15}$$

$$=-\frac{(a-ib)^2(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+f}}{\sqrt{c-id}}\right.}{\sqrt{c-id}\,f}$$

Mathematica [A] time = 6.03, size = 275, normalized size = 0.96

$$\frac{\frac{2\sqrt{c+d\tan(e+fx)}\left(12a^2Cd^2+10abd(3Bd-2cC)+b^2\left(15d^2(A-C)-10Bcd+8c^2C\right)\right)}{d^2}-\frac{15d(a-ib)^2(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}}+\frac{15id(a+ib)^2(A}{}}{15a}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] ((-15*(a - I*b)^2*(I*A + B - I*C)*d*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c
 - I*d]])/Sqrt[c - I*d] + ((15*I)*(a + I*b)^2*(A + I*B - C)*d*ArcTanh[Sqrt[
c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/Sqrt[c + I*d] + (2*(12*a^2*C*d^2 + 10*a
*b*d*(-2*c*C + 3*B*d) + b^2*(8*c^2*C - 10*B*c*d + 15*(A - C)*d^2))*Sqrt[c +
 d*Tan[e + f*x]])/d^2 + (2*b*(-4*b*c*C + 5*b*B*d + 4*a*C*d)*Tan[e + f*x]*Sq
rt[c + d*Tan[e + f*x]])/d + 6*C*(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e + f
*x]])/(15*d*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.41, size = 18289, normalized size = 63.72

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/
2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(1/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [B] time = 47.98, size = 21254, normalized size = 74.06

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(1/2),x)
```

```
[Out] atan(((((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 -
64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 -
 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*
c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 +
4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*
f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^
(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*
a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^
8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2
*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24
*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*
x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d^2))/f^2)*((((8*C^2*a
^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C
^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^
4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2
*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2
)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3
*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*
C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 -
 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 +
 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 -
4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*
c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f
^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 -
(16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^
4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^
3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4
)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C
^2*a^2*b^2*d^2))/f^2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*
d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d
^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)
)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a
```

^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((
16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 - 64*c*d^2*
(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a
*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 +
16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6
*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*
C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((
((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f
^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b
^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^
2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*
b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)
*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d^2))/f^2)*((((8*C^2*a^4*c*f^2
+ 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^
2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6
+ 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^
2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^
2*f^4 + d^2*f^4)))^(1/2) + (((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a
*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c*f^
2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*
b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b
^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*
f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*
a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4
+ 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^
6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16
*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) +
(16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d
^2))/f^2)*((((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C
^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4
*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*
C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 +
24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (32*(2*C^3*a^3*b^3
*d^2 + C^3*a*b^5*d^2 + C^3*a^5*b*d^2))/f^3))*((((8*C^2*a^4*c*f^2 + 8*C^2*b^
4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2
/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a
^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 + 16*C^2
*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^
2*f^4)))^(1/2)*2i - atan(((((8*(4*B*a^2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B
*a*b*d^3*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^
2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*
b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b
^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*
f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(

```
c^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*
a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4
+ 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^
6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16
*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) +
(16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*d
^2))/f^2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B
^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4
*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*
B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 -
 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*B*a^2*c
*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B*a*b*d^3*f^2))/f^3 + 64*c*d^2*(c + d*tan(
e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2
+ 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4
)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2
) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d
*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^4
*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2
*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*
a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b
^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/
(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d
^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*d^2))/f^2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4
*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/
4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^
4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*
a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2
*f^4)))^(1/2)*1i)/((((8*(4*B*a^2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B*a*b*d^
3*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c*f^2 + 8*B
^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f
^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*
B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 1
6*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4
 + d^2*f^4)))^(1/2))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d
*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^
2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))
^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^
3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c
+ d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*d^2))/f^
2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*
b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 +
B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4
*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2
*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(B^3*a^6*d^2 - B^3*b^
6*d^2 - B^3*a^2*b^4*d^2 + B^3*a^4*b^2*d^2))/f^3 + (((8*(4*B*a^2*c*d^2*f^2 -
```

```
4*B*b^2*c*d^2*f^2 + 8*B*a*b*d^3*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^
(1/2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a
^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8
 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*
a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*
B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*B^2*a^4*c*f^2 + 8
*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c
*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 +
6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 -
 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f
^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b
^4*d^2 - 6*B^2*a^2*b^2*d^2))/f^2)*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 3
2*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^
2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*
B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^
2 + 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1
/2)))*((((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a
^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8
 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*
a^4*c*f^2 + 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 - 24*
B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - atan(((((16*(2*A*b^
2*d^3*f^2 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan
(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2
 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^
4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/
2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*
d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^
4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^
2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4
*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*
b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)
/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^4*
d^2 + A^2*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^
4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2
/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a
^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2
*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^
2*f^4)))^(1/2)*1i - (((16*(2*A*b^2*d^3*f^2 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^
2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A
^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f
^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*
A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 1
6*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4
 + d^2*f^4)))^(1/2))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d
*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^
```

```
2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))
^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^
3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c
+ d*tan(e + f*x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^
2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*
b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 +
A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4
*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2
*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((16*(2*A*b^2*d^3*f^2
 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x)
)^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2
*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a
^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^
2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 2
4*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^4*c*f^2 +
 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2
*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6
+ 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2
 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2
*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^4*d^2 + A^2
*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 -
 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*
c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 +
4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*
f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^
(1/2) + (((16*(2*A*b^2*d^3*f^2 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3
+ 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2
 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (1
6*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4
+ 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*
d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4))
)^(1/2))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^
2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*
a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A
^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 +
24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e +
f*x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^2)*((((8*A^2
*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48
*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*
A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A
^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f
^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (32*(2*A^3*a^3*b^3*d^2 + A^3*a*b^5*d^
2 + A^3*a^5*b*d^2))/f^3))*((((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*
b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 +
16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*
```

```
b^2))^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A
^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i -
 atan(((((16*(2*A*b^2*d^3*f^2 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3 -
 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2
 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (1
6*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4
+ 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*
d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4))
)^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A
^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4
*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*
A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 -
 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e +
 f*x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^2)*(-(((8*A
^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 -
48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 +
4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4
*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c
*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((16*(2*A*b^2*d^3*f^2 - 2*A*a^2
*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-
(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*
f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*
b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f
^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2
*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b
^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^
2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*
a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^
2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d
^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2
 - 6*A^2*a^2*b^2*d^2))/f^2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2
*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4
 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a
^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 1
6*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1
i)/((((16*(2*A*b^2*d^3*f^2 - 2*A*a^2*d^3*f^2 + 4*A*a*b*c*d^2*f^2))/f^3 - 64
*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 -
32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c
^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4
*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f
^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(
1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*
a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^
8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2
*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24
```

```
*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*
x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2 - 6*A^2*a^2*b^2*d^2))/f^2)*(-(((8*A^2*
a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*
A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A
^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^
2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^
2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (((16*(2*A*b^2*d^3*f^2 - 2*A*a^2*d^3*f
^2 + 4*A*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A
^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 -
48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 +
4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4
*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c
*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f
^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 -
(16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^
4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^
3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4
)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^4*d^2 + A^2*b^4*d^2 - 6*A
^2*a^2*b^2*d^2))/f^2)*(-(((8*A^2*a^4*c*f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3
*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*
d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2
))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*
a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (32*(
2*A^3*a^3*b^3*d^2 + A^3*a*b^5*d^2 + A^3*a^5*b*d^2))/f^3))*(-(((8*A^2*a^4*c*
f^2 + 8*A^2*b^4*c*f^2 - 32*A^2*a*b^3*d*f^2 + 32*A^2*a^3*b*d*f^2 - 48*A^2*a^
2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2
*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*
c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)/(16
*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - ((6*B*b^2*c - 4*B*a*b*d)/(d^2*f) - (4*B*b
^2*c)/(d^2*f))*(c + d*tan(e + f*x))^(1/2) - ((8*C*b^2*c - 4*C*a*b*d)/(3*d^3
*f) - (4*C*b^2*c)/(3*d^3*f))*(c + d*tan(e + f*x))^(3/2) - (c + d*tan(e + f*
x))^(1/2)*(2*c*((8*C*b^2*c - 4*C*a*b*d)/(d^3*f) - (4*C*b^2*c)/(d^3*f)) - (2
*C*a^2*d^2 + 12*C*b^2*c^2 - 12*C*a*b*c*d)/(d^3*f) + (2*C*b^2*(d^5*f + c^2*d
^3*f))/(d^6*f^2)) - atan(((((8*(4*B*a^2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B
*a*b*d^3*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f
^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2
*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*
b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c
*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*
(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^
2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^
4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*
a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 -
16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)
+ (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2
```

```
*d^2))/f^2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 3
2*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(
B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) -
 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^
2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*B*a^
2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B*a*b*d^3*f^2))/f^3 + 64*c*d^2*(c + d*t
an(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*
f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2
*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^
(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3
*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^
2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 4
8*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4
*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*
B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*
f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*
a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*d^2))/f^2)*(-(((8*B^2*a^4*c*f^2 + 8*B
^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f
^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*
B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 1
6*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4
 + d^2*f^4)))^(1/2)*1i)/((((8*(4*B*a^2*c*d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B*
a*b*d^3*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^
2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*
b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b
^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*
f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(
c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2
*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4
 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a
^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 1
6*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) +
 (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*
d^2))/f^2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32
*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B
^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) -
4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2
 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(B^3*a^6*d^2
 - B^3*b^6*d^2 - B^3*a^2*b^4*d^2 + B^3*a^4*b^2*d^2))/f^3 + (((8*(4*B*a^2*c*
d^2*f^2 - 4*B*b^2*c*d^2*f^2 + 8*B*a*b*d^3*f^2))/f^3 + 64*c*d^2*(c + d*tan(e
 + f*x))^(1/2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2
+ 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4
)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2
) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d
*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*B^2*a^
```

```
4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^
2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4
*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*
b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)
/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(B^2*a^4*
d^2 + B^2*b^4*d^2 - 6*B^2*a^2*b^2*d^2))/f^2)*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b
^4*c*f^2 - 32*B^2*a*b^3*d*f^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^
2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*
a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^
2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d
^2*f^4)))^(1/2)))*(-(((8*B^2*a^4*c*f^2 + 8*B^2*b^4*c*f^2 - 32*B^2*a*b^3*d*f
^2 + 32*B^2*a^3*b*d*f^2 - 48*B^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*
f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(
1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*
b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i + atan((
(((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 - 64*c*d
^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C
^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f
^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4
*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 +
 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)
)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*
b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 +
C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4
*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2
*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^
(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d^2))/f^2)*(-(((8*C^2*a^4*
c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*
a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a
^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^
4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(
16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^
2 + 4*C*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^
2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 4
8*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4
*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*
C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*
f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^
2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (
16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4
 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3
*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)
))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^
2*a^2*b^2*d^2))/f^2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*
d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d
```

```
^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)
)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a
^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((
16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*C*a*b*c*d^2*f^2))/f^3 - 64*c*d^2*
(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*
a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4
+ 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^
6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16
*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(
-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d
*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4
*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*
f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^
2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/
2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*b^2*d^2))/f^2)*(-(((8*C^2*a^4*c*f
^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2
*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*
b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c
*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*
(c^2*f^4 + d^2*f^4)))^(1/2) + (((16*(2*C*b^2*d^3*f^2 - 2*C*a^2*d^3*f^2 + 4*
C*a*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*
c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*
a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a
^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^
4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(
16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32
*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2
*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C
^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2
 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/
2) + (16*(c + d*tan(e + f*x))^(1/2)*(C^2*a^4*d^2 + C^2*b^4*d^2 - 6*C^2*a^2*
b^2*d^2))/f^2)*(-(((8*C^2*a^4*c*f^2 + 8*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2
+ 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4
)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2
) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d
*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (32*(2*C^3*a
^3*b^3*d^2 + C^3*a*b^5*d^2 + C^3*a^5*b*d^2))/f^3))*(-(((8*C^2*a^4*c*f^2 + 8
*C^2*b^4*c*f^2 - 32*C^2*a*b^3*d*f^2 + 32*C^2*a^3*b*d*f^2 - 48*C^2*a^2*b^2*c
*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 +
6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 -
 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)/(16*(c^2*f
^4 + d^2*f^4)))^(1/2)*2i + (2*A*b^2*(c + d*tan(e + f*x))^(1/2))/(d*f) + (2*
B*b^2*(c + d*tan(e + f*x))^(3/2))/(3*d^2*f) + (2*C*b^2*(c + d*tan(e + f*x))
^(5/2))/(5*d^3*f)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(e+fx\right)\right)^2\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)}{\sqrt{c+d\tan\left(e+fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(1/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/s
qrt(c + d*tan(e + f*x)), x)
```

3.112 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=194

$$-\frac{(b+ia)(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f\sqrt{c-id}}+\frac{(-b+ia)(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f\sqrt{c+id}}-\frac{2(-3aCd-3bBd+$$

```
[Out] -(I*a+b)*(A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f/(c-I*d)^
(1/2)+(I*a-b)*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/f/(c+
I*d)^(1/2)-2/3*(-3*B*b*d-3*C*a*d+2*C*b*c)*(c+d*tan(f*x+e))^(1/2)/d^2/f+2/3*
b*C*(c+d*tan(f*x+e))^(1/2)*tan(f*x+e)/d/f
```

Rubi [A] time = 0.50, antiderivative size = 194, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3637, 3630, 3539, 3537, 63, 208}

$$-\frac{(b+ia)(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f\sqrt{c-id}}+\frac{(-b+ia)(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f\sqrt{c+id}}-\frac{2(-3aCd-3bBd+$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sqrt[c +
 d*Tan[e + f*x]],x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])
/(Sqrt[c - I*d]*f)) + ((I*a - b)*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f
*x]]/Sqrt[c + I*d]])/(Sqrt[c + I*d]*f) - (2*(2*b*c*C - 3*b*B*d - 3*a*C*d)*S
qrt[c + d*Tan[e + f*x]])/(3*d^2*f) + (2*b*C*Tan[e + f*x]*Sqrt[c + d*Tan[e +
 f*x]])/(3*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{2bC\tan(e+fx)\sqrt{c+d\tan(e+fx)}}{3df} - \frac{2\int \frac{\frac{1}{2}(2b}{}}{}$$

$$= -\frac{2(2bcC-3bBd-3aCd)\sqrt{c+d\tan(e+fx)}}{3d^2f} -$$

$$= -\frac{2(2bcC-3bBd-3aCd)\sqrt{c+d\tan(e+fx)}}{3d^2f} -$$

$$= -\frac{2(2bcC-3bBd-3aCd)\sqrt{c+d\tan(e+fx)}}{3d^2f} -$$

$$= -\frac{2(2bcC-3bBd-3aCd)\sqrt{c+d\tan(e+fx)}}{3d^2f} -$$

$$= -\frac{(ia+b)(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}\,f} -$$

Mathematica [A] time = 1.46, size = 192, normalized size = 0.99

$$\frac{2\left(-\frac{3id(a-ib)(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{2\sqrt{c-id}}+\frac{3id(a+ib)(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{2\sqrt{c+id}}+\frac{(3aCd+3bBd-2bcC)\sqrt{c+d\tan(e+fx)}}{d}+bC\,\mathrm{ta}\right.}{3df}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sq
rt[c + d*Tan[e + f*x]],x]
```

```
[Out] (2*((((-3*I)/2)*(a - I*b)*(A - I*B - C)*d*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/
Sqrt[c - I*d]])/Sqrt[c - I*d] + (((3*I)/2)*(a + I*b)*(A + I*B - C)*d*ArcTan
h[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/Sqrt[c + I*d] + ((-2*b*c*C + 3*b
*B*d + 3*a*C*d)*Sqrt[c + d*Tan[e + f*x]])/d + b*C*Tan[e + f*x]*Sqrt[c + d*T
an[e + f*x]]))/(3*d*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.40, size = 4138, normalized size = 21.33

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)
,x)
```

```
[Out] 1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1
/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a+1/4/f/d*ln((c+d*tan(
f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))
*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e
))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)*a-1/4/f/d*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2
)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b-2/f/d
^2*C*b*c*(c+d*tan(f*x+e))^(1/2)+1/4/f/(c^2+d^2)^(1/2)*ln((c+d*tan(f*x+e))^(
1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^
2+d^2)^(1/2)+2*c)^(1/2)*b+1/4/f/(c^2+d^2)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(
f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)
^(1/2)+2*c)^(1/2)*b+1/4/f/(c^2+d^2)^(1/2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2
+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)
+2*c)^(1/2)*a-1/4/f/(c^2+d^2)^(1/2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^
(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^
(1/2)*b-1/4/f/d/(c^2+d^2)^(1/2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2
)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2
)*a*c+1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d
^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2
))*C*a*c^2-1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c
```

```
+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)
^(1/2))*C*a*c^2+1/4/f/d/(c^2+d^2)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^
(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2
*c)^(1/2)*b*c-1/4/f/d/(c^2+d^2)^(1/2)*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2
)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*B*(2*(c^2+d^2)^(1/2)+2*c
)^(1/2)*b*c-1/4/f/d/(c^2+d^2)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan(f*x+e))^(1/2
)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^
(1/2)*a*c-1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+
d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^
(1/2))*B*b*c^2+1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((
(2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-
2*c)^(1/2))*B*b*c^2+1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arc
tan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(
1/2)-2*c)^(1/2))*A*a*c^2+1/4/f/d/(c^2+d^2)^(1/2)*ln(d*tan(f*x+e)+c+(c+d*tan
(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*C*(2*(c^2+d^2
)^(1/2)+2*c)^(1/2)*a*c+1/4/f/d/(c^2+d^2)^(1/2)*ln((c+d*tan(f*x+e))^(1/2)*(2
*(c^2+d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^
(1/2)+2*c)^(1/2)*a*c-1/f/d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*ar
ctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^
(1/2)-2*c)^(1/2))*A*a*c^2-1/f/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)
*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^
2)^(1/2)-2*c)^(1/2))*A*b*c+1/f/d*(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1
/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2
+d^2)^(1/2)-2*c)^(1/2))*B*b-1/f/d*(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(
1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^
2+d^2)^(1/2)-2*c)^(1/2))*C*a-2/f*d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^
(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c
^2+d^2)^(1/2)-2*c)^(1/2))*a*A+2/f/d*C*(c+d*tan(f*x+e))^(1/2)*a+2/f*d/(c^2+d
^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/
2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b-1/f/d*(c^2+
d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1
/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b-1/f/d*(c^2
+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+
(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a+1/f/d*(c^
2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^
(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*a+1/f/(c^2
+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(
1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a*c-1/f/(c^
2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^
(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*b*c+1/f/(c
^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)
^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*b*c-1/f/(
c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/
2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a*c+1/f/
(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1
```

```
/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*b*c+2/f
*d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))
^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*a*A-2/
f*d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e)
)^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*b-2
/f*d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e
))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*a*C+
1/f/d*(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+
e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*a
+2/f*d/(c^2+d^2)^(1/2)/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(
1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*a*
C+2/f/d*B*(c+d*tan(f*x+e))^(1/2)*b+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan
(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2
)-2*c)^(1/2))*C*b+2/3/f/d^2*C*b*(c+d*tan(f*x+e))^(3/2)-1/f/(2*(c^2+d^2)^(1/
2)-2*c)^(1/2)*arctan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2
))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*C*b+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arc
tan((2*(c+d*tan(f*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(
1/2)-2*c)^(1/2))*A*b+1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan((2*(c+d*tan(f
*x+e))^(1/2)+(2*(c^2+d^2)^(1/2)+2*c)^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*
B*a-1/f/(2*(c^2+d^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)
-2*(c+d*tan(f*x+e))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*A*b-1/f/(2*(c^2+d
^2)^(1/2)-2*c)^(1/2)*arctan(((2*(c^2+d^2)^(1/2)+2*c)^(1/2)-2*(c+d*tan(f*x+e
))^(1/2))/(2*(c^2+d^2)^(1/2)-2*c)^(1/2))*B*a-1/4/f/(c^2+d^2)^(1/2)*ln(d*tan
(f*x+e)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1
/2))*A*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*b-1/4/f/(c^2+d^2)^(1/2)*ln(d*tan(f*x+e
)+c+(c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)+(c^2+d^2)^(1/2))*B
*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a-1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+
d^2)^(1/2)+2*c)^(1/2)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*A*(2*(c^2+d^2)^(1/2)+
2*c)^(1/2)*a+1/4/f/d*ln((c+d*tan(f*x+e))^(1/2)*(2*(c^2+d^2)^(1/2)+2*c)^(1/2
)-d*tan(f*x+e)-c-(c^2+d^2)^(1/2))*C*(2*(c^2+d^2)^(1/2)+2*c)^(1/2)*a
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [B] time = 23.48, size = 16400, normalized size = 84.54

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x))^(1/2),x)

[Out] ((2*B*b*d - 6*C*b*c)/(d^2*f) + (4*C*b*c)/(d^2*f))*(c + d*tan(e + f*x))^(1/2
) - atan(((((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 - 64*
c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*
C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4
 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4
*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4)
)^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d
*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*
((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2
- 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4
*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^
2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2
*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^
2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(
A^2*a^2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*((((8*A^2*a^
2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2
*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a
^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 +
2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 -
 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16
*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B
*a*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^
2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2
 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C
^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*
C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2
*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*
f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c
*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^
2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^
4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) -
 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*
A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c +
 d*tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2
*d^2))/f^2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*
B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d
^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^
2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c
*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^
2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*C*a*d^3*f
^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^
(1/2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*
```

```
d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4
)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4
+ 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 +
 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*
B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*a^2*c*f^2 - 8*B^2
*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C
*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 -
4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 -
 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^
2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2
*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 +
 C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 +
 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^
2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4
- 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a
^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^
2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2
) + (((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 + 64*c*d^2*
(c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^
2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16
*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C
*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2
) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 +
 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*
A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A
*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 +
 B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*
a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c
*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^
2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^
2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*((((8*A^2*a^2*c*f^
2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2
 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C
^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*
C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2
*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*
f^4 + d^2*f^4)))^(1/2) - (16*(B^3*a^3*d^2 + A^2*B*a^3*d^2 + B*C^2*a^3*d^2 -
 2*A*B*C*a^3*d^2))/f^3))*((((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*
c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c
^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a
^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2)
- 4*A^2*a^2*c*f^2 + 4*B^2*a^2*c*f^2 - 4*C^2*a^2*c*f^2 - 8*A*B*a^2*d*f^2 + 8
*A*C*a^2*c*f^2 + 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - atan
(((((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 - 64*c*d^2*(c
 + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2
```

*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C *a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (((8*(4*C*a*d^3*f^2 - 4*A*a*d^3*f^2 + 4*B*a*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*a^2*d^2 - B^2*a^2*d^2 + C^2*a^2*d^2 - 2*A*C*a^2*d^2))/f^2)*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(B^3*a^3*d^2 + A^2*B*a^3*d^2 + B*C^2*a^3*d^2 - 2*A*B*C*a^3*d^2))/f^3))*(-(((8*A^2*a^2*c*f^2 - 8*B^2*a^2*c*f^2 + 8*C^2*a^2*c*f^2 + 16*A*B*a^2*d*f^2 - 16*A*C*a^2*c*f^2 - 16*B*C*a^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c*f^2 - 4*B^2*a^2*c*f^2 + 4*C^2*a^2*c*f^2 + 8*A*B*a^2*d*f^2 - 8*A*C*a^2*c*f^2 - 8*B*C*a^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - atan(((((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2

```
 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^
4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A
^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^
2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b
^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*
(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*(-(((8*A^2*
b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b
^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4
*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4
+ 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2
 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(
16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i - (((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 -
 4*C*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2
*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*
c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^
4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2
*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 -
4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*
(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2
*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 -
(16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^
3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(
1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^
2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (1
6*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A
*C*b^2*d^2))/f^2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2
+ 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4
 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2
*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^
2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b
^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*B*
b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e
 + f*x))^(1/2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 1
6*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 +
16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^
2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b
^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*
c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*b^2*c*
f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f
^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 +
 C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^
2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C
^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^
2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^2 - B^
2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2
```

```
*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C
*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 -
4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 -
 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^
2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2
*f^4)))^(1/2) - (16*(A^3*b^3*d^2 - C^3*b^3*d^2 + A*B^2*b^3*d^2 + 3*A*C^2*b^
3*d^2 - 3*A^2*C*b^3*d^2 - B^2*C*b^3*d^2))/f^3 + (((8*(4*B*b*d^3*f^2 + 4*A*b
*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 + 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*(-
(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 -
 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*
b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2
*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*
b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2
*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c
*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d
*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^
3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B
^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*
A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4))
)^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b
^2*d^2 - 2*A*C*b^2*d^2))/f^2)*(-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^
2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 -
 (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A
^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^
(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f
^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)))*(
-(((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2
- 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4
*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^
2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c*f^2 + 4*B^2
*b^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*A*B*b^2*d*f^2 + 8*A*C*b^2*c*f^2 + 8*B*C*b^
2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i - atan(((((8*(4*B*b*d^3*f^2 + 4
*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2
)*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^
2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A
^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*
A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B
^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*
b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2
*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2
*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*
C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A
*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 +
8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4
)))^(1/2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2
```

```
*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C
^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4
- (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*
A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))
^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*
f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*1i
 - (((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 + 64*c*d^2
*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b
^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (1
6*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*
C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/
2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2
- 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))*((((8
*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*
A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4
+ B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2
*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*
c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f
^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^2*b
^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*((((8*A^2*b^2*c*f
^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^
2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 +
C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2
*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^
2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2
*f^4 + d^2*f^4)))^(1/2)*1i)/((((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*
c*d^2*f^2))/f^3 - 64*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c*f^2 -
 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 -
16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*
b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2
*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^
2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4
 + d^2*f^4)))^(1/2))*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^
2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f
^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 +
 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*
A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C
*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) + (16*(c + d*
tan(e + f*x))^(1/2)*(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^
2))/f^2)*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b
^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*
f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b
^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^
2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 -
 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(A^3*b^3*d^2 - C^3*
```

```
b^3*d^2 + A*B^2*b^3*d^2 + 3*A*C^2*b^3*d^2 - 3*A^2*C*b^3*d^2 - B^2*C*b^3*d^2
))/f^3 + (((8*(4*B*b*d^3*f^2 + 4*A*b*c*d^2*f^2 - 4*C*b*c*d^2*f^2))/f^3 + 64
*c*d^2*(c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8
*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/
4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 -
4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4
))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*
d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2))
*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2
 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^
4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A
^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^
2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b
^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2) - (16*(c + d*tan(e + f*x))^(1/2)*
(A^2*b^2*d^2 - B^2*b^2*d^2 + C^2*b^2*d^2 - 2*A*C*b^2*d^2))/f^2)*((((8*A^2*b
^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^
2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4 - (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*
b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 +
 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2
+ 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(1
6*(c^2*f^4 + d^2*f^4)))^(1/2)))*((((8*A^2*b^2*c*f^2 - 8*B^2*b^2*c*f^2 + 8*C
^2*b^2*c*f^2 + 16*A*B*b^2*d*f^2 - 16*A*C*b^2*c*f^2 - 16*B*C*b^2*d*f^2)^2/4
- (16*c^2*f^4 + 16*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*
A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))
^(1/2) + 4*A^2*b^2*c*f^2 - 4*B^2*b^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*A*B*b^2*d*
f^2 - 8*A*C*b^2*c*f^2 - 8*B*C*b^2*d*f^2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*2i
 + (2*C*a*(c + d*tan(e + f*x))^(1/2))/(d*f) + (2*C*b*(c + d*tan(e + f*x))^(
3/2))/(3*d^2*f)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(e+fx\right)\right)\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)}{\sqrt{c+d\tan\left(e+fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
)**(1/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/sqrt
(c + d*tan(e + f*x)), x)
```

3.113 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=133

$$-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f\sqrt{c-id}}-\frac{(B-i(A-C))\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f\sqrt{c+id}}+\frac{2C\sqrt{c+d\tan(e+fx)}}{df}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/f/(c-I*d)^(1/2)-
(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/f/(c+I*d)^(1/2)+2
*C*(c+d*tan(f*x+e))^(1/2)/d/f
```

Rubi [A] time = 0.22, antiderivative size = 133, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 5, integrand size = 35, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.143, Rules used = {3630, 3539, 3537, 63, 208}

$$-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f\sqrt{c-id}}-\frac{(B-i(A-C))\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f\sqrt{c+id}}+\frac{2C\sqrt{c+d\tan(e+fx)}}{df}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/(Sqrt[c
 - I*d]*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*
d]])/(Sqrt[c + I*d]*f) + (2*C*Sqrt[c + d*Tan[e + f*x]])/(d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
```

```
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{2C\sqrt{c+d\tan(e+fx)}}{df}+\int \frac{A-C+B\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{df}+\frac{1}{2}(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{df}+\frac{(iA+B-iC)\,\text{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c-idx}}\,dx, x\right.}{2f}$$

$$= \frac{2C\sqrt{c+d\tan(e+fx)}}{df}-\frac{(A-iB-C)\,\text{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\,dx, x, \sqrt{c\right.}}{df}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}\,f}-\frac{(B-i(A-C))\tanh^{-1}\left(\right.}{\sqrt{c+id}\,f}$$

Mathematica [A] time = 0.22, size = 129, normalized size = 0.97

$$\frac{-\frac{i(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}}+\frac{i(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}}+\frac{2C\sqrt{c+d\tan(e+fx)}}{d}}{f}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/Sqrt[c + d*Tan[e + f*x]],
x]
```

```
[Out] (((-I)*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/Sqrt[
c - I*d] + (I*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]
)/Sqrt[c + I*d] + (2*C*Sqrt[c + d*Tan[e + f*x]])/d)/f
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2),x, algorit
hm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2),x, algorit
hm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.37, size = 5570, normalized size = 41.88

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2),x)
```

```
[Out] result too large to display
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A}{\sqrt{d\tan\left(fx+e\right)+c}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2),x, algorit
hm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)/sqrt(d*tan(f*x + e) + c),
 x)
```

mupad [B] time = 14.21, size = 4326, normalized size = 32.53

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x))^(1/2),x)
```

```
[Out] 2*atanh((32*C^2*d^2*((-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (C^
2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*C^
3*c*d^3*f^3)/(c^2*f^4 + d^2*f^4) - (4*C*d^3*f^2*(-16*C^4*d^2*f^4)^(1/2))/(c
^2*f^5 + d^2*f^5)) + (8*c*d^2*((-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f
^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2
)*(-16*C^4*d^2*f^4)^(1/2))/((16*C^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) - (4*C*d
^5*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*C^3*c^3*d^3*f^5)/
(c^2*f^4 + d^2*f^4) - (4*C*c^2*d^3*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 +
d^2*f^5)) - (32*C^2*c^2*d^2*f^2*((-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2
*f^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1
/2))/((16*C^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) - (4*C*d^5*f^4*(-16*C^4*d^2*f^
4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*C^3*c^3*d^3*f^5)/(c^2*f^4 + d^2*f^4) -
(4*C*c^2*d^3*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)))*((-16*C^4*d
^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)
))^(1/2) - 2*atanh((8*c*d^2*(- (-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f
^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2
)*(-16*C^4*d^2*f^4)^(1/2))/((16*C^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) + (4*C*d
^5*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*C^3*c^3*d^3*f^5)/
(c^2*f^4 + d^2*f^4) + (4*C*c^2*d^3*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 +
d^2*f^5)) - (32*C^2*d^2*(- (-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4))
 - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/(
(16*C^3*c*d^3*f^3)/(c^2*f^4 + d^2*f^4) + (4*C*d^3*f^2*(-16*C^4*d^2*f^4)^(1/
2))/(c^2*f^5 + d^2*f^5)) + (32*C^2*c^2*d^2*f^2*(- (-16*C^4*d^2*f^4)^(1/2)/(
16*(c^2*f^4 + d^2*f^4)) - (C^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d
*tan(e + f*x))^(1/2))/((16*C^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4) + (4*C*d^5*f^
4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*C^3*c^3*d^3*f^5)/(c^2*
f^4 + d^2*f^4) + (4*C*c^2*d^3*f^4*(-16*C^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f
^5)))*(- (-16*C^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (C^2*c*f^2)/(4*
(c^2*f^4 + d^2*f^4)))^(1/2) - 2*atanh((32*A^2*d^2*((-16*A^4*d^2*f^4)^(1/2)/
(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c +
d*tan(e + f*x))^(1/2))/((16*A^3*c*d^3*f^3)/(c^2*f^4 + d^2*f^4) - (4*A*d^3*f
^2*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) + (8*c*d^2*((-16*A^4*d^2*f
```

```
^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(
1/2)*(c + d*tan(e + f*x))^(1/2)*(-16*A^4*d^2*f^4)^(1/2))/((16*A^3*c*d^5*f^5
)/(c^2*f^4 + d^2*f^4) - (4*A*d^5*f^4*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^
2*f^5) + (16*A^3*c^3*d^3*f^5)/(c^2*f^4 + d^2*f^4) - (4*A*c^2*d^3*f^4*(-16*A
^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) - (32*A^2*c^2*d^2*f^2*((-16*A^4*d^2
*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))
^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*A^3*c*d^5*f^5)/(c^2*f^4 + d^2*f^4)
- (4*A*d^5*f^4*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (16*A^3*c^3*d
^3*f^5)/(c^2*f^4 + d^2*f^4) - (4*A*c^2*d^3*f^4*(-16*A^4*d^2*f^4)^(1/2))/(c^
2*f^5 + d^2*f^5)))*((-16*A^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2
*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2) + 2*atanh((8*c*d^2*(- (-16*A^4*d^2*f
^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(
1/2)*(c + d*tan(e + f*x))^(1/2)*(-16*A^4*d^2*f^4)^(1/2))/((16*A^3*c*d^5*f^5
)/(c^2*f^4 + d^2*f^4) + (4*A*d^5*f^4*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^
2*f^5) + (16*A^3*c^3*d^3*f^5)/(c^2*f^4 + d^2*f^4) + (4*A*c^2*d^3*f^4*(-16*A
^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) - (32*A^2*d^2*(- (-16*A^4*d^2*f^4)^
(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)
*(c + d*tan(e + f*x))^(1/2))/((16*A^3*c*d^3*f^3)/(c^2*f^4 + d^2*f^4) + (4*A
*d^3*f^2*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) + (32*A^2*c^2*d^2*f^
2*(- (-16*A^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2
*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*A^3*c*d^5*f^5)/(c^
2*f^4 + d^2*f^4) + (4*A*d^5*f^4*(-16*A^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5
) + (16*A^3*c^3*d^3*f^5)/(c^2*f^4 + d^2*f^4) + (4*A*c^2*d^3*f^4*(-16*A^4*d^
2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)))*(- (-16*A^4*d^2*f^4)^(1/2)/(16*(c^2*f^4
 + d^2*f^4)) - (A^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2) - 2*atanh((32*B^2
*d^2*((B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)) - (-16*B^4*d^2*f^4)^(1/2)/(16*(c^
2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*B^3*d^2)/f - (16*
B^3*c^2*d^2*f^3)/(c^2*f^4 + d^2*f^4) + (4*B*c*d^2*f^2*(-16*B^4*d^2*f^4)^(1/
2))/(c^2*f^5 + d^2*f^5)) + (8*c*d^2*((B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)) -
(-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x)
)^(1/2)*(-16*B^4*d^2*f^4)^(1/2))/(16*B^3*d^4*f + 16*B^3*c^2*d^2*f - (16*B^3
*c^2*d^4*f^5)/(c^2*f^4 + d^2*f^4) - (16*B^3*c^4*d^2*f^5)/(c^2*f^4 + d^2*f^4
) + (4*B*c*d^4*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (4*B*c^3*
d^2*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) - (32*B^2*c^2*d^2*f^2
*((B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)) - (-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^
4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/(16*B^3*d^4*f + 16*B^3*c^2
*d^2*f - (16*B^3*c^2*d^4*f^5)/(c^2*f^4 + d^2*f^4) - (16*B^3*c^4*d^2*f^5)/(c
^2*f^4 + d^2*f^4) + (4*B*c*d^4*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*
f^5) + (4*B*c^3*d^2*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)))*((B^
2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)) - (-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d
^2*f^4)))^(1/2) - 2*atanh((8*c*d^2*((-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 +
d^2*f^4)) + (B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))
^(1/2)*(-16*B^4*d^2*f^4)^(1/2))/((16*B^3*c^2*d^4*f^5)/(c^2*f^4 + d^2*f^4) -
 16*B^3*c^2*d^2*f - 16*B^3*d^4*f + (16*B^3*c^4*d^2*f^5)/(c^2*f^4 + d^2*f^4)
 + (4*B*c*d^4*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (4*B*c^3*d
```

```
^2*f^4*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5)) - (32*B^2*d^2*((-16*B^
4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^4)) + (B^2*c*f^2)/(4*(c^2*f^4 + d^2*f
^4)))^(1/2)*(c + d*tan(e + f*x))^(1/2))/((16*B^3*c^2*d^2*f^3)/(c^2*f^4 + d^
2*f^4) - (16*B^3*d^2)/f + (4*B*c*d^2*f^2*(-16*B^4*d^2*f^4)^(1/2))/(c^2*f^5
+ d^2*f^5)) + (32*B^2*c^2*d^2*f^2*((-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d
^2*f^4)) + (B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2)*(c + d*tan(e + f*x))^
(1/2))/((16*B^3*c^2*d^4*f^5)/(c^2*f^4 + d^2*f^4) - 16*B^3*c^2*d^2*f - 16*B^
3*d^4*f + (16*B^3*c^4*d^2*f^5)/(c^2*f^4 + d^2*f^4) + (4*B*c*d^4*f^4*(-16*B^
4*d^2*f^4)^(1/2))/(c^2*f^5 + d^2*f^5) + (4*B*c^3*d^2*f^4*(-16*B^4*d^2*f^4)^
(1/2))/(c^2*f^5 + d^2*f^5)))*((-16*B^4*d^2*f^4)^(1/2)/(16*(c^2*f^4 + d^2*f^
4)) + (B^2*c*f^2)/(4*(c^2*f^4 + d^2*f^4)))^(1/2) + (2*C*(c + d*tan(e + f*x)
)^(1/2))/(d*f)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\sqrt{c + d\tan\left(e + fx\right)}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(1/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/sqrt(c + d*tan(e + f*x)),
 x)
```

3.114 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=210

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\sqrt{b}f\left(a^2+b^2\right)\sqrt{bc-ad}}-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)\sqrt{c-id}}-\frac{(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(-b+ia)\sqrt{c+id}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a-I*b)/f/(c-I*d
)^(1/2)-(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(I*a-b)/f/(
c+I*d)^(1/2)-2*(A*b^2-a*(B*b-C*a))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(
-a*d+b*c)^(1/2))/(a^2+b^2)/f/b^(1/2)/(-a*d+b*c)^(1/2)
```

Rubi [A] time = 0.61, antiderivative size = 210, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 6, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}} = 0.128$, Rules used = {3653, 3539, 3537, 63, 208, 3634}

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\sqrt{b}f\left(a^2+b^2\right)\sqrt{bc-ad}}-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(a-ib)\sqrt{c-id}}-\frac{(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(-b+ia)\sqrt{c+id}}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*Sqrt[c +
d*Tan[e + f*x]]),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((a - I
*b)*Sqrt[c - I*d]*f)) - ((A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqr
t[c + I*d]])/((I*a - b)*Sqrt[c + I*d]*f) - (2*(A*b^2 - a*(b*B - a*C))*ArcTa
nh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(Sqrt[b]*(a^2 + b^2
)*Sqrt[b*c - a*d]*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}\,dx = \frac{\int \frac{bB+a(A-C)-(Ab-aB-bC)\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{a^2+b^2} + \frac{\left(Ab^2-abB+a^2C\right)\int \frac{\cdots}{(a+b\tan\cdots}}{a^2+b^2\cdots}$$

$$= \frac{(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(a-ib)} + \frac{(A+iB-C)\int \frac{1-i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,d\cdots}{2(a+ib)}$$

$$= -\frac{(i(A+iB-C))\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{c+idx}}\,dx, x, -i\tan(e+fx)\right)}{2(a+ib)f} + -\frac{(i\cdots}{\cdots}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\sqrt{b}\left(a^2+b^2\right)\sqrt{bc-ad}\,f} - \frac{(A-iB-C)\cdots}{\cdots}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)\sqrt{c-id}\,f} - \frac{(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{\cdots}}{\cdots}\right)}{(ia-b)\sqrt{c+i\cdots}}$$

Mathematica [A] time = 0.42, size = 194, normalized size = 0.92

$$\frac{-\frac{2\left(a(aC-bB)+Ab^2\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\sqrt{b}\sqrt{bc-ad}} + \frac{(b-ia)(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}} + \frac{(b+ia)(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}}}{f\left(a^2+b^2\right)}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*Sqr
t[c + d*Tan[e + f*x]]),x]
```

```
[Out] ((((-I)*a + b)*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]
])/Sqrt[c - I*d] + ((I*a + b)*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]
]/Sqrt[c + I*d]])/Sqrt[c + I*d] - (2*(A*b^2 + a*(-(b*B) + a*C))*ArcTanh[(Sq
rt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(Sqrt[b]*Sqrt[b*c - a*d])
)/((a^2 + b^2)*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e)),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e)),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [**B**] time = 0.52, size = 13474, normalized size = 64.16

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
,x)
```

```
[Out] result too large to display
```

maxima [**F(-2)**] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e)),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [**B**] time = 69.14, size = 25341, normalized size = 120.67

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))^(1/2)),x)
```

```
[Out] (log(((((((((128*C*b^2*d^8*(a*d + b*c)^2*(a^2 + b^2)^2)/f - 64*b^2*d^8*(a^
2 + b^2)^2*(c + d*tan(e + f*x))^(1/2)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*
c)^2)^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^
2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b
*d^2 + a^3*c*d + a*b^2*c*d))*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/
2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^
2*(c^2 + d^2)))^(1/2))/4 + (64*C^2*b*d^8*(c + d*tan(e + f*x))^(1/2)*(5*b^6*
c - 4*a^6*c - 2*a^2*b^4*c + 5*a^4*b^2*c - 2*a^3*b^3*d + 7*a*b^5*d + 7*a^5*b
*d))/f^2)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*C^2*a^2*c*f^
2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/
2))/4 + (32*C^3*b*d^8*(4*a^5*d - b^5*c - 9*a^2*b^3*c - 15*a^3*b^2*d + 12*a^
4*b*c + a*b^4*d))/f^3)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4
*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2
 + d^2)))^(1/2))/4 - (32*C^4*b*d^8*(2*a^4 + b^4)*(c + d*tan(e + f*x))^(1/2)
)/f^4)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*C^2*a^2*c*f^2 +
 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))
/4 + (32*C^5*a^2*b^2*d^8)/f^5)*(((32*C^4*a^2*b^2*d^2*f^4 - 16*C^4*b^4*d^2*f
^4 - 64*C^4*a^2*b^2*c^2*f^4 - 16*C^4*a^4*d^2*f^4 + 64*C^4*a*b^3*c*d*f^4 - 6
4*C^4*a^3*b*c*d*f^4)^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*
d*f^2)/(a^4*c^2*f^4 + a^4*d^2*f^4 + b^4*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*c
^2*f^4 + 2*a^2*b^2*d^2*f^4))^(1/2))/4 + (log(((((((((128*C*b^2*d^8*(a*d +
b*c)^2*(a^2 + b^2)^2)/f - 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/
2)*(-(4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*
C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*
b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*(-(4*
(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*
c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (64*C^
2*b*d^8*(c + d*tan(e + f*x))^(1/2)*(5*b^6*c - 4*a^6*c - 2*a^2*b^4*c + 5*a^4
*b^2*c - 2*a^3*b^3*d + 7*a*b^5*d + 7*a^5*b*d))/f^2)*(-(4*(-C^4*f^4*(a^2*d -
 b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*
d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*C^3*b*d^8*(4*a^5*d -
 b^5*c - 9*a^2*b^3*c - 15*a^3*b^2*d + 12*a^4*b*c + a*b^4*d))/f^3)*(-(4*(-C^
4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^
2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (32*C^4*b*
d^8*(2*a^4 + b^4)*(c + d*tan(e + f*x))^(1/2))/f^4)*(-(4*(-C^4*f^4*(a^2*d -
b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d
*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*C^5*a^2*b^2*d^8)/f^5)
*(-((32*C^4*a^2*b^2*d^2*f^4 - 16*C^4*b^4*d^2*f^4 - 64*C^4*a^2*b^2*c^2*f^4 -
 16*C^4*a^4*d^2*f^4 + 64*C^4*a*b^3*c*d*f^4 - 64*C^4*a^3*b*c*d*f^4)^(1/2) +
4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(a^4*c^2*f^4 + a^4*d^2
*f^4 + b^4*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*c^2*f^4 + 2*a^2*b^2*d^2*f^4))^
(1/2))/4 - log(((((((((128*C*b^2*d^8*(a*d + b*c)^2*(a^2 + b^2)^2)/f + 64*b
^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/2)*((4*(-C^4*f^4*(a^2*d - b^2*
d + 2*a*b*c)^2)^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2
)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2
```

```
 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b
*c)^2)^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a
^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (64*C^2*b*d^8*(c + d*tan(e + f*x))^(1/
2)*(5*b^6*c - 4*a^6*c - 2*a^2*b^4*c + 5*a^4*b^2*c - 2*a^3*b^3*d + 7*a*b^5*d
 + 7*a^5*b*d))/f^2)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*C^
2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 +
d^2)))^(1/2))/4 + (32*C^3*b*d^8*(4*a^5*d - b^5*c - 9*a^2*b^3*c - 15*a^3*b^2
*d + 12*a^4*b*c + a*b^4*d))/f^3)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)
^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b
^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*C^4*b*d^8*(2*a^4 + b^4)*(c + d*tan(e + f
*x))^(1/2))/f^4)*((4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*C^2*a
^2*c*f^2 + 4*C^2*b^2*c*f^2 + 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2
)))^(1/2))/4 + (32*C^5*a^2*b^2*d^8)/f^5)*(((32*C^4*a^2*b^2*d^2*f^4 - 16*C^4
*b^4*d^2*f^4 - 64*C^4*a^2*b^2*c^2*f^4 - 16*C^4*a^4*d^2*f^4 + 64*C^4*a*b^3*c
*d*f^4 - 64*C^4*a^3*b*c*d*f^4)^(1/2) - 4*C^2*a^2*c*f^2 + 4*C^2*b^2*c*f^2 +
8*C^2*a*b*d*f^2)/(16*a^4*c^2*f^4 + 16*a^4*d^2*f^4 + 16*b^4*c^2*f^4 + 16*b^4
*d^2*f^4 + 32*a^2*b^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4))^(1/2) - log(((((((((1
28*C*b^2*d^8*(a*d + b*c)^2*(a^2 + b^2)^2)/f + 64*b^2*d^8*(a^2 + b^2)^2*(c +
 d*tan(e + f*x))^(1/2)*(-(4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) +
4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^
2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d
 + a*b^2*c*d))*(-(4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^
2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)
))^(1/2))/4 - (64*C^2*b*d^8*(c + d*tan(e + f*x))^(1/2)*(5*b^6*c - 4*a^6*c -
 2*a^2*b^4*c + 5*a^4*b^2*c - 2*a^3*b^3*d + 7*a*b^5*d + 7*a^5*b*d))/f^2)*(-(
4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^
2*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*
C^3*b*d^8*(4*a^5*d - b^5*c - 9*a^2*b^3*c - 15*a^3*b^2*d + 12*a^4*b*c + a*b^
4*d))/f^3)*(-(4*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*
f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(
1/2))/4 + (32*C^4*b*d^8*(2*a^4 + b^4)*(c + d*tan(e + f*x))^(1/2))/f^4)*(-(4
*(-C^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2
*c*f^2 - 8*C^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*C
^5*a^2*b^2*d^8)/f^5)*(-((32*C^4*a^2*b^2*d^2*f^4 - 16*C^4*b^4*d^2*f^4 - 64*C
^4*a^2*b^2*c^2*f^4 - 16*C^4*a^4*d^2*f^4 + 64*C^4*a*b^3*c*d*f^4 - 64*C^4*a^3
*b*c*d*f^4)^(1/2) + 4*C^2*a^2*c*f^2 - 4*C^2*b^2*c*f^2 - 8*C^2*a*b*d*f^2)/(1
6*a^4*c^2*f^4 + 16*a^4*d^2*f^4 + 16*b^4*c^2*f^4 + 16*b^4*d^2*f^4 + 32*a^2*b
^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4))^(1/2) + (log(- (((((((((128*B*b^2*d^8*(a^
2 + b^2)^2*(a*b*c^2 + 3*a*b*d^2 - a^2*c*d + b^2*c*d))/f + 64*b^2*d^8*(a^2 +
 b^2)^2*(c + d*tan(e + f*x))^(1/2)*((4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^
2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 +
 b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^
2 + a^3*c*d + a*b^2*c*d))*((4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2)
+ 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(
c^2 + d^2)))^(1/2))/4 - (64*B^2*b^2*d^8*(c + d*tan(e + f*x))^(1/2)*(a^5*d -
```

5*b^5*c + 6*a^2*b^3*c + 10*a^3*b^2*d - 5*a^4*b*c - 7*a*b^4*d))/f^2)*((4*(-
B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*
f^2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*B^3*
a*b^2*d^8*(a^3*d + 7*b^3*c - 5*a^2*b*c + 13*a*b^2*d))/f^3)*((4*(-B^4*f^4*(a
^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^
2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (32*B^4*b^3*d^8*(2
a^2 - b^2)(c + d*tan(e + f*x))^(1/2))/f^4)*((4*(-B^4*f^4*(a^2*d - b^2*d +
2*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(
f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (32*B^5*a*b^3*d^8)/f^5)*(((32*B^
4*a^2*b^2*d^2*f^4 - 16*B^4*b^4*d^2*f^4 - 64*B^4*a^2*b^2*c^2*f^4 - 16*B^4*a^
4*d^2*f^4 + 64*B^4*a*b^3*c*d*f^4 - 64*B^4*a^3*b*c*d*f^4)^(1/2) + 4*B^2*a^2*
c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(a^4*c^2*f^4 + a^4*d^2*f^4 + b^4
*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*c^2*f^4 + 2*a^2*b^2*d^2*f^4))^(1/2))/4 +
(log(- ((((((((128*B*b^2*d^8*(a^2 + b^2)^2*(a*b*c^2 + 3*a*b*d^2 - a^2*c*d
+ b^2*c*d))/f + 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/2)*(-(4*(
-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c
*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 +
2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*(-(4*(-B^4*f^4
*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8
*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (64*B^2*b^2*d^8
*(c + d*tan(e + f*x))^(1/2)*(a^5*d - 5*b^5*c + 6*a^2*b^3*c + 10*a^3*b^2*d -
5*a^4*b*c - 7*a*b^4*d))/f^2)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(
1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2
)^2*(c^2 + d^2)))^(1/2))/4 + (32*B^3*a*b^2*d^8*(a^3*d + 7*b^3*c - 5*a^2*b*c
+ 13*a*b^2*d))/f^3)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*
B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2
+ d^2)))^(1/2))/4 - (32*B^4*b^3*d^8*(2*a^2 - b^2)*(c + d*tan(e + f*x))^(1/2
))/f^4)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a^2*c*f^2
+ 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2
))/4 - (32*B^5*a*b^3*d^8)/f^5)*(-((32*B^4*a^2*b^2*d^2*f^4 - 16*B^4*b^4*d^2*
f^4 - 64*B^4*a^2*b^2*c^2*f^4 - 16*B^4*a^4*d^2*f^4 + 64*B^4*a*b^3*c*d*f^4 -
64*B^4*a^3*b*c*d*f^4)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b
*d*f^2)/(a^4*c^2*f^4 + a^4*d^2*f^4 + b^4*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*
c^2*f^4 + 2*a^2*b^2*d^2*f^4))^(1/2))/4 - log(- ((((((((128*B*b^2*d^8*(a^2
+ b^2)^2*(a*b*c^2 + 3*a*b*d^2 - a^2*c*d + b^2*c*d))/f - 64*b^2*d^8*(a^2 + b
^2)^2*(c + d*tan(e + f*x))^(1/2)*((4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)
^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b
^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2
+ a^3*c*d + a*b^2*c*d))*((4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) +
4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^
2 + d^2)))^(1/2))/4 + (64*B^2*b^2*d^8*(c + d*tan(e + f*x))^(1/2)*(a^5*d - 5
*b^5*c + 6*a^2*b^3*c + 10*a^3*b^2*d - 5*a^4*b*c - 7*a*b^4*d))/f^2)*((4*(-B^
4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^
2 - 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*B^3*a*
b^2*d^8*(a^3*d + 7*b^3*c - 5*a^2*b*c + 13*a*b^2*d))/f^3)*((4*(-B^4*f^4*(a^2

*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*
a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*B^4*b^3*d^8*(2*a
^2 - b^2)*(c + d*tan(e + f*x))^(1/2))/f^4)*((4*(-B^4*f^4*(a^2*d - b^2*d + 2
*a*b*c)^2)^(1/2) + 4*B^2*a^2*c*f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(f^
4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (32*B^5*a*b^3*d^8)/f^5)*(((32*B^4*
a^2*b^2*d^2*f^4 - 16*B^4*b^4*d^2*f^4 - 64*B^4*a^2*b^2*c^2*f^4 - 16*B^4*a^4*
d^2*f^4 + 64*B^4*a*b^3*c*d*f^4 - 64*B^4*a^3*b*c*d*f^4)^(1/2) + 4*B^2*a^2*c*
f^2 - 4*B^2*b^2*c*f^2 - 8*B^2*a*b*d*f^2)/(16*a^4*c^2*f^4 + 16*a^4*d^2*f^4 +
 16*b^4*c^2*f^4 + 16*b^4*d^2*f^4 + 32*a^2*b^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4)
)^(1/2) - log(- (((((((((128*B*b^2*d^8*(a^2 + b^2)^2*(a*b*c^2 + 3*a*b*d^2 -
 a^2*c*d + b^2*c*d))/f - 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/2
)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B
^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b
^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*(-(4*(
-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c
*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (64*B^2
*b^2*d^8*(c + d*tan(e + f*x))^(1/2)*(a^5*d - 5*b^5*c + 6*a^2*b^3*c + 10*a^3
*b^2*d - 5*a^4*b*c - 7*a*b^4*d))/f^2)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b
*c)^2)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a
^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*B^3*a*b^2*d^8*(a^3*d + 7*b^3*c - 5
*a^2*b*c + 13*a*b^2*d))/f^3)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1
/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)
^2*(c^2 + d^2)))^(1/2))/4 + (32*B^4*b^3*d^8*(2*a^2 - b^2)*(c + d*tan(e + f*
x))^(1/2))/f^4)*(-(4*(-B^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*B^2*a
^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8*B^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2
)))^(1/2))/4 - (32*B^5*a*b^3*d^8)/f^5)*(-((32*B^4*a^2*b^2*d^2*f^4 - 16*B^4*
b^4*d^2*f^4 - 64*B^4*a^2*b^2*c^2*f^4 - 16*B^4*a^4*d^2*f^4 + 64*B^4*a*b^3*c*
d*f^4 - 64*B^4*a^3*b*c*d*f^4)^(1/2) - 4*B^2*a^2*c*f^2 + 4*B^2*b^2*c*f^2 + 8
*B^2*a*b*d*f^2)/(16*a^4*c^2*f^4 + 16*a^4*d^2*f^4 + 16*b^4*c^2*f^4 + 16*b^4*
d^2*f^4 + 32*a^2*b^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4))^(1/2) + (log(((((((((128
*A*b^2*d^8*(a^2 + b^2)^2*(a^2*d^2 - 3*b^2*c^2 - 4*b^2*d^2 + 2*a*b*c*d))/f +
 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/2)*((4*(-A^4*f^4*(a^2*d -
 b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*
d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*
b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*((4*(-A^4*f^4*(a^2*d - b^2*d +
2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(f
^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (64*A^2*b^2*d^8*(a^2 - 3*b^2)*(c
+ d*tan(e + f*x))^(1/2)*(a^3*d + 3*b^3*c - a^2*b*c + 5*a*b^2*d))/f^2)*((4*(
-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c
*f^2 + 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*A^3
*b^3*d^8*(a^3*d + 3*b^3*c - a^2*b*c + 5*a*b^2*d))/f^3)*((4*(-A^4*f^4*(a^2*d
 - b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*
b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (96*A^4*b^5*d^8*(c + d
*tan(e + f*x))^(1/2))/f^4)*(((32*A^4*a^2*b^2*d^2*f^4 - 16*A^4*b^4*d^2*f^4 -
 64*A^4*a^2*b^2*c^2*f^4 - 16*A^4*a^4*d^2*f^4 + 64*A^4*a*b^3*c*d*f^4 - 64*A^

```
4*a^3*b*c*d*f^4)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^
2)/(a^4*c^2*f^4 + a^4*d^2*f^4 + b^4*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*c^2*f
^4 + 2*a^2*b^2*d^2*f^4))^(1/2))/4 + (log(((((((((128*A*b^2*d^8*(a^2 + b^2)^2
*(a^2*d^2 - 3*b^2*c^2 - 4*b^2*d^2 + 2*a*b*c*d))/f + 64*b^2*d^8*(a^2 + b^2)^
2*(c + d*tan(e + f*x))^(1/2)*(-(4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1
/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)
^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a
^3*c*d + a*b^2*c*d))*(-(4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*
A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2
+ d^2)))^(1/2))/4 + (64*A^2*b^2*d^8*(a^2 - 3*b^2)*(c + d*tan(e + f*x))^(1/2
)*(a^3*d + 3*b^3*c - a^2*b*c + 5*a*b^2*d))/f^2)*(-(4*(-A^4*f^4*(a^2*d - b^2
*d + 2*a*b*c)^2)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^
2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 + (32*A^3*b^3*d^8*(a^3*d + 3*b
^3*c - a^2*b*c + 5*a*b^2*d))/f^3)*(-(4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^
2)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 +
 b^2)^2*(c^2 + d^2)))^(1/2))/4 + (96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2)
)/f^4)*(-((32*A^4*a^2*b^2*d^2*f^4 - 16*A^4*b^4*d^2*f^4 - 64*A^4*a^2*b^2*c^2
*f^4 - 16*A^4*a^4*d^2*f^4 + 64*A^4*a*b^3*c*d*f^4 - 64*A^4*a^3*b*c*d*f^4)^(1
/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(a^4*c^2*f^4 + a
^4*d^2*f^4 + b^4*c^2*f^4 + b^4*d^2*f^4 + 2*a^2*b^2*c^2*f^4 + 2*a^2*b^2*d^2*
f^4))^(1/2))/4 - log(((((((((128*A*b^2*d^8*(a^2 + b^2)^2*(a^2*d^2 - 3*b^2*c^
2 - 4*b^2*d^2 + 2*a*b*c*d))/f - 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x
))^(1/2)*((4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2
 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2
)*(3*b^3*c^2 + 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*
((4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f^2 + 4*A^2*
b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (6
4*A^2*b^2*d^8*(a^2 - 3*b^2)*(c + d*tan(e + f*x))^(1/2)*(a^3*d + 3*b^3*c - a
^2*b*c + 5*a*b^2*d))/f^2)*((4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2)
- 4*A^2*a^2*c*f^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(
c^2 + d^2)))^(1/2))/4 + (32*A^3*b^3*d^8*(a^3*d + 3*b^3*c - a^2*b*c + 5*a*b^
2*d))/f^3)*((4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) - 4*A^2*a^2*c*f
^2 + 4*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1
/2))/4 - (96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^4)*(((32*A^4*a^2*b^2
*d^2*f^4 - 16*A^4*b^4*d^2*f^4 - 64*A^4*a^2*b^2*c^2*f^4 - 16*A^4*a^4*d^2*f^4
 + 64*A^4*a*b^3*c*d*f^4 - 64*A^4*a^3*b*c*d*f^4)^(1/2) - 4*A^2*a^2*c*f^2 + 4
*A^2*b^2*c*f^2 + 8*A^2*a*b*d*f^2)/(16*a^4*c^2*f^4 + 16*a^4*d^2*f^4 + 16*b^4
*c^2*f^4 + 16*b^4*d^2*f^4 + 32*a^2*b^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4))^(1/2)
 - log(((((((((128*A*b^2*d^8*(a^2 + b^2)^2*(a^2*d^2 - 3*b^2*c^2 - 4*b^2*d^2
+ 2*a*b*c*d))/f - 64*b^2*d^8*(a^2 + b^2)^2*(c + d*tan(e + f*x))^(1/2)*(-(4*
(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*
c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2)*(3*b^3*c^2
+ 2*b^3*d^2 - a^2*b*c^2 - 2*a^2*b*d^2 + a^3*c*d + a*b^2*c*d))*(-(4*(-A^4*f^
4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*c*f^2 -
8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (64*A^2*b^2*d^
```

```
8*(a^2 - 3*b^2)*(c + d*tan(e + f*x))^(1/2)*(a^3*d + 3*b^3*c - a^2*b*c + 5*a
*b^2*d))/f^2)*(-(4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*A^2*a^2
*c*f^2 - 4*A^2*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2))
)^(1/2))/4 + (32*A^3*b^3*d^8*(a^3*d + 3*b^3*c - a^2*b*c + 5*a*b^2*d))/f^3)*
(-(4*(-A^4*f^4*(a^2*d - b^2*d + 2*a*b*c)^2)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2
*b^2*c*f^2 - 8*A^2*a*b*d*f^2)/(f^4*(a^2 + b^2)^2*(c^2 + d^2)))^(1/2))/4 - (
96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^4)*(-((32*A^4*a^2*b^2*d^2*f^4
- 16*A^4*b^4*d^2*f^4 - 64*A^4*a^2*b^2*c^2*f^4 - 16*A^4*a^4*d^2*f^4 + 64*A^4
*a*b^3*c*d*f^4 - 64*A^4*a^3*b*c*d*f^4)^(1/2) + 4*A^2*a^2*c*f^2 - 4*A^2*b^2*
c*f^2 - 8*A^2*a*b*d*f^2)/(16*a^4*c^2*f^4 + 16*a^4*d^2*f^4 + 16*b^4*c^2*f^4
+ 16*b^4*d^2*f^4 + 32*a^2*b^2*c^2*f^4 + 32*a^2*b^2*d^2*f^4))^(1/2) - (A*ata
n(((A*((A*((32*(A^3*a^3*b^3*d^9 + 5*A^3*a*b^5*d^9 + 3*A^3*b^6*c*d^8 - A^3*a
^2*b^4*c*d^8))/f^3 + (A*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^9*
f^2 + 2*A^2*a^5*b^2*d^9*f^2 - 30*A^2*a*b^6*d^9*f^2 - 18*A^2*b^7*c*d^8*f^2 +
 12*A^2*a^2*b^5*c*d^8*f^2 - 2*A^2*a^4*b^3*c*d^8*f^2))/f^4 - (A*((32*(16*A*b
^8*d^10*f^2 + 28*A*a^2*b^6*d^10*f^2 + 8*A*a^4*b^4*d^10*f^2 - 4*A*a^6*b^2*d^
10*f^2 + 12*A*b^8*c^2*d^8*f^2 - 16*A*a^3*b^5*c*d^9*f^2 - 8*A*a^5*b^3*c*d^9*
f^2 + 24*A*a^2*b^6*c^2*d^8*f^2 + 12*A*a^4*b^4*c^2*d^8*f^2 - 8*A*a*b^7*c*d^9
*f^2))/f^3 - (32*A*(c + d*tan(e + f*x))^(1/2)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2
+ a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*(16*
b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*
f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 -
 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*
b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b
^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^8*c*f^2 + 2*a^
2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2
)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b
*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3
*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2
*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 +
 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*
d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 -
a^4*b*c*f^2 + a*b^4*d*f^2) + (96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^
4)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3
*d*f^2 - a*b^7*d*f^2)^(1/2)*1i)/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 +
2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) - (A*((A*((32*(A^3*a^3*b^3*d^9
 + 5*A^3*a*b^5*d^9 + 3*A^3*b^6*c*d^8 - A^3*a^2*b^4*c*d^8))/f^3 - (A*((32*(c
 + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^9*f^2 + 2*A^2*a^5*b^2*d^9*f^2 - 3
0*A^2*a*b^6*d^9*f^2 - 18*A^2*b^7*c*d^8*f^2 + 12*A^2*a^2*b^5*c*d^8*f^2 - 2*A
^2*a^4*b^3*c*d^8*f^2))/f^4 + (A*((32*(16*A*b^8*d^10*f^2 + 28*A*a^2*b^6*d^10
*f^2 + 8*A*a^4*b^4*d^10*f^2 - 4*A*a^6*b^2*d^10*f^2 + 12*A*b^8*c^2*d^8*f^2 -
 16*A*a^3*b^5*c*d^9*f^2 - 8*A*a^5*b^3*c*d^9*f^2 + 24*A*a^2*b^6*c^2*d^8*f^2
+ 12*A*a^4*b^4*c^2*d^8*f^2 - 8*A*a*b^7*c*d^9*f^2))/f^3 + (32*A*(c + d*tan(e
 + f*x))^(1/2)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f
^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*
```

f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) - (96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^4)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*1i)/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))/((A*((A*((32*(A^3*a^3*b^3*d^9 + 5*A^3*a*b^5*d^9 + 3*A^3*b^6*c*d^8 - A^3*a^2*b^4*c*d^8))/f^3 + (A*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^9*f^2 + 2*A^2*a^5*b^2*d^9*f^2 - 30*A^2*a*b^6*d^9*f^2 - 18*A^2*b^7*c*d^8*f^2 + 12*A^2*a^2*b^5*c*d^8*f^2 - 2*A^2*a^4*b^3*c*d^8*f^2))/f^4 - (A*((32*(16*A*b^8*d^10*f^2 + 28*A*a^2*b^6*d^10*f^2 + 8*A*a^4*b^4*d^10*f^2 - 4*A*a^6*b^2*d^10*f^2 + 12*A*b^8*c^2*d^8*f^2 - 16*A*a^3*b^5*c*d^9*f^2 - 8*A*a^5*b^3*c*d^9*f^2 + 24*A*a^2*b^6*c^2*d^8*f^2 + 12*A*a^4*b^4*c^2*d^8*f^2 - 8*A*a*b^7*c*d^9*f^2))/f^3 - (32*A*(c + d*tan(e + f*x))^(1/2)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) + (96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^4)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) + (A*((A*((32*(A^3*a^3*b^3*d^9 + 5*A^3*a*b^5*d^9 + 3*A^3*b^6*c*d^8 - A^3*a^2*b^4*c*d^8))/f^3 - (A*((32*(c + d*tan(e + f*x))^(1/2)*(4*A^2*a^3*b^4*d^9*f^2 + 2*A^2*a^5*b^2*d^9*f^2 - 30*A^2*a*b^6*d^9*f^2 - 18*A^2*b^7*c*d^8*f^2 + 12*A^2*a^2*b^5*c*d^8*f^2 - 2*A^2*a^4*b^3*c*d^8*f^2))/f^4 + (A*((32*(16*A*b^8*d^10*f^2 + 28*A*a^2*b^6*d^10*f^2 + 8*A*a^4*b^4*d^10*f^2 - 4*A*a^6*b^2*d^10*f^2 + 12*A*b^8*c^2

```
*d^8*f^2 - 16*A*a^3*b^5*c*d^9*f^2 - 8*A*a^5*b^3*c*d^9*f^2 + 24*A*a^2*b^6*c^
2*d^8*f^2 + 12*A*a^4*b^4*c^2*d^8*f^2 - 8*A*a*b^7*c*d^9*f^2))/f^3 + (32*A*(c
 + d*tan(e + f*x))^(1/2)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a
^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2
*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*
f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^
4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7
*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*
d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4
*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 -
 b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)
)*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*
d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^
3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^
4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2))/(a^5*d*
f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d
*f^2) - (96*A^4*b^5*d^8*(c + d*tan(e + f*x))^(1/2))/f^4)*(b^8*c*f^2 + 2*a^2
*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)
^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*
c*f^2 + a*b^4*d*f^2)))*(b^8*c*f^2 + 2*a^2*b^6*c*f^2 + a^4*b^4*c*f^2 - 2*a^3
*b^5*d*f^2 - a^5*b^3*d*f^2 - a*b^7*d*f^2)^(1/2)*2i)/(a^5*d*f^2 - b^5*c*f^2
- 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) + (C*a^2*a
tan(((C*a^2*((32*(C^4*b^5*d^8 + 2*C^4*a^4*b*d^8)*(c + d*tan(e + f*x))^(1/2)
)/f^4 + (C*a^2*((32*(15*C^3*a^3*b^3*d^9*f^2 - C^3*a*b^5*d^9*f^2 - 4*C^3*a^5
*b*d^9*f^2 + C^3*b^6*c*d^8*f^2 + 9*C^3*a^2*b^4*c*d^8*f^2 - 12*C^3*a^4*b^2*c
*d^8*f^2))/f^5 - (C*a^2*((32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a^5*b^2*d^9
*f^2 - 4*C^2*a^3*b^4*d^9*f^2 + 14*C^2*a*b^6*d^9*f^2 + 10*C^2*b^7*c*d^8*f^2
- 8*C^2*a^6*b*c*d^8*f^2 - 4*C^2*a^2*b^5*c*d^8*f^2 + 10*C^2*a^4*b^3*c*d^8*f^
2))/f^4 + (C*a^2*((32*(4*C*a^2*b^6*d^10*f^4 + 8*C*a^4*b^4*d^10*f^4 + 4*C*a^
6*b^2*d^10*f^4 + 4*C*b^8*c^2*d^8*f^4 + 16*C*a^3*b^5*c*d^9*f^4 + 8*C*a^5*b^3
*c*d^9*f^4 + 8*C*a^2*b^6*c^2*d^8*f^4 + 4*C*a^4*b^4*c^2*d^8*f^4 + 8*C*a*b^7*
c*d^9*f^4))/f^5 - (32*C*a^2*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 1
6*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2
*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d
^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 +
8*a^7*b^2*c*d^9*f^4))/(f^4*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2
*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))))/(b^6*c*f^2 + 2*a^2*b^4
*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)
))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d
*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 -
2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))*1i)/(b^6*c*f^2 + 2*a^2*
b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1
/2) + (C*a^2*((32*(C^4*b^5*d^8 + 2*C^4*a^4*b*d^8)*(c + d*tan(e + f*x))^(1/2
))/f^4 - (C*a^2*((32*(15*C^3*a^3*b^3*d^9*f^2 - C^3*a*b^5*d^9*f^2 - 4*C^3*a^
5*b*d^9*f^2 + C^3*b^6*c*d^8*f^2 + 9*C^3*a^2*b^4*c*d^8*f^2 - 12*C^3*a^4*b^2*
```

c*d^8*f^2))/f^5 + (C*a^2*((32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a^5*b^2*d^9*f^2 - 4*C^2*a^3*b^4*d^9*f^2 + 14*C^2*a*b^6*d^9*f^2 + 10*C^2*b^7*c*d^8*f^2 - 8*C^2*a^6*b*c*d^8*f^2 - 4*C^2*a^2*b^5*c*d^8*f^2 + 10*C^2*a^4*b^3*c*d^8*f^2))/f^4 - (C*a^2*((32*(4*C*a^2*b^6*d^10*f^4 + 8*C*a^4*b^4*d^10*f^4 + 4*C*a^6*b^2*d^10*f^4 + 4*C*b^8*c^2*d^8*f^4 + 16*C*a^3*b^5*c*d^9*f^4 + 8*C*a^5*b^3*c*d^9*f^4 + 8*C*a^2*b^6*c^2*d^8*f^4 + 4*C*a^4*b^4*c^2*d^8*f^4 + 8*C*a*b^7*c*d^9*f^4))/f^5 + (32*C*a^2*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))*1i)/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/((C*a^2*((32*(C^4*b^5*d^8 + 2*C^4*a^4*b*d^8)*(c + d*tan(e + f*x))^(1/2))/f^4 - (C*a^2*((32*(15*C^3*a^3*b^3*d^9*f^2 - C^3*a*b^5*d^9*f^2 - 4*C^3*a^5*b*d^9*f^2 + C^3*b^6*c*d^8*f^2 + 9*C^3*a^2*b^4*c*d^8*f^2 - 12*C^3*a^4*b^2*c*d^8*f^2))/f^5 + (C*a^2*((32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a^5*b^2*d^9*f^2 - 4*C^2*a^3*b^4*d^9*f^2 + 14*C^2*a*b^6*d^9*f^2 + 10*C^2*b^7*c*d^8*f^2 - 8*C^2*a^6*b*c*d^8*f^2 - 4*C^2*a^2*b^5*c*d^8*f^2 + 10*C^2*a^4*b^3*c*d^8*f^2))/f^4 - (C*a^2*((32*(4*C*a^2*b^6*d^10*f^4 + 8*C*a^4*b^4*d^10*f^4 + 4*C*a^6*b^2*d^10*f^4 + 4*C*b^8*c^2*d^8*f^4 + 16*C*a^3*b^5*c*d^9*f^4 + 8*C*a^5*b^3*c*d^9*f^4 + 8*C*a^2*b^6*c^2*d^8*f^4 + 4*C*a^4*b^4*c^2*d^8*f^4 + 8*C*a*b^7*c*d^9*f^4))/f^5 + (32*C*a^2*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2) - (C*a^2*((32*(C^4*b^5*d^8 + 2*C^4*a^4*b*d^8)*(c + d*tan(e + f*x))^(1/2))/f^4 + (C*a^2*((32*(15*C^3*a^3*b^3*d^9*f^2 - C^3*a*b^5*d^9*f^2 - 4*C^3*a^5*b*d^9*f^2 + C^3*b^6*c*d^8*f^2 + 9*C^3*a^2*b^4*c*d^8*f^2 - 12*C^3*a^4*b^2*c*d^8*f^2))/f^5 - (C*a^2*((32*(c + d*tan(e + f*x))^(1/2)*(14*C^2*a^5*b^2*d^9*f^2 - 4*C^2*a^3*b^4*d^9*f^2 + 14*C^2*a*b^6*d^9*f^2 + 10*C^2*b^7*c*d^8*f^2 - 8*C^2*a^6*b*c*d^8*f^2 - 4*C^2*a^2*b^5*c*d^8*f^2 + 10*C^2*a^4*b^3*c*d^8*f^2))/f^4 + (C*a^2*((32*(4*C*a^2*b^6*d^10*f^4 + 8*C*a^4*b^4*d^10*f^4 + 4*C*a^6*b^2*d^10*f^4 + 4*C*b^8*c^2*d^8*f^4 + 16*C*a^3*b^5*c*d^9*f^4 + 8*C*a^5*b^3

*c*d^9*f^4 + 8*C*a^2*b^6*c^2*d^8*f^4 + 4*C*a^4*b^4*c^2*d^8*f^4 + 8*C*a*b^7*
c*d^9*f^4))/f^5 - (32*C*a^2*(c + d*tan(e + f*x))^(1/2)*(16*b^9*d^10*f^4 + 1
6*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2
*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d
^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 +
8*a^7*b^2*c*d^9*f^4))/(f^4*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2
*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))))/(b^6*c*f^2 + 2*a^2*b^4
*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)
))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d
*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 -
2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)))/(b^6*c*f^2 + 2*a^2*b^4
*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)
 + (64*C^5*a^2*b^2*d^8)/f^5))*2i)/(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*
f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2) - (B*a*atan(((B*a*
((32*(B^4*b^5*d^8 - 2*B^4*a^2*b^3*d^8)*(c + d*tan(e + f*x))^(1/2))/f^4 - (B
a((32*(13*B^3*a^2*b^4*d^9*f^2 + B^3*a^4*b^2*d^9*f^2 + 7*B^3*a*b^5*c*d^8*f
^2 - 5*B^3*a^3*b^3*c*d^8*f^2))/f^5 + (B*a*((32*(c + d*tan(e + f*x))^(1/2)*(
20*B^2*a^3*b^4*d^9*f^2 + 2*B^2*a^5*b^2*d^9*f^2 - 14*B^2*a*b^6*d^9*f^2 - 10*
B^2*b^7*c*d^8*f^2 + 12*B^2*a^2*b^5*c*d^8*f^2 - 10*B^2*a^4*b^3*c*d^8*f^2))/f
^4 + (B*a*((32*(12*B*a*b^7*d^10*f^4 + 4*B*b^8*c*d^9*f^4 + 24*B*a^3*b^5*d^10
*f^4 + 12*B*a^5*b^3*d^10*f^4 + 4*B*a*b^7*c^2*d^8*f^4 + 4*B*a^2*b^6*c*d^9*f^
4 - 4*B*a^4*b^4*c*d^9*f^4 - 4*B*a^6*b^2*c*d^9*f^4 + 8*B*a^3*b^5*c^2*d^8*f^4
 + 4*B*a^5*b^3*c^2*d^8*f^4))/f^5 - (32*B*a*(c + d*tan(e + f*x))^(1/2)*(b^6*
c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a
^5*b*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*
f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8
*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b
^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2
 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^
2)))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5
*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a
^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a
^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f
^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*
f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^
5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*
a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 +
a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)*1i)/(a^5
*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^
4*d*f^2) + (B*a*((32*(B^4*b^5*d^8 - 2*B^4*a^2*b^3*d^8)*(c + d*tan(e + f*x))
^(1/2))/f^4 + (B*a*((32*(13*B^3*a^2*b^4*d^9*f^2 + B^3*a^4*b^2*d^9*f^2 + 7*B
^3*a*b^5*c*d^8*f^2 - 5*B^3*a^3*b^3*c*d^8*f^2))/f^5 - (B*a*((32*(c + d*tan(e
 + f*x))^(1/2)*(20*B^2*a^3*b^4*d^9*f^2 + 2*B^2*a^5*b^2*d^9*f^2 - 14*B^2*a*b
^6*d^9*f^2 - 10*B^2*b^7*c*d^8*f^2 + 12*B^2*a^2*b^5*c*d^8*f^2 - 10*B^2*a^4*b
^3*c*d^8*f^2))/f^4 - (B*a*((32*(12*B*a*b^7*d^10*f^4 + 4*B*b^8*c*d^9*f^4 + 2

```
4*B*a^3*b^5*d^10*f^4 + 12*B*a^5*b^3*d^10*f^4 + 4*B*a*b^7*c^2*d^8*f^4 + 4*B*
a^2*b^6*c*d^9*f^4 - 4*B*a^4*b^4*c*d^9*f^4 - 4*B*a^6*b^2*c*d^9*f^4 + 8*B*a^3
*b^5*c^2*d^8*f^4 + 4*B*a^5*b^3*c^2*d^8*f^4))/f^5 + (32*B*a*(c + d*tan(e + f
*x))^(1/2)*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 -
 a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2*b^7*d^10*f^4 -
16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*f^4 + 40*a^2*b^7
*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^4 + 8*a*b^8*c*d^
9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7*b^2*c*d^9*f^4))
/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*
f^2 + a*b^4*d*f^2)))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b
^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2
*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a
^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)
^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*
c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*
b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^
2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*
a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2
)^(1/2)*1i)/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^
4*b*c*f^2 + a*b^4*d*f^2))/((B*a*((32*(B^4*b^5*d^8 - 2*B^4*a^2*b^3*d^8)*(c +
 d*tan(e + f*x))^(1/2))/f^4 + (B*a*((32*(13*B^3*a^2*b^4*d^9*f^2 + B^3*a^4*b
^2*d^9*f^2 + 7*B^3*a*b^5*c*d^8*f^2 - 5*B^3*a^3*b^3*c*d^8*f^2))/f^5 - (B*a*(
(32*(c + d*tan(e + f*x))^(1/2)*(20*B^2*a^3*b^4*d^9*f^2 + 2*B^2*a^5*b^2*d^9*
f^2 - 14*B^2*a*b^6*d^9*f^2 - 10*B^2*b^7*c*d^8*f^2 + 12*B^2*a^2*b^5*c*d^8*f^
2 - 10*B^2*a^4*b^3*c*d^8*f^2))/f^4 - (B*a*((32*(12*B*a*b^7*d^10*f^4 + 4*B*b
^8*c*d^9*f^4 + 24*B*a^3*b^5*d^10*f^4 + 12*B*a^5*b^3*d^10*f^4 + 4*B*a*b^7*c^
2*d^8*f^4 + 4*B*a^2*b^6*c*d^9*f^4 - 4*B*a^4*b^4*c*d^9*f^4 - 4*B*a^6*b^2*c*d
^9*f^4 + 8*B*a^3*b^5*c^2*d^8*f^4 + 4*B*a^5*b^3*c^2*d^8*f^4))/f^5 + (32*B*a*
(c + d*tan(e + f*x))^(1/2)*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2
*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)*(16*b^9*d^10*f^4 + 16*a^2
*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24*b^9*c^2*d^8*
f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b^3*c^2*d^8*f^
4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^9*f^4 + 8*a^7
*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*
d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2
*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b
^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*
(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^
2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^
2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^
2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 -
b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))
*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f
^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b
^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) - (B*a*((32*(B^4*b^5*d^8 - 2*B^4*a^2*
```

```
b^3*d^8)*(c + d*tan(e + f*x))^(1/2))/f^4 - (B*a*((32*(13*B^3*a^2*b^4*d^9*f^
2 + B^3*a^4*b^2*d^9*f^2 + 7*B^3*a*b^5*c*d^8*f^2 - 5*B^3*a^3*b^3*c*d^8*f^2))
/f^5 + (B*a*((32*(c + d*tan(e + f*x))^(1/2)*(20*B^2*a^3*b^4*d^9*f^2 + 2*B^2
*a^5*b^2*d^9*f^2 - 14*B^2*a*b^6*d^9*f^2 - 10*B^2*b^7*c*d^8*f^2 + 12*B^2*a^2
*b^5*c*d^8*f^2 - 10*B^2*a^4*b^3*c*d^8*f^2))/f^4 + (B*a*((32*(12*B*a*b^7*d^1
0*f^4 + 4*B*b^8*c*d^9*f^4 + 24*B*a^3*b^5*d^10*f^4 + 12*B*a^5*b^3*d^10*f^4 +
 4*B*a*b^7*c^2*d^8*f^4 + 4*B*a^2*b^6*c*d^9*f^4 - 4*B*a^4*b^4*c*d^9*f^4 - 4*
B*a^6*b^2*c*d^9*f^4 + 8*B*a^3*b^5*c^2*d^8*f^4 + 4*B*a^5*b^3*c^2*d^8*f^4))/f
^5 - (32*B*a*(c + d*tan(e + f*x))^(1/2)*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*
b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2)*(16*b^9*d^10
*f^4 + 16*a^2*b^7*d^10*f^4 - 16*a^4*b^5*d^10*f^4 - 16*a^6*b^3*d^10*f^4 + 24
*b^9*c^2*d^8*f^4 + 40*a^2*b^7*c^2*d^8*f^4 + 8*a^4*b^5*c^2*d^8*f^4 - 8*a^6*b
^3*c^2*d^8*f^4 + 8*a*b^8*c*d^9*f^4 + 24*a^3*b^6*c*d^9*f^4 + 24*a^5*b^4*c*d^
9*f^4 + 8*a^7*b^2*c*d^9*f^4))/(f^4*(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2
 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)))*(b^6*c*f^2 + 2*a^2*b^4*c*
f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(
a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a
*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2
 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f
^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c
*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/
(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 +
a*b^4*d*f^2))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^
2 - a*b^5*d*f^2 - a^5*b*d*f^2)^(1/2))/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*
f^2 + 2*a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2) + (64*B^5*a*b^3*d^8)/f^5
))*(b^6*c*f^2 + 2*a^2*b^4*c*f^2 + a^4*b^2*c*f^2 - 2*a^3*b^3*d*f^2 - a*b^5*d
*f^2 - a^5*b*d*f^2)^(1/2)*2i)/(a^5*d*f^2 - b^5*c*f^2 - 2*a^2*b^3*c*f^2 + 2*
a^3*b^2*d*f^2 - a^4*b*c*f^2 + a*b^4*d*f^2)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)\sqrt{c + d\tan\left(e + fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(1/2)/(a+b*tan
(f*x+e)),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))*sqr
t(c + d*tan(e + f*x))), x)
```

3.115 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=327

$$-\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))}-\frac{\left(a^4(-C)d+3a^3bBd-a^2b^2(5Ad+2Bc-3Cd)+ab^3(4Ac-Bd-4cC)\right.}{\sqrt{b}\,f\left(a^2+b^2\right)^2(bc-ad)^{3/2}}$$

```
[Out] -(3*a^3*b*B*d-a^4*C*d+b^4*(-A*d+2*B*c)+a*b^3*(4*A*c-B*d-4*C*c)-a^2*b^2*(5*A
*d+2*B*c-3*C*d))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-a*d+b*c)^(1/2))/(
a^2+b^2)^2/(-a*d+b*c)^(3/2)/f/b^(1/2)-(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^
(1/2)/(c-I*d)^(1/2))/(a-I*b)^2/f/(c-I*d)^(1/2)-(B-I*(A-C))*arctanh((c+d*tan
(f*x+e))^(1/2)/(c+I*d)^(1/2))/(a+I*b)^2/f/(c+I*d)^(1/2)-(A*b^2-a*(B*b-C*a))
*(c+d*tan(f*x+e))^(1/2)/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))
```

Rubi [A] time = 1.38, antiderivative size = 327, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3649, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))}-\frac{\left(-a^2b^2(5Ad+2Bc-3Cd)+3a^3bBd+a^4(-C)d+ab^3(4Ac-Bd-4cC\right.}{\sqrt{b}\,f\left(a^2+b^2\right)^2(bc-ad)^{3/2}}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*Sqrt[c
+ d*Tan[e + f*x]]),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((a - I
*b)^2*Sqrt[c - I*d]*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]
/Sqrt[c + I*d]])/((a + I*b)^2*Sqrt[c + I*d]*f) - ((3*a^3*b*B*d - a^4*C*d +
b^4*(2*B*c - A*d) + a*b^3*(4*A*c - 4*c*C - B*d) - a^2*b^2*(2*B*c + 5*A*d -
3*C*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(Sqrt[
b]*(a^2 + b^2)^2*(b*c - a*d)^(3/2)*f) - ((A*b^2 - a*(b*B - a*C))*Sqrt[c + d
*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x]))
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
```

```
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))} - \frac{\int \frac{\frac{1}{2}\left(Ab^2d-2aA(bc-ad)-2\right.}{}}{}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))} - \frac{\int \frac{-\left(2abB+a^2(A-C)-b^2(\right.}{}}{}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))} + \frac{(A-iB-C)\int \frac{1+i}{\sqrt{c+d}}}{2(a-ib)}$$

$$= -\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))} - \frac{(i(A+iB-C))\,\mathrm{Sub}}{}$$

$$= -\frac{\left(3a^3bBd-a^4Cd+b^4(2Bc-Ad)+ab^3(4Ac-4cC-Bd)-a^2b\right.}{\sqrt{b}\left(a^2+b^2\right)^2(bc-a}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2\sqrt{c-id}\,f} - \frac{(B-i(A-C))\tanh^{-1}\left(\right.}{(a+ib)^2\sqrt{c+}}$$

Mathematica [A] time = 6.22, size = 521, normalized size = 1.59

$$-\frac{\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))} - \frac{\frac{2\sqrt{bc-ad}\left(\frac{1}{2}a^2d\left(Ab^2-a(bB-aC)\right)+\frac{1}{2}b^2\left(-2aA(bc-ad)-2(bB-aC)\left(bc-\frac{ad}{2}\right)+Ab^2d\right)-ab(bc-ad)(\right.}{\sqrt{b}\,f\left(a^2+b^2\right)(ad-bc)}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*S
qrt[c + d*Tan[e + f*x]]),x]
```

```
[Out] -((((I*Sqrt[c - I*d]*(I*(a^2*B - b^2*B - 2*a*b*(A - C))*(b*c - a*d) - (2*a*
b*B + a^2*(A - C) - b^2*(A - C))*(b*c - a*d))*ArcTanh[Sqrt[c + d*Tan[e + f*
x]]/Sqrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*((-I)*(a^2*B - b^2*B
- 2*a*b*(A - C))*(b*c - a*d) - (2*a*b*B + a^2*(A - C) - b^2*(A - C))*(b*c -
 a*d))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((-c - I*d)*f))/(a^
2 + b^2) + (2*Sqrt[b*c - a*d]*((a^2*(A*b^2 - a*(b*B - a*C))*d)/2 - a*b*(A*b
 - a*B - b*C)*(b*c - a*d) + (b^2*(A*b^2*d - 2*a*A*(b*c - a*d) - 2*(b*B - a*
C)*(b*c - (a*d)/2)))/2)*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c
 - a*d]])/(Sqrt[b]*(a^2 + b^2)*(-(b*c) + a*d)*f))/((a^2 + b^2)*(b*c - a*d))
) - ((A*b^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c -
a*d)*f*(a + b*Tan[e + f*x]))
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^2,x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^2,x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.64, size = 20870, normalized size = 63.82

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
^2,x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^2,x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [B] time = 57.65, size = 225004, normalized size = 688.09

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))^(1/2)),x)
```

```
[Out] (atan(((((((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a
^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3
*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*
f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^
2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 -
216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*
c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4
*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5
+ a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f
^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d
*f^5 - 8*a^7*b^3*c*d*f^5) - (((((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^1
2*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*
b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c
^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^
4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 -
520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11
*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*
b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4
 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10
*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 76
0*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d
^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^1
2*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f
```

^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (16*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3* f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7

*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c
^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*
d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b
*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C
^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 2
4*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a
^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 -
 a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^
2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a
^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5
*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*
f^2))^(1/2)*1i)/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^
2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^
6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9
*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^
2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3
*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) - (((((16*(8*C^3*a^6*b^7*
d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*
b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*
a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^
2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11
*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 +
96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c
^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c
^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*
d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b
*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((
(((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^
12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^
12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*
d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*
b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 12
8*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f
^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b
^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4
- 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10
*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160
*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^
10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*
b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4
*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^
4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2
*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5
) + (16*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*
b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^

3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2

- 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c
*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^
2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d
^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C
^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d
^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^
10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2
*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*
c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b
^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5
*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C
^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d
)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 +
4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2
- 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c
*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f
^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3
*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6
*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*
f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a
^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^
5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f
^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c
*d^2*f^2))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a
^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b
*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8
*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*
a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^
2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^
2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a
^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2
*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2)
+ b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d
f^2) - b^9(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a
^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2
*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) - (16*(c + d*t
an(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^
5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4
*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3
*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b
^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a
^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 +
4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 -
12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^
2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*

b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*1i)/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2))/((((((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (((((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (16*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*(c + d*

tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6
*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b
^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^
10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4
*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b
^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 72
0*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f
^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^
2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*
b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304
*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4
- 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 -
304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4
+ 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11
*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^10*(8*a^2*c^3
*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*
a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) -
b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f
^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5
*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^
10*b^2*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*
b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4
*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*
a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)
))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d
^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2
+ 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3
*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*
d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2
+ 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a
^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*
c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3
*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8
*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) -
b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d
f^2) - b^7(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*
d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) + (16*(c + d*tan(e + f
x))^(1/2)(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a
^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^
2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 2
0*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*
d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2
*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*
d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^
2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^

8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3* a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) - (32*(3*C^5*a^3*b^6*d^10 - C^5*a^5*b^4*d^10 + 4*C^5*a*b^8*c^2*d^8 - 7*C^5*a^2*b^7*c*d^9 + C^5*a^4*b^5*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (((((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (((((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (16*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*(c + d*tan(e + f*x))^

(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2

*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9
*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4
 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^
4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f
^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((C^2*a^8
*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*
a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^1
0*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b
^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*
a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8
*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d
*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^
(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^
8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2
 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d
^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*
(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*
a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2))*((C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^
2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*C^2*a^5*
b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^
3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a^5*b^7*d
^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2
*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*
c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*
f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2))/(b^10*(8*a^2*c^3*
f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^8*(12*a
^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b
^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^
2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^2 + 36*a^5*
c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*f^2 + 6*a^1
0*b^2*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C
^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2
*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 -
4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^
7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*
c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6
*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*
b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*((
C^2*a^8*d^2 + 16*C^2*a^2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 -
24*C^2*a^3*b^5*c*d + 8*C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*
a^2*b^10*c^3*f^2 + 6*a^4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2
- a^3*b^9*d^3*f^2 - 4*a^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f
^2 - 3*a*b^11*c^2*d*f^2 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*
a^4*b^8*c*d^2*f^2 - 18*a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^
5*c^2*d*f^2 + 12*a^8*b^4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2

```
*f^2))^(1/2))/(b^10*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2
+ 24*a^8*c*d^2*f^2) + b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*
c^3*f^2 + 36*a^6*c*d^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(
2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2)
 - b^7*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f
^2 - 6*a*b^11*c^2*d*f^2 + 6*a^10*b^2*c*d^2*f^2)))*((C^2*a^8*d^2 + 16*C^2*a^
2*b^6*c^2 + 9*C^2*a^4*b^4*d^2 - 6*C^2*a^6*b^2*d^2 - 24*C^2*a^3*b^5*c*d + 8*
C^2*a^5*b^3*c*d)*(b^12*c^3*f^2 - a^11*b*d^3*f^2 + 4*a^2*b^10*c^3*f^2 + 6*a^
4*b^8*c^3*f^2 + 4*a^6*b^6*c^3*f^2 + a^8*b^4*c^3*f^2 - a^3*b^9*d^3*f^2 - 4*a
^5*b^7*d^3*f^2 - 6*a^7*b^5*d^3*f^2 - 4*a^9*b^3*d^3*f^2 - 3*a*b^11*c^2*d*f^2
 + 3*a^2*b^10*c*d^2*f^2 - 12*a^3*b^9*c^2*d*f^2 + 12*a^4*b^8*c*d^2*f^2 - 18*
a^5*b^7*c^2*d*f^2 + 18*a^6*b^6*c*d^2*f^2 - 12*a^7*b^5*c^2*d*f^2 + 12*a^8*b^
4*c*d^2*f^2 - 3*a^9*b^3*c^2*d*f^2 + 3*a^10*b^2*c*d^2*f^2))^(1/2)*2i)/(b^10*
(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^4*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2)
+ b^8*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^6*(8*a^6*c^3*f^2 + 36*a^6*c*d
^2*f^2) - b^3*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^9*(2*a^3*d^3*f^2 + 24*a
^3*c^2*d*f^2) - b^5*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^7*(8*a^5*d^3*f^
2 + 36*a^5*c^2*d*f^2) + 2*b^12*c^3*f^2 - 2*a^11*b*d^3*f^2 - 6*a*b^11*c^2*d*
f^2 + 6*a^10*b^2*c*d^2*f^2) - (atan((((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b
^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2
*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*
b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a
*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c
*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8
*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*
a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9
*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B
^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^
6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*
c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2
*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^
2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9
*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7
*b^3*c*d*f^4) + ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*
a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^
2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4
*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^
4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d
*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^
4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a
^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)
)^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a
^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B
^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f
```

^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^
5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 +
192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*
c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20
*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^1
0*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8
*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*
a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 1
2*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B
^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6
*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*
a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B
^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b
^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*
(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
+ 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2
*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^
3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B
^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 +
36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^
10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*
a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f
^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a
^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8
*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*
b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2
))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4
*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 +
6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 -
8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*B^4*
a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8
*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*
b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a
^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B
^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 -
24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^
4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d
^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d
^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*
d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2
*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^
14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*
B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^
4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15
*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 4

```
00*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2
*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B
*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4
 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6
*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48
*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f
^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 +
 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5
+ 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5
 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(((512*
B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4
*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*
a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B
^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) +
 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^
2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^
2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b
^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f
*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^1
2*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f
^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 +
48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4
- 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d
^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^1
3*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*
a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^
4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d
^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5
*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16
*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*
b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^
9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 17
6*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^
4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a
^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)
*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a
^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*
a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8
*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f
^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^
4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a
^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^
4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d
^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^
2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 +
```

```
4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*
f^4)))*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^
2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 +
 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + ((((512*B^4*a^4*b^4*c^2*f^4 - 16
*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a
^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^
4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128
*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2
*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2
)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^
4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d
^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a
^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10
 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*
B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^
5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b
^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*
b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2
*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 -
 8*a^7*b^3*c*d*f^4) - ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 25
6*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*
B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 8
96*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 -
128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a
*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*
d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^
2*f^4))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24
*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2
+ 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*
d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*
a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*
f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^
5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^
2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5
 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5
 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^
5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f
^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((((512*B^4*a^4*b^4*c^2*f^4
- 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B
^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 19
2*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 +
 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4
*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c
*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^
```

2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3* b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))/((16*(9*B^5*a^6*b^3*d^10 - B^5*a^2*b^7*d^10 - 4*B^5*a^2*b^7*c^2*d^8 + 4*B^5*a^4*b^5*c^2*d^8 + 2*B^5*a*b^8*c*d^9 + 6*B^5*a^3*b^6*c*d^9 - 12*B^5*a^5*b^4*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)

```
*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^
2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^
2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10
+ 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B
^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5
*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^
8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b
^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*
a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 -
8*a^7*b^3*c*d*f^4) + ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256
*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B
^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 89
6*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 1
28*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*
b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d
^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
+ 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2
*f^4))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*
B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 +
 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d
^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a
^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f
^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5
*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2
 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5
+ b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5
+ a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5
 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^
5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((512*B^4*a^4*b^4*c^2*f^4 -
 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^
4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192
*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 +
128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*
B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*
f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2
*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^
4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B
^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 +
 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f
^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13
*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32
*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d
^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*
B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^
```

3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2
*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^
9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^
4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f
^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*
f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512
*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^
4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4
*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*
B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2)
+ 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f
^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d
^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*
b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b
^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10
*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^1
7*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a
^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 +
480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^
11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2
*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^
4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^1
1*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 +
320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^
9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^1
1*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4
+ 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d
^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*
f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2
*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*
d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*((
(512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 1
6*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608
*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 +
896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1
/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b
*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b
^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*
a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(
e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^1
3*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d
^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f
^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8
*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*
c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^

6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2 *a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a

^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a

```
^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 +
6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f
^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^
8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a
^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4
+ a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c
^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*
b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 +
 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*
f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))*(((512*B^4*a^4*b^4*c^2*f^4
- 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B
^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 19
2*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 +
 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) + 4*B^2*a^4*c*f^2 + 4
*B^2*b^4*c*f^2 + 16*B^2*a*b^3*d*f^2 - 16*B^2*a^3*b*d*f^2 - 24*B^2*a^2*b^2*c
*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^
2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b
^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*d^2*f^4 +
b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b
^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) -
(atan((((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*
c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^
2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5
*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*
c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 1
6*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*
c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c
^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*
((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 +
17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b
^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*
c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9)
)/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*
a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6
*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 -
8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*B^4*
a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8
*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*
b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a
^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B
^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 +
24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^
4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d
^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(52*B^3*a^3*b^10
*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^
```

9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3
*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f
^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b
^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2
 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7
*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 6
0*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^
5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f
^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*
d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^
3*c*d*f^5) - ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^
2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*
b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a
^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*
a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f
^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4
 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6
*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^
(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^
5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^
2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 3
6*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c
*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^
2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^
8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B
^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^
9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^
10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2
*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*
c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b
^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5
*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b
^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2
*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*
b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a
*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c
*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8
*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*
a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^
12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*
d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*
d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*
b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464
*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f

```
^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b
^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4
- 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11
*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80
*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10
*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*
b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 -
48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*
c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b
^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^
8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*
a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(-((512*B^4*a^4*b^4*c^2*f^4 - 16*
B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^
6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4
*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*
B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*
b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)
*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^
2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d
^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12
*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^
4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176
*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^
4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3
*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b
^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 24
0*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f
^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4
*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b
^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*
b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5
*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*
a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 1
6*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2
*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^
6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2
*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*
c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b
^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5
*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^
2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2
*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^
8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6
*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^
```

```
4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8
*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^
2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*c^2*f^
4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4
*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^
6*b^2*d^2*f^4)) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B
^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4
*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*
B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128
*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^
3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2
*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 +
4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f
^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*
b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 -
8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*
B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7
*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c
^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*
d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b
*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((
-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 -
 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 6
08*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4
+ 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^
(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3
*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 +
 b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 +
4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(52*B
^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 +
 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f
^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b
^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48
*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*
c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*
B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*
d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2
*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^
2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 -
 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5
 - 8*a^7*b^3*c*d*f^5) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 -
 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 1
92*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4
- 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4
 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^
```

2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a
^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*
f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2
*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2
+ 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*
f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*
d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2
*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f
^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*
b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*
f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7
*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2
- 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4
 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4
 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^
4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f
^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((512*B^4*a^4*b^4*c^2*f^4
 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*
B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 1
92*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4
+ 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 -
4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*
c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c
^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*
b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*
a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*
B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*
B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 -
 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^1
1*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b
^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 -
512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c
^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320
*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^
10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b
^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 -
192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3
*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*
f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*
f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2
*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*
d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(-((512*B^4*a^4*b^4*c^
2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 -
 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^
4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d

*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f
^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2
*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*
b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6
*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32
*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^
8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*
b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^
8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16
*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a
^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4
- 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d
^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^
9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a
^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4
- 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*
f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^
4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*
f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d
^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^
4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4
 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4
 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4
 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^
4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f
^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f
^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*
a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4
))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*
c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4
*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^
2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2
*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*
(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
+ 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2
*f^4 + 4*a^6*b^2*d^2*f^4)))/((16*(9*B^5*a^6*b^3*d^10 - B^5*a^2*b^7*d^10 - 4
*B^5*a^2*b^7*c^2*d^8 + 4*B^5*a^4*b^5*c^2*d^8 + 2*B^5*a*b^8*c*d^9 + 6*B^5*a^
3*b^6*c*d^9 - 12*B^5*a^5*b^4*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b
^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*
b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2
*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 -
 8*a^7*b^3*c*d*f^5) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 2
56*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192
*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 -
896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 -

```
128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*
a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8
*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^
4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d
^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*
a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^
8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8
- 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4
*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b
^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*
b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a
^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)
+ ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f
^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4
- 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*
f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f
^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2
*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f
^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^
4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(
52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f
^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d
^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a
^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2
- 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b
^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 +
80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*
c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4
*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5
+ a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f
^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d
*f^5 - 8*a^7*b^3*c*d*f^5) - ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f
^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4
+ 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*
f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d
*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 1
6*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4
+ a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*
c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6
*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*
f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d
^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*
c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112
*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^
10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*
```

a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*

b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 9
12*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*
f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c
^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^1
0*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 -
48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*
f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d
^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^1
1*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d
^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4
*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^
2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2
*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2
*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2
*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^
2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c
*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d
^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
+ 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2
f^4))))/(4(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*
b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6
*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^
8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2
*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4
*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^
2*f^4 + 4*a^6*b^2*d^2*f^4)) - ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2
*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f
^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^
2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c
*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 -
 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f
^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^
4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a
^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10
- 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^1
1*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*
c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9
+ 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 +
 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4
+ 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f
^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c
*d*f^4) - ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b
^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6
*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*

b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*

f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((8*(32*B*a^2*b^15*d^12
*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^1
2*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^
10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*
c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a
^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 +
16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^
4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*
B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^
10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^
8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 -
176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^
3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*
a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4)
)/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*
a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6
*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 -
8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(-((512*B^
4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a
^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^
4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4
*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4
*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2
+ 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*
f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6
*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x
))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*
f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4
- 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48
*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 -
432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8
*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*
c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^
8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4
+ 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^1
0*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c
^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b
^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^
16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*
b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*
a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4
+ b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6
*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(
a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6
*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^

6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))*(-((512*B^4*a^4*b^4*c^2*f^4 - 16*B^4*b^8*d^2*f^4 - 256*B^4*a^2*b^6*c^2*f^4 - 16*B^4*a^8*d^2*f^4 - 256*B^4*a^6*b^2*c^2*f^4 + 192*B^4*a^2*b^6*d^2*f^4 - 608*B^4*a^4*b^4*d^2*f^4 + 192*B^4*a^6*b^2*d^2*f^4 - 896*B^4*a^3*b^5*c*d*f^4 + 896*B^4*a^5*b^3*c*d*f^4 + 128*B^4*a*b^7*c*d*f^4 - 128*B^4*a^7*b*c*d*f^4)^(1/2) - 4*B^2*a^4*c*f^2 - 4*B^2*b^4*c*f^2 - 16*B^2*a*b^3*d*f^2 + 16*B^2*a^3*b*d*f^2 + 24*B^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (atan((((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*

```
b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 -
 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*
f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2
*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^
3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f
^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a
^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^
2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b
^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b
^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*
b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a
^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5)
- ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d
^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*
b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2
 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^
2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^
2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 -
 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^
7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*(
c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*
f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d
^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14
*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 +
128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c
*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*
B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*
c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B
^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d
^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2
*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^
2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 -
 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4
 - 8*a^7*b^3*c*d*f^4) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b
^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3
*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f
^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^
3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2
*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 +
12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6
*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^
2*d*f^2))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*
B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64
*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112
*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f
```

^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c *d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^ 2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f ^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b ^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^ 4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a ^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10* f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3 *c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a ^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f ^5 - 8*a^7*b^3*c*d*f^5) - (16*(-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2* a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^ 2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11* c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b ^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^ 9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f ^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 1 8*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b ^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160 *a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^ 10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b ^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2 *d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b ^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 11 2*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f ^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^ 3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11 *b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 20 8*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^ 4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f ^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11* f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^ 11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c* d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 2 4*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3 *f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a ^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a ^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f

^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*1i)/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 1

2*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*

d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11
*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 36
8*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3
*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B
*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^
8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*
b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 -
32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2
*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^
14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5
 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*
f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d
^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5
*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (16*(-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 +
4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 +
 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 -
 b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 -
 a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2
+ 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c
*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f
^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3
*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4
 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 -
160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*
a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^
17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336
*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^
4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3
*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b
^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 24
0*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^
4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*
d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*
d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c
*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^
8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*
b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f
^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a
^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^
8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f
^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*
f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4
 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*
f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d
^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5

```
*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b
^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*
f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*
c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 +
 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^
2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2
*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*
(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2
) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2
*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36
*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*
a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3
*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8
*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) -
b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d
*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^
3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*1i)/(b^9*(8*a^2*c^3*f^2 +
 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^
3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8
*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) -
b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d
*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d
^2*f^2))/((16*(9*B^5*a^6*b^3*d^10 - B^5*a^2*b^7*d^10 - 4*B^5*a^2*b^7*c^2*d^
8 + 4*B^5*a^4*b^5*c^2*d^8 + 2*B^5*a*b^8*c*d^9 + 6*B^5*a^3*b^6*c*d^9 - 12*B^
5*a^5*b^4*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*
b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4
*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*
a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5)
 + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*
d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5
*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^
2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f
^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c
^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2
- 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a
^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*
(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17*B^
4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9*c^
2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*d^9
 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/(a^
10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b
^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*
b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3
*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-(4*B^2*b^7*c^2
- 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d
```

^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^
6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2
 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^
2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c
*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^
2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12
*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(52*B^3*a^3*b^10*d^11*
f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b^4*
d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*b^1
2*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2 + 1
76*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*c*d
^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 - 24*
B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^3*d
^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B^3*
a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*
a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4
*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5
 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*
f^5) - ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*
b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2
*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^
3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d
^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^
10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*
f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 +
12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(
(16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^10*
d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^11*
b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^2*a
*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^10*f
^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^10*
b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^2 +
 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a^5*
b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^2 +
 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b^5*
c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 +
4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4
 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*
f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*
d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*
a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^
2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*
c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b
^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^
9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f

^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (16*(-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8

```
*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^
2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f
^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4
+ 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f
^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^
2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*
c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^
3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f
^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c
^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 +
24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2
 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*
c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(
12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2)
 - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*
d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*
a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a
^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*
f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*
a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b
^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*
f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3
*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a
^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^
2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9
*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*
(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2
) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f
^2) - ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b
^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*
a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3
*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^
3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^1
0*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f
^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 1
2*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((
16*(c + d*tan(e + f*x))^(1/2)*(3*B^4*a^2*b^9*d^10 - 3*B^4*a^4*b^7*d^10 + 17
*B^4*a^6*b^5*d^10 - 9*B^4*a^8*b^3*d^10 + 6*B^4*b^11*c^2*d^8 - 8*B^4*a^2*b^9
*c^2*d^8 + 14*B^4*a^4*b^7*c^2*d^8 - 4*B^4*a^6*b^5*c^2*d^8 - 8*B^4*a*b^10*c*
d^9 + 12*B^4*a^3*b^8*c*d^9 - 32*B^4*a^5*b^6*c*d^9 + 12*B^4*a^7*b^4*c*d^9))/
(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^
6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a
^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*
a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-(4*B^2*b^7*c
^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^
```

3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a
*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*
f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3
*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*
b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2
*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 -
12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(52*B^3*a^3*b^10*d^
11*f^2 - 128*B^3*a^5*b^8*d^11*f^2 - 24*B^3*a^7*b^6*d^11*f^2 + 160*B^3*a^9*b
^4*d^11*f^2 + 4*B^3*a^11*b^2*d^11*f^2 + 12*B^3*b^13*c^3*d^8*f^2 + 44*B^3*a*
b^12*c^2*d^9*f^2 - 128*B^3*a^2*b^11*c*d^10*f^2 + 48*B^3*a^4*b^9*c*d^10*f^2
+ 176*B^3*a^6*b^7*c*d^10*f^2 - 48*B^3*a^8*b^5*c*d^10*f^2 - 48*B^3*a^10*b^3*
c*d^10*f^2 - 112*B^3*a^2*b^11*c^3*d^8*f^2 + 192*B^3*a^3*b^10*c^2*d^9*f^2 -
24*B^3*a^4*b^9*c^3*d^8*f^2 - 72*B^3*a^5*b^8*c^2*d^9*f^2 + 80*B^3*a^6*b^7*c^
3*d^8*f^2 - 160*B^3*a^7*b^6*c^2*d^9*f^2 - 20*B^3*a^8*b^5*c^3*d^8*f^2 + 60*B
^3*a^9*b^4*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 +
6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5
+ 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f
^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c
*d*f^5) + ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a
^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*
B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9
*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^
8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a
*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2
*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2
+ 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2
)*((16*(c + d*tan(e + f*x))^(1/2)*(68*B^2*a^3*b^12*d^11*f^2 + 20*B^2*a^5*b^
10*d^11*f^2 - 88*B^2*a^7*b^8*d^11*f^2 + 40*B^2*a^9*b^6*d^11*f^2 + 84*B^2*a^
11*b^4*d^11*f^2 + 4*B^2*a^13*b^2*d^11*f^2 + 36*B^2*b^15*c^3*d^8*f^2 + 36*B^
2*a*b^14*c^2*d^9*f^2 - 128*B^2*a^2*b^13*c*d^10*f^2 - 112*B^2*a^4*b^11*c*d^1
0*f^2 + 128*B^2*a^6*b^9*c*d^10*f^2 + 32*B^2*a^8*b^7*c*d^10*f^2 - 128*B^2*a^
10*b^5*c*d^10*f^2 - 48*B^2*a^12*b^3*c*d^10*f^2 - 68*B^2*a^2*b^13*c^3*d^8*f^
2 + 204*B^2*a^3*b^12*c^2*d^9*f^2 - 184*B^2*a^4*b^11*c^3*d^8*f^2 + 200*B^2*a
^5*b^10*c^2*d^9*f^2 - 40*B^2*a^6*b^9*c^3*d^8*f^2 - 8*B^2*a^7*b^8*c^2*d^9*f^
2 + 20*B^2*a^8*b^7*c^3*d^8*f^2 + 20*B^2*a^9*b^6*c^2*d^9*f^2 - 20*B^2*a^10*b
^5*c^3*d^8*f^2 + 60*B^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4
+ 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*
f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d
^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5
*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B
^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16
*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^
11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^
8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4
*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^

2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((8*(32*B*a^2*b^15*d^12*f^4 + 96*B*a^4*b^13*d^12*f^4 - 320*B*a^8*b^9*d^12*f^4 - 480*B*a^10*b^7*d^12*f^4 - 288*B*a^12*b^5*d^12*f^4 - 64*B*a^14*b^3*d^12*f^4 + 64*B*b^17*c^2*d^10*f^4 + 48*B*b^17*c^4*d^8*f^4 - 112*B*a*b^16*c^3*d^9*f^4 - 400*B*a^3*b^14*c*d^11*f^4 - 544*B*a^5*b^12*c*d^11*f^4 - 80*B*a^7*b^10*c*d^11*f^4 + 480*B*a^9*b^8*c*d^11*f^4 + 464*B*a^11*b^6*c*d^11*f^4 + 160*B*a^13*b^4*c*d^11*f^4 + 16*B*a^15*b^2*c*d^11*f^4 + 368*B*a^2*b^15*c^2*d^10*f^4 + 224*B*a^2*b^15*c^4*d^8*f^4 - 512*B*a^3*b^14*c^3*d^9*f^4 + 832*B*a^4*b^13*c^2*d^10*f^4 + 400*B*a^4*b^13*c^4*d^8*f^4 - 880*B*a^5*b^12*c^3*d^9*f^4 + 880*B*a^6*b^11*c^2*d^10*f^4 + 320*B*a^6*b^11*c^4*d^8*f^4 - 640*B*a^7*b^10*c^3*d^9*f^4 + 320*B*a^8*b^9*c^2*d^10*f^4 + 80*B*a^8*b^9*c^4*d^8*f^4 - 80*B*a^9*b^8*c^3*d^9*f^4 - 176*B*a^10*b^7*c^2*d^10*f^4 - 32*B*a^10*b^7*c^4*d^8*f^4 + 128*B*a^11*b^6*c^3*d^9*f^4 - 192*B*a^12*b^5*c^2*d^10*f^4 - 16*B*a^12*b^5*c^4*d^8*f^4 + 48*B*a^13*b^4*c^3*d^9*f^4 - 48*B*a^14*b^3*c^2*d^10*f^4 - 96*B*a*b^16*c*d^11*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (16*(-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) -

```
b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d
*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^
3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f
^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^
2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2
*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b
^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) +
 b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^
2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^
9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2
 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*
f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a
^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^
7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f
^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c
^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 +
36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 +
6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c
^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*
(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2)
- b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2
*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*
c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))/(b^9*(8*a^2*c^3*f^2 +
6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3
*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*
a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b
^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*
f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^
2*f^2)))*(-(4*B^2*b^7*c^2 - 8*B^2*a^2*b^5*c^2 + 4*B^2*a^4*b^3*c^2 + B^2*a^2
*b^5*d^2 - 6*B^2*a^4*b^3*d^2 + 9*B^2*a^6*b*d^2 + 16*B^2*a^3*b^4*c*d - 12*B^
2*a^5*b^2*c*d - 4*B^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c
^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*
d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b
^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d
*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 +
 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*
2i)/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*
d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 3
6*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f
^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a
^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10
*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - (atan((((((512*C^4*a^4*b^4*c^2*f^4 - 16*
C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^
6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4
*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*
```

```
C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*
b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)
*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^
2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^
2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10
+ 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^
4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*
c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*
f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2
*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*
c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*
b^3*c*d*f^4) + ((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a
^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2
*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*
a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4
*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*
f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4
 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^
6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))
^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*
b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*
b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2
 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c
*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216
*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2
*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^
2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a
^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5
- 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^
5 - 8*a^7*b^3*c*d*f^5) - ((((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4
 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 +
 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^
4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f
^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*
C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 +
 a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^
2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b
^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 +
 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4
 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4
+ 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^1
1*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b
^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*
a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8
```

*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10

```
*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^1
2*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*
f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^
6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^
2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b
^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b
^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*
b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a
^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))
*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4
- 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 -
608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4
 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)
^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^
3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4
+ b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 +
 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8
*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*
a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*
d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2
*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*c^2*f^4
 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*
c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6
*b^2*d^2*f^4)) + ((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4
*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a
^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^
4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C
^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*
d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f
^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*
a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4
))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^
7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^
4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b
^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d
^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 +
 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4
+ 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4
 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*C^
4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a
^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^
4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4
*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4
*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2
```

+ 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*ta

n(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b
^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7
*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10
*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d
^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^1
5*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*
a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4
 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*
d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^
7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a
^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 -
48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 3
04*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 +
 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f
^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8
*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^
4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d
^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^
4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f
^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*
f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^
8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6
*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^
4 + 4*a^6*b^2*d^2*f^4)) - (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d
^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*
b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*
b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 10
0*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*
d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^
2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^
3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C
^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^
9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^
2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^
2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d
^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*
c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(((512*C^4*a^4*b^4*c^2*f
^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 25
6*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 +
 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^
4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2
 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^
2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6
*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^
4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 +

b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b
^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/
(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f
^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*
d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*
f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f
^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))/((32*(3*C
^5*a^3*b^6*d^10 - C^5*a^5*b^4*d^10 + 4*C^5*a*b^8*c^2*d^8 - 7*C^5*a^2*b^7*c*
d^9 + C^5*a^4*b^5*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5
+ 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5
 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*
f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*
c*d*f^5) - ((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b
^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6
*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*
b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7
*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2
+ 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b
^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^
2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/
2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10
 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10
*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*
d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(
a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6
*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^
6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a
^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*C^4*a^4*
b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2
*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*
d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b
^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a
^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C
^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 +
4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f
^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11
*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*
d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*
b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 -
32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2
*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C
^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d
^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f
^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*
f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d

*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((((
512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16
*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*
C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 8
96*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/
2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*
d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^
8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a
^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a
^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*
C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*
b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*
C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f
^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*
c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^
3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f
^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b
^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4
- 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^
4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C
*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*
f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5
+ 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5
 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^
5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(((512
*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^
4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4
*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*
C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2)
- 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f
^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d
^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*
b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e +
f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^
12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*
f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 +
 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4
 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*
d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^
13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400
*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f
^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*
d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^
5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^1
6*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3

*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + ((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^^

2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7* b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a

```
^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^
9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^
8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2
 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c
^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*
d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b
^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a
^7*b^3*c*d*f^4))*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*
a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^
2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4
*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^
4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 16*C^2*a*b^3*d
*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^
4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a
^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)
)^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*
b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6
*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^
8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2
*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4
*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4
 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^
2*f^4 + 4*a^6*b^2*d^2*f^4))))*(((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f
^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4
 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*
f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d
*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) - 4*C^2*a^4*c*f^2 - 4*C^2*b^4*c*f^2 - 1
6*C^2*a*b^3*d*f^2 + 16*C^2*a^3*b*d*f^2 + 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4
 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*
c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6
*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*
d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2
*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (atan((((-((512*C^
4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a
^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^
4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4
*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4
*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2
 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*
f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6
*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e
+ f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^1
0 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*
b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*
c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c
```

^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f

^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(-((512*C^4*a^4*b^4*c^2*f
^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 25
6*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 +
192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^
4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2
+ 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^
2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6
*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^
4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^
2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b
^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5
*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f
^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^
3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*
b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 8
80*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*
f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c
^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12
*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 1
6*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4
- 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 -
464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4
+ 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11
*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 +
b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 +
4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 +
b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 +
a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 +
4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4
- 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4
+ b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6
*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))
+ (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^1
2*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^
11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2
*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^
2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b
^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 -
124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b
^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 -
68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c
^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4
*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4
+ a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f
^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d

```
*f^4 - 8*a^7*b^3*c*d*f^4))*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4
 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 +
 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^
4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f
^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*
C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 +
 a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^
2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b
^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f
^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*
d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d
^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
+ 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2
*f^4)))*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a
^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4
+ 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + ((-((512*C^4*a^4*b^4*c^2*f^4 -
16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4
*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*
C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 1
28*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C
^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f
^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*
f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4
*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4
*a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^
10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4
*a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b
^6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c
^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*
d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b
^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a
^7*b^3*c*d*f^4) - ((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C
^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4
*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*
C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128
*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^
3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2
*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 +
4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f
^4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*
a^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^
3*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10
*f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b
^2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 -
 216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7
```

*c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^1

1*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 2
08*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f
^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*
f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11
*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d
^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c
*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2
*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 +
6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2
*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^
2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 -
2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4
- 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*
d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2
*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (16*(c + d*tan(e +
f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2
*a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*
C^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 -
20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*
c*d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C
^2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^
2*d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*
C^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*
d^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*
a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6
*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 +
4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4
- 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d
f^4))(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c
^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2
*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*
c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c
*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16
*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c
^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^
2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/
(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f
^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*
d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4
+ b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4
+ 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*1i)/(4*(a^8*
c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a
^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
+ 4*a^6*b^2*d^2*f^4)))/((32*(3*C^5*a^3*b^6*d^10 - C^5*a^5*b^4*d^10 + 4*C^5*
a*b^8*c^2*d^8 - 7*C^5*a^2*b^7*c*d^9 + C^5*a^4*b^5*c*d^9))/(a^10*d^2*f^5 + b

^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a
^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 +
4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 -
12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((-((512*C^4*a^4*b^4*c^2*f^4 - 1
6*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*
a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C
^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 12
8*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^
2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^
2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f
^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*
d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*
a^2*b^9*d^10 - 5*C^4*a^4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^1
0 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*
a^4*b^7*c^2*d^8 - 4*C^4*a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^
6*c*d^9 + 8*C^4*a^7*b^4*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^
2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d
^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^
9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^
7*b^3*c*d*f^4) + ((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^
4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*
a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C
^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*
C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3
*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*
f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4
*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^
4))^(1/2)*((16*(8*C^3*a^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a
^8*b^5*d^11*f^2 - 24*C^3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3
*a*b^12*c^3*d^8*f^2 + 152*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*
f^2 - 64*C^3*a^7*b^6*c*d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^
2*c*d^10*f^2 - 40*C^3*a^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 -
216*C^3*a^4*b^9*c^2*d^9*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*
c^2*d^9*f^2 + 56*C^3*a^8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4
*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5
+ a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f
^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d
*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2
*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f
^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^
2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c
*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 +
16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f
^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^
4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a

^6*b^2*d^2*f^4))^(1/2)*((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^12*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^12*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 - 640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^10*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4

- 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d ^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2* b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2* a^7*b^8*d^11*f^2 + 40*C^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C ^2*a^13*b^2*d^11*f^2 - 20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c *d^10*f^2 - 160*C^2*a^8*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^ 2*a^12*b^3*c*d^10*f^2 + 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2 *d^9*f^2 + 216*C^2*a^4*b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C ^2*a^6*b^9*c^3*d^8*f^2 + 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d ^8*f^2 + 60*C^2*a^9*b^6*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a ^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6* a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4 *a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d* f^4))*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^ 2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2* f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c *d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c* d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16* C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^ 2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2 *f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^ 4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d ^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2* f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b ^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4* a^6*b^2*d^2*f^4)) + ((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256 *C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C ^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 89 6*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 1 28*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a* b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d ^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2 *f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(2*C^4*a^2*b^9*d^10 - 5*C^4*a^ 4*b^7*d^10 + 17*C^4*a^6*b^5*d^10 - 7*C^4*a^8*b^3*d^10 + 2*C^4*b^11*c^2*d^8 + C^4*a^10*b*d^10 - 12*C^4*a^2*b^9*c^2*d^8 + 18*C^4*a^4*b^7*c^2*d^8 - 4*C^4 *a*b^10*c*d^9 + 16*C^4*a^3*b^8*c*d^9 - 36*C^4*a^5*b^6*c*d^9 + 8*C^4*a^7*b^4 *c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f ^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*

f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d
*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((5
12*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*
C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C
^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 89
6*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2
) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d
*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8
*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^
2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(8*C^3*a
^6*b^7*d^11*f^2 - 78*C^3*a^4*b^9*d^11*f^2 + 60*C^3*a^8*b^5*d^11*f^2 - 24*C^
3*a^10*b^3*d^11*f^2 + 2*C^3*a^12*b*d^11*f^2 - 32*C^3*a*b^12*c^3*d^8*f^2 + 1
52*C^3*a^3*b^10*c*d^10*f^2 + 128*C^3*a^5*b^8*c*d^10*f^2 - 64*C^3*a^7*b^6*c*
d^10*f^2 - 32*C^3*a^9*b^4*c*d^10*f^2 + 8*C^3*a^11*b^2*c*d^10*f^2 - 40*C^3*a
^2*b^11*c^2*d^9*f^2 + 64*C^3*a^3*b^10*c^3*d^8*f^2 - 216*C^3*a^4*b^9*c^2*d^9
*f^2 + 96*C^3*a^5*b^8*c^3*d^8*f^2 - 120*C^3*a^6*b^7*c^2*d^9*f^2 + 56*C^3*a^
8*b^5*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^
4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a
^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 -
2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^
5) - ((((-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*
c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^
2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5
*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*
c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 1
6*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*
c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c
^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*
((16*(40*C*a^3*b^14*d^12*f^4 + 192*C*a^5*b^12*d^12*f^4 + 360*C*a^7*b^10*d^1
2*f^4 + 320*C*a^9*b^8*d^12*f^4 + 120*C*a^11*b^6*d^12*f^4 - 8*C*a^15*b^2*d^1
2*f^4 + 8*C*b^17*c^3*d^9*f^4 + 40*C*a*b^16*c^2*d^10*f^4 + 32*C*a*b^16*c^4*d
^8*f^4 - 88*C*a^2*b^15*c*d^11*f^4 - 448*C*a^4*b^13*c*d^11*f^4 - 920*C*a^6*b
^11*c*d^11*f^4 - 960*C*a^8*b^9*c*d^11*f^4 - 520*C*a^10*b^7*c*d^11*f^4 - 128
*C*a^12*b^5*c*d^11*f^4 - 8*C*a^14*b^3*c*d^11*f^4 - 32*C*a^2*b^15*c^3*d^9*f^
4 + 256*C*a^3*b^14*c^2*d^10*f^4 + 160*C*a^3*b^14*c^4*d^8*f^4 - 280*C*a^4*b^
13*c^3*d^9*f^4 + 680*C*a^5*b^12*c^2*d^10*f^4 + 320*C*a^5*b^12*c^4*d^8*f^4 -
640*C*a^6*b^11*c^3*d^9*f^4 + 960*C*a^7*b^10*c^2*d^10*f^4 + 320*C*a^7*b^10*
c^4*d^8*f^4 - 680*C*a^8*b^9*c^3*d^9*f^4 + 760*C*a^9*b^8*c^2*d^10*f^4 + 160*
C*a^9*b^8*c^4*d^8*f^4 - 352*C*a^10*b^7*c^3*d^9*f^4 + 320*C*a^11*b^6*c^2*d^1
0*f^4 + 32*C*a^11*b^6*c^4*d^8*f^4 - 72*C*a^12*b^5*c^3*d^9*f^4 + 56*C*a^13*b
^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*
b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4
*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*
a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5)
+ (4*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^

```
2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*
f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c
*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*
d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*
C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^
2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2
*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c
 + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 28
8*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a
^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c
^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^1
7*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*
a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4
 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d
^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b
^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*
a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4
 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9
*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*
f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^1
1*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c
*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^
4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4
*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^
6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6
*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^
6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9
*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))
/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*
f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4
*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (16*(c + d*tan(e + f*x))^(1/2)*(20*C^2*a^5
*b^10*d^11*f^2 - 60*C^2*a^3*b^12*d^11*f^2 + 168*C^2*a^7*b^8*d^11*f^2 + 40*C
^2*a^9*b^6*d^11*f^2 - 44*C^2*a^11*b^4*d^11*f^2 + 4*C^2*a^13*b^2*d^11*f^2 -
20*C^2*b^15*c^3*d^8*f^2 - 4*C^2*a^14*b*c*d^10*f^2 - 20*C^2*a*b^14*c^2*d^9*f
^2 + 100*C^2*a^2*b^13*c*d^10*f^2 - 300*C^2*a^6*b^9*c*d^10*f^2 - 160*C^2*a^8
*b^7*c*d^10*f^2 + 76*C^2*a^10*b^5*c*d^10*f^2 + 32*C^2*a^12*b^3*c*d^10*f^2 +
 116*C^2*a^2*b^13*c^3*d^8*f^2 - 124*C^2*a^3*b^12*c^2*d^9*f^2 + 216*C^2*a^4*
b^11*c^3*d^8*f^2 - 40*C^2*a^5*b^10*c^2*d^9*f^2 + 8*C^2*a^6*b^9*c^3*d^8*f^2
+ 168*C^2*a^7*b^8*c^2*d^9*f^2 - 68*C^2*a^8*b^7*c^3*d^8*f^2 + 60*C^2*a^9*b^6
*c^2*d^9*f^2 + 4*C^2*a^10*b^5*c^3*d^8*f^2 - 44*C^2*a^11*b^4*c^2*d^9*f^2))/(
a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6
*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^
6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a
^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(-((512*C^4*a^4*b
^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*
```

f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))*(-((512*C^4*a^4*b^4*c^2*f^4 - 16*C^4*b^8*d^2*f^4 - 256*C^4*a^2*b^6*c^2*f^4 - 16*C^4*a^8*d^2*f^4 - 256*C^4*a^6*b^2*c^2*f^4 + 192*C^4*a^2*b^6*d^2*f^4 - 608*C^4*a^4*b^4*d^2*f^4 + 192*C^4*a^6*b^2*d^2*f^4 - 896*C^4*a^3*b^5*c*d*f^4 + 896*C^4*a^5*b^3*c*d*f^4 + 128*C^4*a*b^7*c*d*f^4 - 128*C^4*a^7*b*c*d*f^4)^(1/2) + 4*C^2*a^4*c*f^2 + 4*C^2*b^4*c*f^2 + 16*C^2*a*b^3*d*f^2 - 16*C^2*a^3*b*d*f^2 - 24*C^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (atan(((((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*

d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*
f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2
*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*
d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((5
12*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*
A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A
^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 89
6*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2
) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d
*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8
*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^
2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*t
an(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 +
552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*
f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11
*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*
c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500
*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^
10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*
A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3
*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A
^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d
^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f
^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*
f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d
*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((51
2*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A
^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^
4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896
*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2)
 - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*
f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*
d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2
*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A*a*b
^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b
^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^1
1*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*
c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^
2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4
- 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*
d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3
*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^
4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^
11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4
- 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c

^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*
A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10
*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5
 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^
5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f
^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(((51
2*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A
^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^
4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896
*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2)
 - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*
f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*
d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2
*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e +
 f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d
^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12
*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4
 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^
4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4
*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b
^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 40
0*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*
f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2
*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b
^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^
16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^
3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*
a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 +
176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*
f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4
*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^
4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4
*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 +
6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 -
 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2
*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*
b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4
*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*
f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6
*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*
d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^
2*f^4)))*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*
c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^
2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5

*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) + (((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 -

500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*

```
f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^
8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15
*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a
^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4
+ 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d
^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7
*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^
14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 4
8*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 30
4*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 +
880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^
4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*
d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4
 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^
2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4
 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^
4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f
^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8
*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*
a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*
d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2
*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 +
a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2
*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^
2*d^2*f^4)))*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*
b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^
6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3
*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^
7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2
 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 +
b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b
^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1
/2)*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b
^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*
a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))/((32*(5*A^5*a^3*b^6*d^10 + A^5*a*b^8
*d^10 - A^5*b^9*c*d^9 + 4*A^5*a*b^8*c^2*d^8 - 9*A^5*a^2*b^7*c*d^9))/(a^10*d
^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c
^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*
d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7
*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (((16*(c + d*tan(e + f
*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A
^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*
b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*
d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4
```

+ 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*

a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4
*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*
f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4
 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^
6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))
^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d
^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8
*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2
*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c
^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A
*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^
4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^
3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*
A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d
^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*
a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*
f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b
^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 7
2*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^
5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f
^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*
d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^
3*c*d*f^5) - (4*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a
^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2
*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*
a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4
*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*
f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4
 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^
6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))
^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12
*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^
4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 +
32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 27
2*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f
^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2
*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b
^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 10
08*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f
^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3
*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^
4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18
*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b
^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a
^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a

```
^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 +
6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f
^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 +
6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 +
 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^
4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*
d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2
*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 +
6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b
^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^
2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(
4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^
4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d
^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f
^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4
 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*
f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d
*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 1
6*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4
 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*
c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6
*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2
*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^
6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (((16*(c + d*tan(e +
f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*
A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4
*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c
*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4
 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^
4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f
^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((((512*
A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4
*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*
a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A
^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) -
 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^
2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^
2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b
^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13
*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a
^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3
*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*
f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b
^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2
- 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^
```

2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*

b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 +
72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f
^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*
f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c
*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b
^3*c*d*f^5) + (4*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*
a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^
2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4
*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^
4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d
*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^
4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a
^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)
)^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^1
2*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f
^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 +
32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 2
72*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*
f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^
2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*
b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1
008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*
f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^
3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b
^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^1
8*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*
b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*
a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((
a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 +
6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*
f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 +
6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4
+ 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f
^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c
*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^
2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 +
6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 +
b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b
^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/
(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f
^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*
d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*
f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^
4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2
*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*

d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))*(((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) - 4*A^2*a^4*c*f^2 - 4*A^2*b^4*c*f^2 - 16*A^2*a*b^3*d*f^2 + 16*A^2*a^3*b*d*f^2 + 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (atan(((((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c* d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (4*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4

- 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 +

```
4*a^6*b^2*d^2*f^4)) + (((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*
A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c
^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d
^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2
*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*
c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b
^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5
*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b
^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2
*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*
b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a
*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c
*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8
*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*
a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4
 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^
11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b
^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*
a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^
2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*
b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2)
)/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*
a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6
*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 -
8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((-((512*A^4*
a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8
*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*
b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a
^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A
^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 -
24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^
4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d
^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(c + d*tan(e +
f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2
*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4
*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 +
4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*
f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^
8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2
+ 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4
*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^
2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*
b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2)
)/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*
a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6
```

*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (4*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^

2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))/((32*(5*A^5*a^3*b^6*d^10 + A^5*a*b^8*d^10 - A^5*b^9*c*d^9 + 4*A^5*a*b^8*c^2*d^8 - 9*A^5*a^2*b^7*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (((16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) - ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a

^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A
^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 +
4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f
^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d^11*f^
2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d
^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*
c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80
*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*
d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^
3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 +
6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 +
 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^
5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*
d*f^5) - ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^
6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*
d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b
^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*
b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 -
 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^
8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2
*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2
)*((16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b
^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2
*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*
A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 8
0*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c
*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^
2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2
*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*
A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*
d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2
*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 +
6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 +
 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^
4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*
d*f^4) + ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^
6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*
d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b
^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*
b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 -
 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^
8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2
*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2
)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f
^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12

```
*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12
*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^
8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*
b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 1
44*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9
*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4
*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^
4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b
^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 +
 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^
2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a
^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6
*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 +
4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5
 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d
*f^5) - (4*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b
^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6
*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*
b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7
*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2
 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b
^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^
2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/
2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4
 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 -
288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b
^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^
2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 +
 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^1
0*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*
c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a
^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 -
 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9
*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^
3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3
*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*
c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*
b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c
^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^
4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 +
 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^
4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a
^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 -
2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^
4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6
```

```
*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^
4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c
^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^
2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a
^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 +
6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f
^4 + 4*a^6*b^2*d^2*f^4)))*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4
- 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 +
192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4
 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^
4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A
^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 +
a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2
*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^
2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^
4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d
^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (((16*(c + d*tan(e + f*x
))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4
*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^
7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^
9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 +
4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 +
 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4
- 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4) + ((-((512*A^
4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a
^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^
4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4
*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4
*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2
- 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*
f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6
*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*((16*(2*A^3*b^13*d
^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6
*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a
*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^
2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^1
0*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 -
216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*
f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2
*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*
c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*
b^3*c*d*f^5) + ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*
a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^
2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4
*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^
```

```
4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d
*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^
4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a
^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)
)^(1/2)*((16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2
*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 -
12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^
2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f
^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6
*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 +
 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^
12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^
2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^
7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 -
12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*
f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2
*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*
c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*
b^3*c*d*f^4) - ((-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*
a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^
2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4
*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^
4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d
*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^
4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a
^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)
)^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*
d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^
8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^
2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*
c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*
A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f
^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c
^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520
*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*
d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A
*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10
*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*
b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 +
72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f
^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*
f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c
*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b
^3*c*d*f^5) + (4*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4
*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a
```

^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)))*(-((512*A^4*a^4*b^4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2))/(4*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))))*(-((512*A^4*a^4*b^

4*c^2*f^4 - 16*A^4*b^8*d^2*f^4 - 256*A^4*a^2*b^6*c^2*f^4 - 16*A^4*a^8*d^2*f^4 - 256*A^4*a^6*b^2*c^2*f^4 + 192*A^4*a^2*b^6*d^2*f^4 - 608*A^4*a^4*b^4*d^2*f^4 + 192*A^4*a^6*b^2*d^2*f^4 - 896*A^4*a^3*b^5*c*d*f^4 + 896*A^4*a^5*b^3*c*d*f^4 + 128*A^4*a*b^7*c*d*f^4 - 128*A^4*a^7*b*c*d*f^4)^(1/2) + 4*A^2*a^4*c*f^2 + 4*A^2*b^4*c*f^2 + 16*A^2*a*b^3*d*f^2 - 16*A^2*a^3*b*d*f^2 - 24*A^2*a^2*b^2*c*f^2)*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4))^(1/2)*1i)/(2*(a^8*c^2*f^4 + a^8*d^2*f^4 + b^8*c^2*f^4 + b^8*d^2*f^4 + 4*a^2*b^6*c^2*f^4 + 6*a^4*b^4*c^2*f^4 + 4*a^6*b^2*c^2*f^4 + 4*a^2*b^6*d^2*f^4 + 6*a^4*b^4*d^2*f^4 + 4*a^6*b^2*d^2*f^4)) - (atan((((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 +

160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (16*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 -

4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*1i)/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24* a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f ^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3 *d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^ 6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6* a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - ((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c ^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a *b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3* f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3 *f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10* b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2 *f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((((16*(2*A^3*b^13*d^11 *f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^ 7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^ 12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c ^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216 *A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^ 5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d *f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3 *c*d*f^5) - ((((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 2 5*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b ^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a ^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d ^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a ^9*b^2*c^2*d*f^2))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f ^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12* f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f ^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c *d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A* a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14* c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 32 0*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2 *d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A *a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9* f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b ^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f ^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^ 2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5* c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (16*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2

+ 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*
c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12
*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^
4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f
^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*
d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^
3*c*d*f^4))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A
^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11
*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*
b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a
^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*
f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 -
18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*
b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*
c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5
*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2)
 - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^
2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11
*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*(-(A^2*b^7*d^2 + 16*A^
2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*
d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a
^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*
a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^
2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^
4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*
c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2
*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*
(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2
) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2
*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36
*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*
a^10*b*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a
^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^
8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 -
24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4
+ 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f
^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^
2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*
c*d*f^4 - 8*a^7*b^3*c*d*f^4))*1i)/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) +
b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2
f^2) + b^5(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9
*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2
+ 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f
^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))/(((-(A^2*b^
7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A

^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - (16*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 -

12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) + (16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^

5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - (32*(5*A^5*a^3*b^6*d^10 + A^5*a*b^8*d^10 - A^5*b^9*c*d^9 + 4*A^5*a*b^8*c^2*d^8 - 9*A^5*a^2*b^7*c*d^9))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + ((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d

^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((((16*(2*A^3*b^13*d^11*f^2 - 24*A^3*a^2*b^11*d^11*f^2 - 196*A^3*a^4*b^9*d^11*f^2 - 120*A^3*a^6*b^7*d^11*f^2 + 50*A^3*a^8*b^5*d^11*f^2 + 8*A^3*b^13*c^2*d^9*f^2 - 32*A^3*a*b^12*c^3*d^8*f^2 + 208*A^3*a^3*b^10*c*d^10*f^2 + 288*A^3*a^5*b^8*c*d^10*f^2 + 80*A^3*a^7*b^6*c*d^10*f^2 - 8*A^3*a^2*b^11*c^2*d^9*f^2 + 64*A^3*a^3*b^10*c^3*d^8*f^2 - 232*A^3*a^4*b^9*c^2*d^9*f^2 + 96*A^3*a^5*b^8*c^3*d^8*f^2 - 216*A^3*a^6*b^7*c^2*d^9*f^2))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) - ((((-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*((16*(16*A*a*b^16*d^12*f^4 - 16*A*b^17*c*d^11*f^4 + 136*A*a^3*b^14*d^12*f^4 + 432*A*a^5*b^12*d^12*f^4 + 680*A*a^7*b^10*d^12*f^4 + 560*A*a^9*b^8*d^12*f^4 + 216*A*a^11*b^6*d^12*f^4 + 16*A*a^13*b^4*d^12*f^4 - 8*A*a^15*b^2*d^12*f^4 - 8*A*b^17*c^3*d^9*f^4 + 56*A*a*b^16*c^2*d^10*f^4 + 32*A*a*b^16*c^4*d^8*f^4 - 184*A*a^2*b^15*c*d^11*f^4 - 688*A*a^4*b^13*c*d^11*f^4 - 1240*A*a^6*b^11*c*d^11*f^4 - 1200*A*a^8*b^9*c*d^11*f^4 - 616*A*a^10*b^7*c*d^11*f^4 - 144*A*a^12*b^5*c*d^11*f^4 - 8*A*a^14*b^3*c*d^11*f^4 - 128*A*a^2*b^15*c^3*d^9*f^4 + 352*A*a^3*b^14*c^2*d^10*f^4 + 160*A*a^3*b^14*c^4*d^8*f^4 - 520*A*a^4*b^13*c^3*d^9*f^4 + 920*A*a^5*b^12*c^2*d^10*f^4 + 320*A*a^5*b^12*c^4*d^8*f^4 - 960*A*a^6*b^11*c^3*d^9*f^4 + 1280*A*a^7*b^10*c^2*d^10*f^4 + 320*A*a^7*b^10*c^4*d^8*f^4 - 920*A*a^8*b^9*c^3*d^9*f^4 + 1000*A*a^9*b^8*c^2*d^10*f^4 + 160*A*a^9*b^8*c^4*d^8*f^4 - 448*A*a^10*b^7*c^3*d^9*f^4 + 416*A*a^11*b^6*c^2*d^10*f^4 + 32*A*a^11*b^6*c^4*d^8*f^4 - 88*A*a^12*b^5*c^3*d^9*f^4 + 72*A*a^13*b^4*c^2*d^10*f^4))/(a^10*d^2*f^5 + b^10*c^2*f^5 + 4*a^2*b^8*c^2*f^5 + 6*a^4*b^6*c^2*f^5 + 4*a^6*b^4*c^2*f^5 + a^8*b^2*c^2*f^5 + a^2*b^8*d^2*f^5 + 4*a^4*b^6*d^2*f^5 + 6*a^6*b^4*d^2*f^5 + 4*a^8*b^2*d^2*f^5 - 2*a*b^9*c*d*f^5 - 2*a^9*b*c*d*f^5 - 8*a^3*b^7*c*d*f^5 - 12*a^5*b^5*c*d*f^5 - 8*a^7*b^3*c*d*f^5) + (16*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d* f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(32*a^2*b^17*d^12*f^4 + 160*a^4*b^15*d^12*f^4 + 288*a^6*b^13*d^12*f^4 + 160*a^8*b^11*d^12*f^4 - 160*a^10*b^9*d^12*f^4 - 288*a^12*b^7*d^12*f^4 - 160*a^14*b^5*d^12*f^4 - 32*a^16*b^3*d^12*f^4 + 32*b^19*c^2*d^10*f^4 + 48*b^19*c^4*d^8*f^4 + 176*a^2*b^17*c^2*d^10*f^4 + 272*a^2*b^17*c^4*d^8*f^4 - 432*a^3*b^16*c^3*d^9*f^4 + 336*a^4*b^15*c^2*d^10*f^4 + 624*a^4*b^15*c^4*d^8*f^4 - 912*a^5*b^14*c^3*d^9*f^4 + 112*a^6*b^13*c^2*d^10*f^4 + 720*a^6*b^13*c^4*d^8*f^4 - 880*a^7*b^12*c^3*d^9*f^4 - 560*a^8*b^11*c^2*d^10*f^4 + 400*a^8*b^11*c^4*d^8*f^4 - 240*a^9*b^10*c^3*d^9*f^4 - 1008*a^10*b^9*c^2*d^10*f^4 + 48*a^10*b^9*c^4*d^8*f^4 + 240*a^11*b^8*c^3*d^9*f^4 - 784*a^12*b^7*c^2*d^10*f^4 - 48*a^12*b^7*c^4*d^8*f^4 + 208*a^13*b^6*c^3*d^9*f^4 - 304*a^14*b^5*c^2*d^10*f^4 - 16*a^14*b^5*c^4*d^8*f^4 + 48*a^15*b^4*c^3*d^9*f^4 - 48*a^16*b^3*c^2*d^10*f^4 - 64*a*b^18*c*d^11*f^4 - 80*a*b^18*c^3*d^9*f^4 - 304*a^3*b^16*c*d^11*f^4 - 464*a^5*b^14*c*d^11*f^4 + 16*a^7*b^12*c*d^11*f^4 + 880*a^9*b^10*c*d^11*f^4 + 1136*a^11*b^8*c*d^11*f^4 + 656*a^13*b^6*c*d^11*f^4 + 176*a^15*b^4*c*d^11*f^4 + 16*a^17*b^2*c*d^11*f^4))/((b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)*(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2) - (16*(c + d*tan(e + f*x))^(1/2)*(36*A^2*a^3*b^12*d^11*f^2 + 316*A^2*a^5*b^10*d^11*f^2 + 552*A^2*a^7*b^8*d^11*f^2 + 256*A^2*a^9*b^6*d^11*f^2 - 12*A^2*a^11*b^4*d^11*f^2 - 4*A^2*a^13*b^2*d^11*f^2 - 20*A^2*b^15*c^3*d^8*f^2 + 8*A^2*a*b^14*d^11*f^2 + 4*A^2*b^15*c*d^10*f^2 - 52*A^2*a*b^14*c^2*d^9*f^2 + 80*A^2*a^2*b^13*c*d^10*f^2 - 156*A^2*a^4*b^11*c*d^10*f^2 - 640*A^2*a^6*b^9*c*d^10*f^2 - 500*A^2*a^8*b^7*c*d^10*f^2 - 80*A^2*a^10*b^5*c*d^10*f^2 + 12*A^2*a^12*b^3*c*d^10*f^2 + 116*A^2*a^2*b^13*c^3*d^8*f^2 - 220*A^2*a^3*b^12*c^2*d^9*f^2 + 216*A^2*a^4*b^11*c^3*d^8*f^2 - 104*A^2*a^5*b^10*c^2*d^9*f^2 + 8*A^2*a^6*b^9*c^3*d^8*f^2 + 232*A^2*a^7*b^8*c^2*d^9*f^2 - 68*A^2*a^8*b^7*c^3*d^8*f^2 + 156*A^2*a^9*b^6*c^2*d^9*f^2 + 4*A^2*a^10*b^5*c^3*d^8*f^2 - 12*A^2*a^11*b^4*c^2*d^9*f^2))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b^3*c*d*f^4))*(-(A^

```
2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 -
40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*
b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3
*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 +
3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*
c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*
f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(
1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*
c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 +
 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3
*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8
*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^
10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^5*c^2 + 1
0*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^2*a*b^6*c
*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c^3*f^2 -
4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*d^3*f^2 +
 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^10*b*c*d^
2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*d^2*f^2 +
 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^2 - 12*a^
8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*
c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 +
 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^
3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12
*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) -
 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)
 + (16*(c + d*tan(e + f*x))^(1/2)*(A^4*b^11*d^10 + 7*A^4*a^2*b^9*d^10 + 11*
A^4*a^4*b^7*d^10 - 27*A^4*a^6*b^5*d^10 - 2*A^4*b^11*c^2*d^8 + 12*A^4*a^2*b^
9*c^2*d^8 - 18*A^4*a^4*b^7*c^2*d^8 - 4*A^4*a*b^10*c*d^9 - 24*A^4*a^3*b^8*c*
d^9 + 44*A^4*a^5*b^6*c*d^9))/(a^10*d^2*f^4 + b^10*c^2*f^4 + 4*a^2*b^8*c^2*f
^4 + 6*a^4*b^6*c^2*f^4 + 4*a^6*b^4*c^2*f^4 + a^8*b^2*c^2*f^4 + a^2*b^8*d^2*
f^4 + 4*a^4*b^6*d^2*f^4 + 6*a^6*b^4*d^2*f^4 + 4*a^8*b^2*d^2*f^4 - 2*a*b^9*c
*d*f^4 - 2*a^9*b*c*d*f^4 - 8*a^3*b^7*c*d*f^4 - 12*a^5*b^5*c*d*f^4 - 8*a^7*b
^3*c*d*f^4)))/(b^9*(8*a^2*c^3*f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 +
 24*a^8*c*d^2*f^2) + b^7*(12*a^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c
^3*f^2 + 36*a^6*c*d^2*f^2) - b^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2
*a^3*d^3*f^2 + 24*a^3*c^2*d*f^2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2)
- b^6*(8*a^5*d^3*f^2 + 36*a^5*c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2
- 6*a*b^10*c^2*d*f^2 + 6*a^10*b*c*d^2*f^2)))*(-(A^2*b^7*d^2 + 16*A^2*a^2*b^
5*c^2 + 10*A^2*a^2*b^5*d^2 + 25*A^2*a^4*b^3*d^2 - 40*A^2*a^3*b^4*c*d - 8*A^
2*a*b^6*c*d)*(a^11*d^3*f^2 - b^11*c^3*f^2 - 4*a^2*b^9*c^3*f^2 - 6*a^4*b^7*c
^3*f^2 - 4*a^6*b^5*c^3*f^2 - a^8*b^3*c^3*f^2 + a^3*b^8*d^3*f^2 + 4*a^5*b^6*
d^3*f^2 + 6*a^7*b^4*d^3*f^2 + 4*a^9*b^2*d^3*f^2 + 3*a*b^10*c^2*d*f^2 - 3*a^
10*b*c*d^2*f^2 - 3*a^2*b^9*c*d^2*f^2 + 12*a^3*b^8*c^2*d*f^2 - 12*a^4*b^7*c*
d^2*f^2 + 18*a^5*b^6*c^2*d*f^2 - 18*a^6*b^5*c*d^2*f^2 + 12*a^7*b^4*c^2*d*f^
2 - 12*a^8*b^3*c*d^2*f^2 + 3*a^9*b^2*c^2*d*f^2))^(1/2)*2i)/(b^9*(8*a^2*c^3*
```

```
f^2 + 6*a^2*c*d^2*f^2) + b^3*(2*a^8*c^3*f^2 + 24*a^8*c*d^2*f^2) + b^7*(12*a
^4*c^3*f^2 + 24*a^4*c*d^2*f^2) + b^5*(8*a^6*c^3*f^2 + 36*a^6*c*d^2*f^2) - b
^2*(8*a^9*d^3*f^2 + 6*a^9*c^2*d*f^2) - b^8*(2*a^3*d^3*f^2 + 24*a^3*c^2*d*f^
2) - b^4*(12*a^7*d^3*f^2 + 24*a^7*c^2*d*f^2) - b^6*(8*a^5*d^3*f^2 + 36*a^5*
c^2*d*f^2) - 2*a^11*d^3*f^2 + 2*b^11*c^3*f^2 - 6*a*b^10*c^2*d*f^2 + 6*a^10*
b*c*d^2*f^2) + (A*b^2*d*(c + d*tan(e + f*x))^(1/2))/((b*f*(c + d*tan(e + f*
x)) + a*d*f - b*c*f)*(a^3*d - b^3*c - a^2*b*c + a*b^2*d)) + (C*a^2*d*(c + d
*tan(e + f*x))^(1/2))/((b*f*(c + d*tan(e + f*x)) + a*d*f - b*c*f)*(a^3*d -
b^3*c - a^2*b*c + a*b^2*d)) - (B*a*b*d*(c + d*tan(e + f*x))^(1/2))/((b*f*(c
 + d*tan(e + f*x)) + a*d*f - b*c*f)*(a^3*d - b^3*c - a^2*b*c + a*b^2*d))
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)^2\sqrt{c + d\tan\left(e + fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(1/2)/(a+b*tan
(f*x+e))**2,x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**2*
sqrt(c + d*tan(e + f*x))), x)
```

3.116 $\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=511

$$\frac{2b\sqrt{c+d\tan(e+fx)}\left(6a^2d^2\left(d^2(5A+7C)-5Bcd+12c^2C\right)-15abd\left(cd^2(3A+5C)-6Bc^2d-3Bd^3+8c^3C\right)+\right.}{15d^4f\left(c^2+d^2\right)}$$

```
[Out] -(a-I*b)^3*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d
)^(3/2)/f-(I*a-b)^3*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))
/(c+I*d)^(3/2)/f+2/15*b*(6*a^2*d^2*(12*c^2*C-5*B*c*d+(5*A+7*C)*d^2)-15*a*b*
d*(8*c^3*C-6*B*c^2*d+c*(3*A+5*C)*d^2-3*B*d^3)+b^2*(48*c^4*C-40*B*c^3*d+6*c^
2*(5*A+3*C)*d^2-25*B*c*d^3+15*(A-C)*d^4))*(c+d*tan(f*x+e))^(1/2)/d^4/(c^2+d
^2)/f-2/15*b^2*(4*(-a*d+b*c)*(6*c^2*C-5*B*c*d+(5*A+C)*d^2)-5*d^2*((A-C)*(-a
*d+b*c)+B*(a*c+b*d)))*(c+d*tan(f*x+e))^(1/2)*tan(f*x+e)/d^3/(c^2+d^2)/f+2/5
*b*(6*c^2*C-5*B*c*d+(5*A+C)*d^2)*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^2/
d^2/(c^2+d^2)/f-2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^3/d/(c^2+d^2)/f/(c+d
*tan(f*x+e))^(1/2)
```

Rubi [A] time = 2.46, antiderivative size = 511, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 8, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.170, Rules used = {3645, 3647, 3637, 3630, 3539, 3537, 63, 208}

$$\frac{2b\sqrt{c+d\tan(e+fx)}\left(6a^2d^2\left(d^2(5A+7C)-5Bcd+12c^2C\right)-15abd\left(cd^2(3A+5C)-6Bc^2d-3Bd^3+8c^3C\right)+\right.}{15d^4f\left(c^2+d^2\right)}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((a - I*b)^3*(I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/((c - I*d)^(3/2)*f)) - ((I*a - b)^3*(A + I*B - C)*ArcTanh[Sqrt[c + d*T
an[e + f*x]]/Sqrt[c + I*d]])/((c + I*d)^(3/2)*f) - (2*(c^2*C - B*c*d + A*d^
2)*(a + b*Tan[e + f*x])^3)/(d*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]]) + (2*
b*(6*a^2*d^2*(12*c^2*C - 5*B*c*d + (5*A + 7*C)*d^2) - 15*a*b*d*(8*c^3*C - 6
*B*c^2*d + c*(3*A + 5*C)*d^2 - 3*B*d^3) + b^2*(48*c^4*C - 40*B*c^3*d + 6*c^
2*(5*A + 3*C)*d^2 - 25*B*c*d^3 + 15*(A - C)*d^4))*Sqrt[c + d*Tan[e + f*x]])
/(15*d^4*(c^2 + d^2)*f) - (2*b^2*(4*(b*c - a*d)*(6*c^2*C - 5*B*c*d + (5*A +
 C)*d^2) - 5*d^2*((A - C)*(b*c - a*d) + B*(a*c + b*d)))*Tan[e + f*x]*Sqrt[c
 + d*Tan[e + f*x]])/(15*d^3*(c^2 + d^2)*f) + (2*b*(6*c^2*C - 5*B*c*d + (5*A
 + C)*d^2)*(a + b*Tan[e + f*x])^2*Sqrt[c + d*Tan[e + f*x]])/(5*d^2*(c^2 + d
^2)*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
```

```
!LtQ[n, -1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}-$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$=-\frac{(a-ib)^3(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right.}{(c-id)^{3/2}f}$$

Mathematica [C] time = 6.79, size = 920, normalized size = 1.80

$$\frac{2C(a+b\tan(e+fx))^3}{5df\sqrt{c+d\tan(e+fx)}}+\frac{2\left(\frac{(-6bcC+6adC+5bBd)(a+b\tan(e+fx))^2}{3df\sqrt{c+d\tan(e+fx)}}+\frac{2\left(\frac{\left(15b(Ab-Cb+aB)d^2+4(bc-ad)(6bcC-6adC-5bBd)\right)(a+b\tan(e+fx))}{2df\sqrt{c+d\tan(e+fx)}}+\frac{2\left(\frac{1}{2}(-1\ldots\right)}{\ldots}\right)}{\ldots}\right)}{\ldots}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^3)/(5*d*f*Sqrt[c + d*Tan[e + f*x]]) + (2*(((-6*b*
c*C + 5*b*B*d + 6*a*C*d)*(a + b*Tan[e + f*x])^2)/(3*d*f*Sqrt[c + d*Tan[e +
f*x]]) + (2*(((15*b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(6*b*c*C - 5*b*B*
d - 6*a*C*d))*(a + b*Tan[e + f*x]))/(2*d*f*Sqrt[c + d*Tan[e + f*x]]) + ((-2
*(-48*b^3*c^3*C + 40*b^3*B*c^2*d + 144*a*b^2*c^2*C*d - 30*A*b^3*c*d^2 - 110
*a*b^2*B*c*d^2 - 144*a^2*b*c*C*d^2 + 30*b^3*c*C*d^2 + 60*a*A*b^2*d^3 + 85*a
^2*b*B*d^3 - 15*b^3*B*d^3 + 48*a^3*C*d^3 - 60*a*b^2*C*d^3))/(d*Sqrt[c + d*T
an[e + f*x]]) + (2*(((45*a^2*A*b*d^3 - 15*A*b^3*d^3 + 15*a^3*B*d^3 - 45*a*b
^2*B*d^3 - 45*a^2*b*C*d^3 + 15*b^3*C*d^3)*(((-I)*ArcTanh[Sqrt[c + d*Tan[e +
 f*x]]/Sqrt[c - I*d]])/Sqrt[c - I*d] + (I*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/
Sqrt[c + I*d]])/Sqrt[c + I*d]))/2 + ((-1/2*(c*d*(45*a^2*A*b*d^3 - 15*A*b^3*
d^3 + 15*a^3*B*d^3 - 45*a*b^2*B*d^3 - 45*a^2*b*C*d^3 + 15*b^3*C*d^3)) + d^2
```

```
*((-48*b^3*c^3*C + 40*b^3*B*c^2*d + 144*a*b^2*c^2*C*d - 30*A*b^3*c*d^2 - 11
0*a*b^2*B*c*d^2 - 144*a^2*b*c*C*d^2 + 30*b^3*c*C*d^2 + 15*a^3*A*d^3 + 15*a*
A*b^2*d^3 + 40*a^2*b*B*d^3 + 33*a^3*C*d^3 - 15*a*b^2*C*d^3)/2 + (48*b^3*c^3
*C - 40*b^3*B*c^2*d - 144*a*b^2*c^2*C*d + 30*A*b^3*c*d^2 + 110*a*b^2*B*c*d^
2 + 144*a^2*b*c*C*d^2 - 30*b^3*c*C*d^2 - 60*a*A*b^2*d^3 - 85*a^2*b*B*d^3 +
15*b^3*B*d^3 - 48*a^3*C*d^3 + 60*a*b^2*C*d^3)/2))*(-(Hypergeometric2F1[-1/2
, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)]/((I*c + d)*Sqrt[c + d*Tan[e + f*x
]])) + Hypergeometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)]/((I*
c - d)*Sqrt[c + d*Tan[e + f*x]])))/d))/d)/(4*d*f)))/(3*d)))/(5*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.58, size = 49725, normalized size = 97.31

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/
2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(3/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a + b\tan\left(e + fx\right)\right)^{3} \left(A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)\right)}{\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(3/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**3*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x))**(3/2), x)
```

3.117 $\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=343

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)(a+b\tan(e+fx))^2}{df\left(c^2+d^2\right)\sqrt{c+d\tan(e+fx)}}+\frac{2b\sqrt{c+d\tan(e+fx)}\left(6ad\left(d^2(A+C)-Bcd+2c^2C\right)-b\left(cd^2(3A+\right.\right.}{3d^3f\left(c^2+d^2\right)}$$

```
[Out] -(a-I*b)^2*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d
)^(3/2)/f-(a+I*b)^2*(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/(c+I*d)^(3/2)/f+2/3*b*(6*a*d*(2*c^2*C-B*c*d+(A+C)*d^2)-b*(8*c^3*C-6*B*c^
2*d+c*(3*A+5*C)*d^2-3*B*d^3))*(c+d*tan(f*x+e))^(1/2)/d^3/(c^2+d^2)/f+2/3*b^
2*(4*c^2*C-3*B*c*d+(3*A+C)*d^2)*(c+d*tan(f*x+e))^(1/2)*tan(f*x+e)/d^2/(c^2+
d^2)/f-2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^2/d/(c^2+d^2)/f/(c+d*tan(f*x+
e))^(1/2)
```

Rubi [A] time = 1.35, antiderivative size = 343, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3645, 3637, 3630, 3539, 3537, 63, 208}

$$\frac{2b\sqrt{c+d\tan(e+fx)}\left(6ad\left(d^2(A+C)-Bcd+2c^2C\right)-b\left(cd^2(3A+5C)-6Bc^2d-3Bd^3+8c^3C\right)\right)}{3d^3f\left(c^2+d^2\right)}-\frac{2\left(Ad^2-Bc\right.}{df\left(c^2\right.}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((a - I*b)^2*(I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/((c - I*d)^(3/2)*f)) - ((a + I*b)^2*(B - I*(A - C))*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((c + I*d)^(3/2)*f) - (2*(c^2*C - B*c*d + A*
d^2)*(a + b*Tan[e + f*x])^2)/(d*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]]) + (
2*b*(6*a*d*(2*c^2*C - B*c*d + (A + C)*d^2) - b*(8*c^3*C - 6*B*c^2*d + c*(3*
A + 5*C)*d^2 - 3*B*d^3))*Sqrt[c + d*Tan[e + f*x]])/(3*d^3*(c^2 + d^2)*f) +
(2*b^2*(4*c^2*C - 3*B*c*d + (3*A + C)*d^2)*Tan[e + f*x]*Sqrt[c + d*Tan[e +
f*x]])/(3*d^2*(c^2 + d^2)*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
```

```
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{(a-ib)^2(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right.}{(c-id)^{3/2}f}$$

Mathematica [C] time = 6.51, size = 476, normalized size = 1.39

$$\frac{2C(a+b\tan(e+fx))^2}{3df\sqrt{c+d\tan(e+fx)}}+\frac{2\left(\frac{(4aCd+3bBd-4bcC)(a+b\tan(e+fx))}{df\sqrt{c+d\tan(e+fx)}}+\frac{-\frac{2(8a^2Cd^2+9abBd^2-16abcCd+3Ab^2d^2-6b^2Bcd+8b^2c^2C-3b^2Cd^2)}{d\sqrt{c+d\tan(e+fx)}}+\frac{2\left(\frac{3}{2}d^2(a^2B+\cdots\right.}{\cdots}}{\cdots}\right.}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^2)/(3*d*f*Sqrt[c + d*Tan[e + f*x]]) + (2*(((-4*b*
c*C + 3*b*B*d + 4*a*C*d)*(a + b*Tan[e + f*x]))/(d*f*Sqrt[c + d*Tan[e + f*x]
]) + ((-2*(8*b^2*c^2*C - 6*b^2*B*c*d - 16*a*b*c*C*d + 3*A*b^2*d^2 + 9*a*b*B
*d^2 + 8*a^2*C*d^2 - 3*b^2*C*d^2))/(d*Sqrt[c + d*Tan[e + f*x]]) + (2*((3*(a
^2*B - b^2*B + 2*a*b*(A - C))*d^2*(((-I)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/S
qrt[c - I*d]])/Sqrt[c - I*d] + (I*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c +
 I*d]])/Sqrt[c + I*d]))/2 + (((-3*c*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/2
- (3*(2*a*b*B - a^2*(A - C) + b^2*(A - C))*d^4)/2)*(-(Hypergeometric2F1[-1/
2, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)]/((I*c + d)*Sqrt[c + d*Tan[e + f*
x]])) + Hypergeometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)]/((I
*c - d)*Sqrt[c + d*Tan[e + f*x]])))/d))/d)/(2*d*f)))/(3*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.46, size = 36710, normalized size = 107.03

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/
2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [B] time = 66.25, size = 54886, normalized size = 160.02

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(3/2),x)
```

```
[Out] (2*(B*b^2*c^3 + B*a^2*c*d^2 - 2*B*a*b*c^2*d))/(d^2*f*(c^2 + d^2)*(c + d*tan
(e + f*x))^(1/2)) - atan((((-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*
B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*
a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*
b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 4
8*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*
a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 2
4*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^
2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^
3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^
```

4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*a^2*d^12*f^4 + 32*B*b^2*d^12*f^4 - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i - ((-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(32*B*b^2*d^12*f^4 - 32*B*a^2*d^12*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2 *b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i)/(64*B^3*a^3*b^3*d^9*f^2 - ((-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(32*B*b^2*d^12*f^4 - 32*B*a^2*d^12*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4* a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f

^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8
*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*a^2*d^12
*f^4 + 32*B*b^2*d^12*f^4 - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 6
4*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b
^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^
8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^
9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f
^4) + (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3
 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^
3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^
3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*
f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3
*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*
B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d
^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^
3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^
2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 +
96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*
d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b
^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*
c^3*f^2 + 24*B^2*a^2*b^2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*
f^2 + 12*B^2*a^4*c*d^2*f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2
- 48*B^2*a^3*b*c^2*d*f^2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4
 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) + 48*B^3*a^6*c^3*d^6*f^2 + 48*B^3
*a^6*c^5*d^4*f^2 + 16*B^3*a^6*c^7*d^2*f^2 - 48*B^3*b^6*c^3*d^6*f^2 - 48*B^3
*b^6*c^5*d^4*f^2 - 16*B^3*b^6*c^7*d^2*f^2 + 32*B^3*a*b^5*d^9*f^2 + 32*B^3*a
^5*b*d^9*f^2 + 16*B^3*a^6*c*d^8*f^2 - 16*B^3*b^6*c*d^8*f^2 + 96*B^3*a*b^5*c
^2*d^7*f^2 + 96*B^3*a*b^5*c^4*d^5*f^2 + 32*B^3*a*b^5*c^6*d^3*f^2 - 16*B^3*a
^2*b^4*c*d^8*f^2 + 16*B^3*a^4*b^2*c*d^8*f^2 + 96*B^3*a^5*b*c^2*d^7*f^2 + 96
*B^3*a^5*b*c^4*d^5*f^2 + 32*B^3*a^5*b*c^6*d^3*f^2 - 48*B^3*a^2*b^4*c^3*d^6*
f^2 - 48*B^3*a^2*b^4*c^5*d^4*f^2 - 16*B^3*a^2*b^4*c^7*d^2*f^2 + 192*B^3*a^3
*b^3*c^2*d^7*f^2 + 192*B^3*a^3*b^3*c^4*d^5*f^2 + 64*B^3*a^3*b^3*c^6*d^3*f^2
 + 48*B^3*a^4*b^2*c^3*d^6*f^2 + 48*B^3*a^4*b^2*c^5*d^4*f^2 + 16*B^3*a^4*b^2
*c^7*d^2*f^2))*(-(((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*
c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^
2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2
+ 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^
4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*
B^4*a^6*b^2))^(1/2) - 4*B^2*a^4*c^3*f^2 - 4*B^2*b^4*c^3*f^2 + 24*B^2*a^2*b^
2*c^3*f^2 - 16*B^2*a*b^3*d^3*f^2 + 16*B^2*a^3*b*d^3*f^2 + 12*B^2*a^4*c*d^2*
f^2 + 12*B^2*b^4*c*d^2*f^2 + 48*B^2*a*b^3*c^2*d*f^2 - 48*B^2*a^3*b*c^2*d*f^
2 - 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^
4*d^2*f^4)))^(1/2)*2i - atan(((((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 -
48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B
^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a

```
^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4
+ 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B
^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2
- 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12
*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2
*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2
*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4
*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^
2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96
*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^
2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8
+ B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a
^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*
f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 -
48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)
/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12
*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f
^5 + 64*c^11*d^2*f^5) - 32*B*a^2*d^12*f^4 + 32*B*b^2*d^12*f^4 - 96*B*a^2*c^
2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4
*f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4
 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 12
8*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*
a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*B
^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^
4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^
4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a
*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*
B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f
^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^
2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*
((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2
*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c
*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^
2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f
^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1
/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B
^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4
*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b
^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/
2)*1i - (((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2
 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*
B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B
^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*
c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6
*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f
```

^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(32*B*b^2*d^12*f^4 - 32*B*a^2*d^12*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i)/(64*B^3*a^3*b^3*d^9*f^2 - (((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^

2*f^4)))^(1/2)*(32*B*b^2*d^12*f^4 - 32*B*a^2*d^12*f^4 - (c + d*tan(e + f*x)
)^(1/2)*((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2
+ 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B
^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^
2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c
^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*
b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^
2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12
*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B
^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^
4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^
6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*B*a^2*c^2*d^10*f^4 - 64*B*a
^2*c^4*d^8*f^4 + 64*B*a^2*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^1
0*d^2*f^4 + 96*B*b^2*c^2*d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6
*f^4 - 96*B*b^2*c^8*d^4*f^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4
+ 512*B*a*b*c^3*d^9*f^4 + 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 1
28*B*a*b*c^9*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 1
6*B^2*b^4*d^10*f^3 - 96*B^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*
B^2*a^4*c^6*d^4*f^3 - 16*B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*
B^2*b^4*c^6*d^4*f^3 - 16*B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 12
8*B^2*a^3*b*c*d^9*f^3 + 384*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f
^3 + 128*B^2*a*b^3*c^7*d^3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*
c^5*d^5*f^3 - 128*B^2*a^3*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192
*B^2*a^2*b^2*c^6*d^4*f^3 + 96*B^2*a^2*b^2*c^8*d^2*f^3))*((((8*B^2*a^4*c^3*f
^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32
*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a
*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 -
(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*
b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3
*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 -
16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2
*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(
c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - (((((8*B^2*a^4
*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^
2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96
*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^
2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8
+ B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a
^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*
f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 -
48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)
/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*ta
n(e + f*x))^(1/2)*((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^
2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*
f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^

```
2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*
f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 +
4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*
b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^
2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*
f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*
c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 +
640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*a^2*d^12*f^4 +
32*B*b^2*d^12*f^4 - 96*B*a^2*c^2*d^10*f^4 - 64*B*a^2*c^4*d^8*f^4 + 64*B*a^2
*c^6*d^6*f^4 + 96*B*a^2*c^8*d^4*f^4 + 32*B*a^2*c^10*d^2*f^4 + 96*B*b^2*c^2*
d^10*f^4 + 64*B*b^2*c^4*d^8*f^4 - 64*B*b^2*c^6*d^6*f^4 - 96*B*b^2*c^8*d^4*f
^4 - 32*B*b^2*c^10*d^2*f^4 + 128*B*a*b*c*d^11*f^4 + 512*B*a*b*c^3*d^9*f^4 +
 768*B*a*b*c^5*d^7*f^4 + 512*B*a*b*c^7*d^5*f^4 + 128*B*a*b*c^9*d^3*f^4) + (
c + d*tan(e + f*x))^(1/2)*(16*B^2*a^4*d^10*f^3 + 16*B^2*b^4*d^10*f^3 - 96*B
^2*a^2*b^2*d^10*f^3 + 32*B^2*a^4*c^2*d^8*f^3 - 32*B^2*a^4*c^6*d^4*f^3 - 16*
B^2*a^4*c^8*d^2*f^3 + 32*B^2*b^4*c^2*d^8*f^3 - 32*B^2*b^4*c^6*d^4*f^3 - 16*
B^2*b^4*c^8*d^2*f^3 + 128*B^2*a*b^3*c*d^9*f^3 - 128*B^2*a^3*b*c*d^9*f^3 + 3
84*B^2*a*b^3*c^3*d^7*f^3 + 384*B^2*a*b^3*c^5*d^5*f^3 + 128*B^2*a*b^3*c^7*d^
3*f^3 - 384*B^2*a^3*b*c^3*d^7*f^3 - 384*B^2*a^3*b*c^5*d^5*f^3 - 128*B^2*a^3
*b*c^7*d^3*f^3 - 192*B^2*a^2*b^2*c^2*d^8*f^3 + 192*B^2*a^2*b^2*c^6*d^4*f^3
+ 96*B^2*a^2*b^2*c^8*d^2*f^3))*((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 -
48*B^2*a^2*b^2*c^3*f^2 + 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B
^2*a^4*c*d^2*f^2 - 24*B^2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a
^3*b*c^2*d*f^2 + 144*B^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4
+ 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B
^4*a^4*b^4 + 4*B^4*a^6*b^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2
- 24*B^2*a^2*b^2*c^3*f^2 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12
*B^2*a^4*c*d^2*f^2 - 12*B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2
*a^3*b*c^2*d*f^2 + 72*B^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2
*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) + 48*B^3*a^6*c^3*d^6*f^2 + 48*B^3*a^6*c^5
*d^4*f^2 + 16*B^3*a^6*c^7*d^2*f^2 - 48*B^3*b^6*c^3*d^6*f^2 - 48*B^3*b^6*c^5
*d^4*f^2 - 16*B^3*b^6*c^7*d^2*f^2 + 32*B^3*a*b^5*d^9*f^2 + 32*B^3*a^5*b*d^9
*f^2 + 16*B^3*a^6*c*d^8*f^2 - 16*B^3*b^6*c*d^8*f^2 + 96*B^3*a*b^5*c^2*d^7*f
^2 + 96*B^3*a*b^5*c^4*d^5*f^2 + 32*B^3*a*b^5*c^6*d^3*f^2 - 16*B^3*a^2*b^4*c
*d^8*f^2 + 16*B^3*a^4*b^2*c*d^8*f^2 + 96*B^3*a^5*b*c^2*d^7*f^2 + 96*B^3*a^5
*b*c^4*d^5*f^2 + 32*B^3*a^5*b*c^6*d^3*f^2 - 48*B^3*a^2*b^4*c^3*d^6*f^2 - 48
*B^3*a^2*b^4*c^5*d^4*f^2 - 16*B^3*a^2*b^4*c^7*d^2*f^2 + 192*B^3*a^3*b^3*c^2
*d^7*f^2 + 192*B^3*a^3*b^3*c^4*d^5*f^2 + 64*B^3*a^3*b^3*c^6*d^3*f^2 + 48*B^
3*a^4*b^2*c^3*d^6*f^2 + 48*B^3*a^4*b^2*c^5*d^4*f^2 + 16*B^3*a^4*b^2*c^7*d^2
*f^2))*((((8*B^2*a^4*c^3*f^2 + 8*B^2*b^4*c^3*f^2 - 48*B^2*a^2*b^2*c^3*f^2 +
 32*B^2*a*b^3*d^3*f^2 - 32*B^2*a^3*b*d^3*f^2 - 24*B^2*a^4*c*d^2*f^2 - 24*B^
2*b^4*c*d^2*f^2 - 96*B^2*a*b^3*c^2*d*f^2 + 96*B^2*a^3*b*c^2*d*f^2 + 144*B^2
*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^
4*d^2*f^4)*(B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b
^2))^(1/2) + 4*B^2*a^4*c^3*f^2 + 4*B^2*b^4*c^3*f^2 - 24*B^2*a^2*b^2*c^3*f^2
```

```
 + 16*B^2*a*b^3*d^3*f^2 - 16*B^2*a^3*b*d^3*f^2 - 12*B^2*a^4*c*d^2*f^2 - 12*
B^2*b^4*c*d^2*f^2 - 48*B^2*a*b^3*c^2*d*f^2 + 48*B^2*a^3*b*c^2*d*f^2 + 72*B^
2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4
)))^(1/2)*2i - atan((((c + d*tan(e + f*x))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*
C^2*b^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^
2*a^4*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^
2*b^4*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*
C^2*a^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3
 + 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^
5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C
^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2*f^3) - ((((8*C^2*a^4*c^3*f^
2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*
C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*
b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (
16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b
^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*
f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 1
6*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*
a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c
^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f
*x))^(1/2)*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f
^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 2
4*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144
*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 4
8*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a
^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3
*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 +
 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 7
2*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2
*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7
*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*C*a^2*c*d^11*f^4 + 64*C*
b^2*c*d^11*f^4 - 256*C*a^2*c^3*d^9*f^4 - 384*C*a^2*c^5*d^7*f^4 - 256*C*a^2*
c^7*d^5*f^4 - 64*C*a^2*c^9*d^3*f^4 + 256*C*b^2*c^3*d^9*f^4 + 384*C*b^2*c^5*
d^7*f^4 + 256*C*b^2*c^7*d^5*f^4 + 64*C*b^2*c^9*d^3*f^4 - 64*C*a*b*d^12*f^4
 - 192*C*a*b*c^2*d^10*f^4 - 128*C*a*b*c^4*d^8*f^4 + 128*C*a*b*c^6*d^6*f^4 +
192*C*a*b*c^8*d^4*f^4 + 64*C*a*b*c^10*d^2*f^4))*((((8*C^2*a^4*c^3*f^2 + 8*C
^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3
*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2
*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*
f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*
C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4
*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a
^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c
^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4
+ d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x
```

))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f
^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f
^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f
^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*
d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a
^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 -
192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*
c^8*d^2*f^3) - ((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c
^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2
 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 +
 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4
 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C
^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2
*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f
^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2
 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4
*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c^3*f^2 + 8*C^2
*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b
*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d
*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^
4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^
4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C
^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3
*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2
*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 +
d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d
^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2
*f^5) + 64*C*a^2*c*d^11*f^4 - 64*C*b^2*c*d^11*f^4 + 256*C*a^2*c^3*d^9*f^4 +
 384*C*a^2*c^5*d^7*f^4 + 256*C*a^2*c^7*d^5*f^4 + 64*C*a^2*c^9*d^3*f^4 - 256
*C*b^2*c^3*d^9*f^4 - 384*C*b^2*c^5*d^7*f^4 - 256*C*b^2*c^7*d^5*f^4 - 64*C*b
^2*c^9*d^3*f^4 + 64*C*a*b*d^12*f^4 + 192*C*a*b*c^2*d^10*f^4 + 128*C*a*b*c^4
*d^8*f^4 - 128*C*a*b*c^6*d^6*f^4 - 192*C*a*b*c^8*d^4*f^4 - 64*C*a*b*c^10*d^
2*f^4))*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2
+ 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C
^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^
2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c
^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*
b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^
2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12
*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C
^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^
4)))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b
^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4
*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4
*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a

^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2*f^3) - ((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*C*a^2*c*d^11*f^4 + 64*C*b^2*c*d^11*f^4 - 256*C*a^2*c^3*d^9*f^4 - 384*C*a^2*c^5*d^7*f^4 - 256*C*a^2*c^7*d^5*f^4 - 64*C*a^2*c^9*d^3*f^4 + 256*C*b^2*c^3*d^9*f^4 + 384*C*b^2*c^5*d^7*f^4 + 256*C*b^2*c^7*d^5*f^4 + 64*C*b^2*c^9*d^3*f^4 - 64*C*a*b*d^12*f^4 - 192*C*a*b*c^2*d^10*f^4 - 128*C*a*b*c^4*d^8*f^4 + 128*C*a*b*c^6*d^6*f^4 + 192*C*a*b*c^8*d^4*f^4 + 64*C*a*b*c^10*d^2*f^4))*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2* f^3) - ((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 64*C*a^2*c*d^11*f^4 - 64*C*b^2*c*d^11*f^4 + 256*C*a^2*c^3*d^9*f^4 + 384*C*a^2*c^5*d^7*f^4 + 256*C*a^2*c^7*d^5*f^4 + 64*C*a^2*c^9*d^3*f^4 - 256*C*b^2*c^3*d^9*f^4 - 384*C*b^2*c^5*d^7*f^4 - 256*C*b^2*c^7*d^5*f^4 - 64*C*b^2*c^9*d^3*f^4 + 64*C*a*b*d^12*f^4 + 192*C*a*b*c^2*d^10*f^4 + 128*C*a*b*c^4*d^8*f^4 - 128*C*a*b*c^6*d^6*f^4 - 192*C*a*b*c^8*d^4*f^4 - 64*C*a*b*c^10*d^2*f^4))*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - 16*C^3*a^6*d^9*f^2 + 16*C^3*b^6*d^9*f^2 + 16*C^3*a^2*b^4*d^9*f^2 - 16*C^3*a^4*b^2*d^9*f^2 - 48*C^3*a^6*c^2*d^7*f^2 - 48*C^3*a^6*c^4*d^5*f^2 - 16*C^3*a^6*c^6*d^3*f^2 + 48*C^3*b^6*c^2*d^7*f^2 + 48*C^3*b^6*c^4*d^5*f^2 + 16*C^3*b^6*c^6*d^3*f^2 + 32*C^3*a*b^5*c*d^8*f^2 + 32*C^3*a^5*b*c*d^8*f^2 + 96*C^3*a*b^5*c^3*d^6*f^2 + 96*C^3*a*b^5*c^5*d^4*f^2 + 32*C^3*a*b^5*c^7*d^2*f^2 + 64*C^3*a^3*b^3*c*d^8*f^2 + 96*C^3*a^5*b*c^3*d^6*f^2 + 96*C^3*a^5*b*c^5*d^4*f^2 + 32*C^3*a^5*b*c^7*d^2*f^2 + 48*C^3*a^2*b^4*c^2*d^7*f^2 + 48*C^3*a^2*b^4*c^4*d^5*f^2 + 16*C^3*a^2*b^4*c^6*d^3*f^2 + 192*C^3*a^3*b^3*c^3*d^6*f^2 + 192*C^3*a^3*b^3*c^5*d^4*f^2 + 64*C^3*a^3*b^3*c^7*d^2*f^2 - 48*C^3*a^4*b^2*c^2*d^7*f^2 - 48*C^3*a^4*b^2*c^4*d^5*f^2 - 16*C^3*a^4*b^2*c^6*d^3*f^2))*((((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*

a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*
d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2
*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^
4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/
2) - 4*C^2*a^4*c^3*f^2 - 4*C^2*b^4*c^3*f^2 + 24*C^2*a^2*b^2*c^3*f^2 - 16*C^
2*a*b^3*d^3*f^2 + 16*C^2*a^3*b*d^3*f^2 + 12*C^2*a^4*c*d^2*f^2 + 12*C^2*b^4*
c*d^2*f^2 + 48*C^2*a*b^3*c^2*d*f^2 - 48*C^2*a^3*b*c^2*d*f^2 - 72*C^2*a^2*b^
2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2
)*2i - atan((((c + d*tan(e + f*x))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4*
d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c^
6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c^
6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3*
b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C
^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^
3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b
^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2*f^3) - (-(((8*C^2*a^4*c^3*f^2 + 8*C
^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3
*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2
*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*
f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*
C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4
*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a
^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c
^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4
+ d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1
/2)*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 3
2*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*
b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a
^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*
d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2
))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 +
16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^
2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*
a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
)^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f
^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*C*a^2*c*d^11*f^4 + 64*C*b^2*c*
d^11*f^4 - 256*C*a^2*c^3*d^9*f^4 - 384*C*a^2*c^5*d^7*f^4 - 256*C*a^2*c^7*d^
5*f^4 - 64*C*a^2*c^9*d^3*f^4 + 256*C*b^2*c^3*d^9*f^4 + 384*C*b^2*c^5*d^7*f^
4 + 256*C*b^2*c^7*d^5*f^4 + 64*C*b^2*c^9*d^3*f^4 - 64*C*a*b*d^12*f^4 - 192*
C*a*b*c^2*d^10*f^4 - 128*C*a*b*c^4*d^8*f^4 + 128*C*a*b*c^6*d^6*f^4 + 192*C*
a*b*c^8*d^4*f^4 + 64*C*a*b*c^10*d^2*f^4))*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^
4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^
3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^
2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 +
16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a

^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*
b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*
d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*
f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6
*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1
/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 +
32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 +
32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 +
128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f
^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*
c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C
^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d
^2*f^3) - (-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f
^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 2
4*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144
*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 4
8*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a
^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3
*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 -
12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 7
2*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2
f^4)))^(1/2)((c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^
4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^
3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^
2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 +
16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a
^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*
b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*
d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*
f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6
*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10
*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^
5) + 64*C*a^2*c*d^11*f^4 - 64*C*b^2*c*d^11*f^4 + 256*C*a^2*c^3*d^9*f^4 + 38
4*C*a^2*c^5*d^7*f^4 + 256*C*a^2*c^7*d^5*f^4 + 64*C*a^2*c^9*d^3*f^4 - 256*C*
b^2*c^3*d^9*f^4 - 384*C*b^2*c^5*d^7*f^4 - 256*C*b^2*c^7*d^5*f^4 - 64*C*b^2*
c^9*d^3*f^4 + 64*C*a*b*d^12*f^4 + 192*C*a*b*c^2*d^10*f^4 + 128*C*a*b*c^4*d^
8*f^4 - 128*C*a*b*c^6*d^6*f^4 - 192*C*a*b*c^8*d^4*f^4 - 64*C*a*b*c^10*d^2*f
^4))*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 +
32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2
*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*
a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4
*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^
2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2
+ 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C
^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2

*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)
))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4
*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c
^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c
^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 128*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3
*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*
C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f
^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*
b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2*f^3) - (-(((8*C^2*a^4*c^3*f^2 + 8*
C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^
3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^
2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6
*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4
*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 +
4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*
a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*
c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4
 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(
1/2)*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 +
32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2
*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*
a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4
*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^
2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2
+ 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C
^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2
*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)
))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*
f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*C*a^2*c*d^11*f^4 + 64*C*b^2*c
*d^11*f^4 - 256*C*a^2*c^3*d^9*f^4 - 384*C*a^2*c^5*d^7*f^4 - 256*C*a^2*c^7*d
^5*f^4 - 64*C*a^2*c^9*d^3*f^4 + 256*C*b^2*c^3*d^9*f^4 + 384*C*b^2*c^5*d^7*f
^4 + 256*C*b^2*c^7*d^5*f^4 + 64*C*b^2*c^9*d^3*f^4 - 64*C*a*b*d^12*f^4 - 192
*C*a*b*c^2*d^10*f^4 - 128*C*a*b*c^4*d^8*f^4 + 128*C*a*b*c^6*d^6*f^4 + 192*C
*a*b*c^8*d^4*f^4 + 64*C*a*b*c^10*d^2*f^4))*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b
^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d
^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f
^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4
+ 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*
a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2
*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b
*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d
*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^
6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2
)*(16*C^2*a^4*d^10*f^3 + 16*C^2*b^4*d^10*f^3 - 96*C^2*a^2*b^2*d^10*f^3 + 32
*C^2*a^4*c^2*d^8*f^3 - 32*C^2*a^4*c^6*d^4*f^3 - 16*C^2*a^4*c^8*d^2*f^3 + 32

*C^2*b^4*c^2*d^8*f^3 - 32*C^2*b^4*c^6*d^4*f^3 - 16*C^2*b^4*c^8*d^2*f^3 + 12
8*C^2*a*b^3*c*d^9*f^3 - 128*C^2*a^3*b*c*d^9*f^3 + 384*C^2*a*b^3*c^3*d^7*f^3
 + 384*C^2*a*b^3*c^5*d^5*f^3 + 128*C^2*a*b^3*c^7*d^3*f^3 - 384*C^2*a^3*b*c^
3*d^7*f^3 - 384*C^2*a^3*b*c^5*d^5*f^3 - 128*C^2*a^3*b*c^7*d^3*f^3 - 192*C^2
*a^2*b^2*c^2*d^8*f^3 + 192*C^2*a^2*b^2*c^6*d^4*f^3 + 96*C^2*a^2*b^2*c^8*d^2
*f^3) - (-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2
 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*
C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C
^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*
c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6
*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f
^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 1
2*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*
C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f
^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*
c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*
f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2
 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 1
6*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2
*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2))^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^
4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 + 16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^
3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^
2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f
^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f
^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5)
 + 64*C*a^2*c*d^11*f^4 - 64*C*b^2*c*d^11*f^4 + 256*C*a^2*c^3*d^9*f^4 + 384*
C*a^2*c^5*d^7*f^4 + 256*C*a^2*c^7*d^5*f^4 + 64*C*a^2*c^9*d^3*f^4 - 256*C*b^
2*c^3*d^9*f^4 - 384*C*b^2*c^5*d^7*f^4 - 256*C*b^2*c^7*d^5*f^4 - 64*C*b^2*c^
9*d^3*f^4 + 64*C*a*b*d^12*f^4 + 192*C*a*b*c^2*d^10*f^4 + 128*C*a*b*c^4*d^8*
f^4 - 128*C*a*b*c^6*d^6*f^4 - 192*C*a*b*c^8*d^4*f^4 - 64*C*a*b*c^10*d^2*f^4
))*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32
*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b
^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^
2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d
^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)
)^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 +
16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2
*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a
^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))
^(1/2) - 16*C^3*a^6*d^9*f^2 + 16*C^3*b^6*d^9*f^2 + 16*C^3*a^2*b^4*d^9*f^2 -
 16*C^3*a^4*b^2*d^9*f^2 - 48*C^3*a^6*c^2*d^7*f^2 - 48*C^3*a^6*c^4*d^5*f^2 -
 16*C^3*a^6*c^6*d^3*f^2 + 48*C^3*b^6*c^2*d^7*f^2 + 48*C^3*b^6*c^4*d^5*f^2 +
 16*C^3*b^6*c^6*d^3*f^2 + 32*C^3*a*b^5*c*d^8*f^2 + 32*C^3*a^5*b*c*d^8*f^2 +
 96*C^3*a*b^5*c^3*d^6*f^2 + 96*C^3*a*b^5*c^5*d^4*f^2 + 32*C^3*a*b^5*c^7*d^2
*f^2 + 64*C^3*a^3*b^3*c*d^8*f^2 + 96*C^3*a^5*b*c^3*d^6*f^2 + 96*C^3*a^5*b*c

^5*d^4*f^2 + 32*C^3*a^5*b*c^7*d^2*f^2 + 48*C^3*a^2*b^4*c^2*d^7*f^2 + 48*C^3
*a^2*b^4*c^4*d^5*f^2 + 16*C^3*a^2*b^4*c^6*d^3*f^2 + 192*C^3*a^3*b^3*c^3*d^6
*f^2 + 192*C^3*a^3*b^3*c^5*d^4*f^2 + 64*C^3*a^3*b^3*c^7*d^2*f^2 - 48*C^3*a^
4*b^2*c^2*d^7*f^2 - 48*C^3*a^4*b^2*c^4*d^5*f^2 - 16*C^3*a^4*b^2*c^6*d^3*f^2
))*(-(((8*C^2*a^4*c^3*f^2 + 8*C^2*b^4*c^3*f^2 - 48*C^2*a^2*b^2*c^3*f^2 + 32
*C^2*a*b^3*d^3*f^2 - 32*C^2*a^3*b*d^3*f^2 - 24*C^2*a^4*c*d^2*f^2 - 24*C^2*b
^4*c*d^2*f^2 - 96*C^2*a*b^3*c^2*d*f^2 + 96*C^2*a^3*b*c^2*d*f^2 + 144*C^2*a^
2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d
^2*f^4)*(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)
)^(1/2) + 4*C^2*a^4*c^3*f^2 + 4*C^2*b^4*c^3*f^2 - 24*C^2*a^2*b^2*c^3*f^2 +
16*C^2*a*b^3*d^3*f^2 - 16*C^2*a^3*b*d^3*f^2 - 12*C^2*a^4*c*d^2*f^2 - 12*C^2
*b^4*c*d^2*f^2 - 48*C^2*a*b^3*c^2*d*f^2 + 48*C^2*a^3*b*c^2*d*f^2 + 72*C^2*a
^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))
^(1/2)*2i - atan(-(((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^
2*b^4*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*
a^4*c^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*
b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A^2*a*b^3*c*d^9*f^3 - 128*A^
2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 +
128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3*d^7*f^3 - 384*A^2*a^3*b*c^5*
d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*a^2*b^2*c^2*d^8*f^3 + 192*A^2
*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*f^3) - ((((8*A^2*a^4*c^3*f^2
+ 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^
2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^
3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16
*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8
+ 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^
2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*
A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*
b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6
*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x
))^(1/2)*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2
+ 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*
A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A
^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*
c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6
*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f
^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 1
2*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*
A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f
^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d
^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*A*a^2*c*d^11*f^4 + 64*A*b^
2*c*d^11*f^4 - 256*A*a^2*c^3*d^9*f^4 - 384*A*a^2*c^5*d^7*f^4 - 256*A*a^2*c^
7*d^5*f^4 - 64*A*a^2*c^9*d^3*f^4 + 256*A*b^2*c^3*d^9*f^4 + 384*A*b^2*c^5*d^
7*f^4 + 256*A*b^2*c^7*d^5*f^4 + 64*A*b^2*c^9*d^3*f^4 - 64*A*a*b*d^12*f^4 -
192*A*a*b*c^2*d^10*f^4 - 128*A*a*b*c^4*d^8*f^4 + 128*A*a*b*c^6*d^6*f^4 + 19

```
2*A*a*b*c^8*d^4*f^4 + 64*A*a*b*c^10*d^2*f^4))*((((8*A^2*a^4*c^3*f^2 + 8*A^2
*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b
*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d
*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^
4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^
4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A
^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3
*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2
*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 +
d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))
^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3
 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3
 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3
 + 128*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^
7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3
*b*c^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 19
2*A^2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^
8*d^2*f^3) - ((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3
*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 -
 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 1
44*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 +
 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4
*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c
^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2
 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 -
 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d
^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b
^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d
^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f
^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4
+ 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*
a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2
*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b
*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d
*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^
6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^1
0*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f
^5) + 64*A*a^2*c*d^11*f^4 - 64*A*b^2*c*d^11*f^4 + 256*A*a^2*c^3*d^9*f^4 + 3
84*A*a^2*c^5*d^7*f^4 + 256*A*a^2*c^7*d^5*f^4 + 64*A*a^2*c^9*d^3*f^4 - 256*A
*b^2*c^3*d^9*f^4 - 384*A*b^2*c^5*d^7*f^4 - 256*A*b^2*c^7*d^5*f^4 - 64*A*b^2
*c^9*d^3*f^4 + 64*A*a*b*d^12*f^4 + 192*A*a*b*c^2*d^10*f^4 + 128*A*a*b*c^4*d
^8*f^4 - 128*A*a*b*c^6*d^6*f^4 - 192*A*a*b*c^8*d^4*f^4 - 64*A*a*b*c^10*d^2*
f^4))*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 +
32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2
*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*
```

a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4
*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^
2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2
- 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A
^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2
*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)
))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4
*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c
^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c
^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3
*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*
A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f
^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*
b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*f^3) - ((((8*A^2*a^4*c^3*f^2 + 8*A
^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3
*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2
*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*
f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*
A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4
*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a
^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c
^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4
+ d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1
/2)*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32
*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b
^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^
2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d
^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)
)^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 -
16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2
*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a
^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))
^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^
5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*A*a^2*c*d^11*f^4 + 64*A*b^2*c*d
^11*f^4 - 256*A*a^2*c^3*d^9*f^4 - 384*A*a^2*c^5*d^7*f^4 - 256*A*a^2*c^7*d^5
*f^4 - 64*A*a^2*c^9*d^3*f^4 + 256*A*b^2*c^3*d^9*f^4 + 384*A*b^2*c^5*d^7*f^4
 + 256*A*b^2*c^7*d^5*f^4 + 64*A*b^2*c^9*d^3*f^4 - 64*A*a*b*d^12*f^4 - 192*A
*a*b*c^2*d^10*f^4 - 128*A*a*b*c^4*d^8*f^4 + 128*A*a*b*c^6*d^6*f^4 + 192*A*a
*b*c^8*d^4*f^4 + 64*A*a*b*c^10*d^2*f^4))*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*
c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*
f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2
+ 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 1
6*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2
*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^
4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^

```
3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^
2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f
^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(
16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^
2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^
2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A
^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 +
384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3*d
^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*a^
2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*f^
3) - ((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 3
2*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*
b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a
^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*
d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2
))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 -
 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^
2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*
a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
)^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f
^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 -
 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*
A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6
*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6
 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3
*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2
 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 4
8*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 +
3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 +
640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 64
*A*a^2*c*d^11*f^4 - 64*A*b^2*c*d^11*f^4 + 256*A*a^2*c^3*d^9*f^4 + 384*A*a^2
*c^5*d^7*f^4 + 256*A*a^2*c^7*d^5*f^4 + 64*A*a^2*c^9*d^3*f^4 - 256*A*b^2*c^3
*d^9*f^4 - 384*A*b^2*c^5*d^7*f^4 - 256*A*b^2*c^7*d^5*f^4 - 64*A*b^2*c^9*d^3
*f^4 + 64*A*a*b*d^12*f^4 + 192*A*a*b*c^2*d^10*f^4 + 128*A*a*b*c^4*d^8*f^4 -
 128*A*a*b*c^6*d^6*f^4 - 192*A*a*b*c^8*d^4*f^4 - 64*A*a*b*c^10*d^2*f^4))*((
((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a
*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d
^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*
c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4
)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2
) - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2
*a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c
*d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2
*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)
 - 16*A^3*a^6*d^9*f^2 + 16*A^3*b^6*d^9*f^2 + 16*A^3*a^2*b^4*d^9*f^2 - 16*A^
```

```
3*a^4*b^2*d^9*f^2 - 48*A^3*a^6*c^2*d^7*f^2 - 48*A^3*a^6*c^4*d^5*f^2 - 16*A^
3*a^6*c^6*d^3*f^2 + 48*A^3*b^6*c^2*d^7*f^2 + 48*A^3*b^6*c^4*d^5*f^2 + 16*A^
3*b^6*c^6*d^3*f^2 + 32*A^3*a*b^5*c*d^8*f^2 + 32*A^3*a^5*b*c*d^8*f^2 + 96*A^
3*a*b^5*c^3*d^6*f^2 + 96*A^3*a*b^5*c^5*d^4*f^2 + 32*A^3*a*b^5*c^7*d^2*f^2 +
 64*A^3*a^3*b^3*c*d^8*f^2 + 96*A^3*a^5*b*c^3*d^6*f^2 + 96*A^3*a^5*b*c^5*d^4
*f^2 + 32*A^3*a^5*b*c^7*d^2*f^2 + 48*A^3*a^2*b^4*c^2*d^7*f^2 + 48*A^3*a^2*b
^4*c^4*d^5*f^2 + 16*A^3*a^2*b^4*c^6*d^3*f^2 + 192*A^3*a^3*b^3*c^3*d^6*f^2 +
 192*A^3*a^3*b^3*c^5*d^4*f^2 + 64*A^3*a^3*b^3*c^7*d^2*f^2 - 48*A^3*a^4*b^2*
c^2*d^7*f^2 - 48*A^3*a^4*b^2*c^4*d^5*f^2 - 16*A^3*a^4*b^2*c^6*d^3*f^2))*(((
(8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*
b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^
2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c
*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)
*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2)
 - 4*A^2*a^4*c^3*f^2 - 4*A^2*b^4*c^3*f^2 + 24*A^2*a^2*b^2*c^3*f^2 - 16*A^2*
a*b^3*d^3*f^2 + 16*A^2*a^3*b*d^3*f^2 + 12*A^2*a^4*c*d^2*f^2 + 12*A^2*b^4*c*
d^2*f^2 + 48*A^2*a*b^3*c^2*d*f^2 - 48*A^2*a^3*b*c^2*d*f^2 - 72*A^2*a^2*b^2*
c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*
2i - atan(-(((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d
^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6
*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6
*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b
*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^
2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3
 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^
2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*f^3) - (-(((8*A^2*a^4*c^3*f^2 + 8*A^
2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*
b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*
d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f
^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A
^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*
A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^
3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^
2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 +
 d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/
2)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32
*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b
^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^
2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d
^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)
)^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 +
16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2
*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a
^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))
^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^
```

5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*A*a^2*c*d^11*f^4 + 64*A*b^2*c*d
^11*f^4 - 256*A*a^2*c^3*d^9*f^4 - 384*A*a^2*c^5*d^7*f^4 - 256*A*a^2*c^7*d^5
*f^4 - 64*A*a^2*c^9*d^3*f^4 + 256*A*b^2*c^3*d^9*f^4 + 384*A*b^2*c^5*d^7*f^4
 + 256*A*b^2*c^7*d^5*f^4 + 64*A*b^2*c^9*d^3*f^4 - 64*A*a*b*d^12*f^4 - 192*A
*a*b*c^2*d^10*f^4 - 128*A*a*b*c^4*d^8*f^4 + 128*A*a*b*c^6*d^6*f^4 + 192*A*a
*b*c^8*d^4*f^4 + 64*A*a*b*c^10*d^2*f^4))*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4
*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3
*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2
 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 +
16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^
2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b
^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d
^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f
^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*
f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1/
2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 3
2*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 3
2*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 1
28*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^
3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c
^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^
2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^
2*f^3) - (-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^
2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24
*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*
A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48
*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^
6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*
f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 -
12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72
*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*
f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4
*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3
*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2
 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 +
16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^
2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b
^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d
^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f
^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*
f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*
f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5
) + 64*A*a^2*c*d^11*f^4 - 64*A*b^2*c*d^11*f^4 + 256*A*a^2*c^3*d^9*f^4 + 384
*A*a^2*c^5*d^7*f^4 + 256*A*a^2*c^7*d^5*f^4 + 64*A*a^2*c^9*d^3*f^4 - 256*A*b
^2*c^3*d^9*f^4 - 384*A*b^2*c^5*d^7*f^4 - 256*A*b^2*c^7*d^5*f^4 - 64*A*b^2*c

```
^9*d^3*f^4 + 64*A*a*b*d^12*f^4 + 192*A*a*b*c^2*d^10*f^4 + 128*A*a*b*c^4*d^8
*f^4 - 128*A*a*b*c^6*d^6*f^4 - 192*A*a*b*c^8*d^4*f^4 - 64*A*a*b*c^10*d^2*f^
4))*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 3
2*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*
b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a
^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*
d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2
))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 +
 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^
2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*
a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
)^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*
d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^
6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^
6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*
b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3 + 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A
^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^
3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b
^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*f^3) - (-(((8*A^2*a^4*c^3*f^2 + 8*A
^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3
*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2
*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*
f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*
A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4
*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a
^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c
^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4
+ d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1
/2)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 3
2*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*
b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a
^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*
d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2
))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 +
 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^
2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*
a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
)^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f
^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 64*A*a^2*c*d^11*f^4 + 64*A*b^2*c*
d^11*f^4 - 256*A*a^2*c^3*d^9*f^4 - 384*A*a^2*c^5*d^7*f^4 - 256*A*a^2*c^7*d^
5*f^4 - 64*A*a^2*c^9*d^3*f^4 + 256*A*b^2*c^3*d^9*f^4 + 384*A*b^2*c^5*d^7*f^
4 + 256*A*b^2*c^7*d^5*f^4 + 64*A*b^2*c^9*d^3*f^4 - 64*A*a*b*d^12*f^4 - 192*
A*a*b*c^2*d^10*f^4 - 128*A*a*b*c^4*d^8*f^4 + 128*A*a*b*c^6*d^6*f^4 + 192*A*
a*b*c^8*d^4*f^4 + 64*A*a*b*c^10*d^2*f^4))*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^
4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^
```

```
3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^
2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 +
 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a
^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*
b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*
d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*
f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6
*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)
*(16*A^2*a^4*d^10*f^3 + 16*A^2*b^4*d^10*f^3 - 96*A^2*a^2*b^2*d^10*f^3 + 32*
A^2*a^4*c^2*d^8*f^3 - 32*A^2*a^4*c^6*d^4*f^3 - 16*A^2*a^4*c^8*d^2*f^3 + 32*
A^2*b^4*c^2*d^8*f^3 - 32*A^2*b^4*c^6*d^4*f^3 - 16*A^2*b^4*c^8*d^2*f^3 + 128
*A^2*a*b^3*c*d^9*f^3 - 128*A^2*a^3*b*c*d^9*f^3 + 384*A^2*a*b^3*c^3*d^7*f^3
+ 384*A^2*a*b^3*c^5*d^5*f^3 + 128*A^2*a*b^3*c^7*d^3*f^3 - 384*A^2*a^3*b*c^3
*d^7*f^3 - 384*A^2*a^3*b*c^5*d^5*f^3 - 128*A^2*a^3*b*c^7*d^3*f^3 - 192*A^2*
a^2*b^2*c^2*d^8*f^3 + 192*A^2*a^2*b^2*c^6*d^4*f^3 + 96*A^2*a^2*b^2*c^8*d^2*
f^3) - (-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2
+ 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A
^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^
2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c
^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*
b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^
2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12
*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A
^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^
4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c
^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f
^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 +
 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16
*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*
b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4
*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 16*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3
*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2
 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^
4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^
5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5)
+ 64*A*a^2*c*d^11*f^4 - 64*A*b^2*c*d^11*f^4 + 256*A*a^2*c^3*d^9*f^4 + 384*A
*a^2*c^5*d^7*f^4 + 256*A*a^2*c^7*d^5*f^4 + 64*A*a^2*c^9*d^3*f^4 - 256*A*b^2
*c^3*d^9*f^4 - 384*A*b^2*c^5*d^7*f^4 - 256*A*b^2*c^7*d^5*f^4 - 64*A*b^2*c^9
*d^3*f^4 + 64*A*a*b*d^12*f^4 + 192*A*a*b*c^2*d^10*f^4 + 128*A*a*b*c^4*d^8*f
^4 - 128*A*a*b*c^6*d^6*f^4 - 192*A*a*b*c^8*d^4*f^4 - 64*A*a*b*c^10*d^2*f^4)
)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*
A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^
4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2
*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^
2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))
```

```
^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 1
6*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*
b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^
2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^
(1/2) - 16*A^3*a^6*d^9*f^2 + 16*A^3*b^6*d^9*f^2 + 16*A^3*a^2*b^4*d^9*f^2 -
16*A^3*a^4*b^2*d^9*f^2 - 48*A^3*a^6*c^2*d^7*f^2 - 48*A^3*a^6*c^4*d^5*f^2 -
16*A^3*a^6*c^6*d^3*f^2 + 48*A^3*b^6*c^2*d^7*f^2 + 48*A^3*b^6*c^4*d^5*f^2 +
16*A^3*b^6*c^6*d^3*f^2 + 32*A^3*a*b^5*c*d^8*f^2 + 32*A^3*a^5*b*c*d^8*f^2 +
96*A^3*a*b^5*c^3*d^6*f^2 + 96*A^3*a*b^5*c^5*d^4*f^2 + 32*A^3*a*b^5*c^7*d^2*
f^2 + 64*A^3*a^3*b^3*c*d^8*f^2 + 96*A^3*a^5*b*c^3*d^6*f^2 + 96*A^3*a^5*b*c^
5*d^4*f^2 + 32*A^3*a^5*b*c^7*d^2*f^2 + 48*A^3*a^2*b^4*c^2*d^7*f^2 + 48*A^3*
a^2*b^4*c^4*d^5*f^2 + 16*A^3*a^2*b^4*c^6*d^3*f^2 + 192*A^3*a^3*b^3*c^3*d^6*
f^2 + 192*A^3*a^3*b^3*c^5*d^4*f^2 + 64*A^3*a^3*b^3*c^7*d^2*f^2 - 48*A^3*a^4
*b^2*c^2*d^7*f^2 - 48*A^3*a^4*b^2*c^4*d^5*f^2 - 16*A^3*a^4*b^2*c^6*d^3*f^2)
)*(-(((8*A^2*a^4*c^3*f^2 + 8*A^2*b^4*c^3*f^2 - 48*A^2*a^2*b^2*c^3*f^2 + 32*
A^2*a*b^3*d^3*f^2 - 32*A^2*a^3*b*d^3*f^2 - 24*A^2*a^4*c*d^2*f^2 - 24*A^2*b^
4*c*d^2*f^2 - 96*A^2*a*b^3*c^2*d*f^2 + 96*A^2*a^3*b*c^2*d*f^2 + 144*A^2*a^2
*b^2*c*d^2*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^
2*f^4)*(A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2))
^(1/2) + 4*A^2*a^4*c^3*f^2 + 4*A^2*b^4*c^3*f^2 - 24*A^2*a^2*b^2*c^3*f^2 + 1
6*A^2*a*b^3*d^3*f^2 - 16*A^2*a^3*b*d^3*f^2 - 12*A^2*a^4*c*d^2*f^2 - 12*A^2*
b^4*c*d^2*f^2 - 48*A^2*a*b^3*c^2*d*f^2 + 48*A^2*a^3*b*c^2*d*f^2 + 72*A^2*a^
2*b^2*c*d^2*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^
(1/2)*2i - ((8*C*b^2*c - 4*C*a*b*d)/(d^3*f) - (4*C*b^2*c)/(d^3*f))*(c + d*t
an(e + f*x))^(1/2) + (2*B*b^2*(c + d*tan(e + f*x))^(1/2))/(d^2*f) + (2*C*b^
2*(c + d*tan(e + f*x))^(3/2))/(3*d^3*f) - (2*(A*a^2*d^2 + A*b^2*c^2 - 2*A*a
*b*c*d))/(d*f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2)) - (2*(C*b^2*c^4 + C*a
^2*c^2*d^2 - 2*C*a*b*c^3*d))/(d^3*f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2))
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(e+fx\right)\right)^2\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)}{\left(c+d\tan\left(e+fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(3/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x))**(3/2), x)
```

3.118 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=201

$$\frac{2(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{d^2f\left(c^2+d^2\right)\sqrt{c+d\tan(e+fx)}}-\frac{(b+ia)(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{3/2}}+\frac{(-b+ia)(A+iB-C)\tanh^{-1}\left(\right.}{f(c+id)^{3/2}}$$

```
[Out] -(I*a+b)*(A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d)^(3
/2)/f+(I*a-b)*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(c+I*
d)^(3/2)/f+2*(-a*d+b*c)*(A*d^2-B*c*d+C*c^2)/d^2/(c^2+d^2)/f/(c+d*tan(f*x+e)
)^(1/2)+2*b*C*(c+d*tan(f*x+e))^(1/2)/d^2/f
```

Rubi [A] time = 0.55, antiderivative size = 201, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3635, 3630, 3539, 3537, 63, 208}

$$\frac{2(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{d^2f\left(c^2+d^2\right)\sqrt{c+d\tan(e+fx)}}-\frac{(b+ia)(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{3/2}}+\frac{(-b+ia)(A+iB-C)\tanh^{-1}\left(\right.}{f(c+id)^{3/2}}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d*T
an[e + f*x])^(3/2),x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])
/((c - I*d)^(3/2)*f)) + ((I*a - b)*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e +
 f*x]]/Sqrt[c + I*d]])/((c + I*d)^(3/2)*f) + (2*(b*c - a*d)*(c^2*C - B*c*d
+ A*d^2))/(d^2*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]]) + (2*b*C*Sqrt[c + d*
Tan[e + f*x]])/(d^2*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \int \frac{ad(Ac-cC}{}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{2bC\sqrt{c+}}{}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{2bC\sqrt{c+}}{}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{2bC\sqrt{c+}}{}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{d^2\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{2bC\sqrt{c+}}{}$$

$$= -\frac{(ia+b)(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{3/2}f}$$

Mathematica [C] time = 2.59, size = 290, normalized size = 1.44

$$\frac{\frac{(-aAd+aBc+aCd+Abc+bBd-bcC)\left((d-ic)\,{}_2F_1\left(-\frac{1}{2},1;\frac{1}{2};\frac{c+d\tan(e+fx)}{c-id}\right)+(d+ic)\,{}_2F_1\left(-\frac{1}{2},1;\frac{1}{2};\frac{c+d\tan(e+fx)}{c+id}\right)\right)}{\left(c^2+d^2\right)\sqrt{c+d\tan(e+fx)}} + (aB+Ab-bC)\left(\frac{i\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+}}{\sqrt{c+id}}\right)}{\sqrt{c+id}}\right.}{df}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] ((A*b + a*B - b*C)*(((-I)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/
Sqrt[c - I*d] + (I*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/Sqrt[c
+ I*d]) - (2*(-2*b*c*C + b*B*d + 2*a*C*d))/(d*Sqrt[c + d*Tan[e + f*x]]) + (
(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*(((-I)*c + d)*Hypergeometri
c2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)] + (I*c + d)*Hypergeometr
ic2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)]))/((c^2 + d^2)*Sqrt[c +
 d*Tan[e + f*x]]) + (2*C*(a + b*Tan[e + f*x]))/Sqrt[c + d*Tan[e + f*x]])/(d
*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.44, size = 23472, normalized size = 116.78

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/2)
,x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [B] time = 41.07, size = 40542, normalized size = 201.70

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x))^(3/2),x)

[Out] atan((((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^
3 + 16*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 -
 16*A^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 +
 16*B^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 -
 16*C^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B
*C*a^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A
*B*a^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A
*C*a^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64
*B*C*a^2*c^7*d^3*f^3) - ((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a
^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 -
 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*
B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^
6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 +
C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2
*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 -
4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f
^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 2
4*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6
*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x
))^(1/2)*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16
*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d
^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2
 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*
f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C
^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*
B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f
^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2
*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d
*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4
+ 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5
+ 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) -
32*B*a*d^12*f^4 - 256*A*a*c^3*d^9*f^4 - 384*A*a*c^5*d^7*f^4 - 256*A*a*c^7*d
^5*f^4 - 64*A*a*c^9*d^3*f^4 - 96*B*a*c^2*d^10*f^4 - 64*B*a*c^4*d^8*f^4 + 64
*B*a*c^6*d^6*f^4 + 96*B*a*c^8*d^4*f^4 + 32*B*a*c^10*d^2*f^4 + 256*C*a*c^3*d
^9*f^4 + 384*C*a*c^5*d^7*f^4 + 256*C*a*c^7*d^5*f^4 + 64*C*a*c^9*d^3*f^4 - 6
4*A*a*c*d^11*f^4 + 64*C*a*c*d^11*f^4))*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^
3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*
C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*
d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^
2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a
^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*
C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2
```

*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 + 16*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*A^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*B^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*C^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C*a^2*c^7*d^3*f^3) - ((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 32*B*a*d^12*f^4 + 256*A*a*c^3*d^9*f^4 + 384*A*a*c^5*d^7*f^4 + 256*A*a*c^7*d^5*f^4 + 64*A*a*c^9*d^3*f^4 + 96*B*a*c^2*d^10*f^4 + 64*B*a*c^4*d^8*f^4 - 64*B*a*c^6*d^6*f^4 - 96*B*a*c^8*d^4*f^4 - 32*B*a*c^10*d^2*f^4 - 256*C*a*c^3*d^9*f^4 - 384*C*a*c^5*d^7*f^4 - 256*C*a*c^7*d^5*f^4 - 64*C*a*c^9*d^3*f^4 + 64*A*a*c*d^11*f^4 - 64*C*a*c*d^11*f^4))*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a

```
^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*
c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C
*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d
^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2
)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i)/(((c
+ d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 + 16*C^2
*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*A^2*a^
2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*B^2*a^
2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*C^2*a^
2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a^2*c*d
^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a^2*c^7
*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a^2*c^8
*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C*a^2*c
^7*d^3*f^3) - ((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2
 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^
2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*
d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16
*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 -
4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 -
 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*
c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^
2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*
c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6
*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(
(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d
^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 2
4*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*
a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c
^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4
*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4)
)^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B
*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2
 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*
A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^
4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*
d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*B*a*d^1
2*f^4 - 256*A*a*c^3*d^9*f^4 - 384*A*a*c^5*d^7*f^4 - 256*A*a*c^7*d^5*f^4 - 6
4*A*a*c^9*d^3*f^4 - 96*B*a*c^2*d^10*f^4 - 64*B*a*c^4*d^8*f^4 + 64*B*a*c^6*d
^6*f^4 + 96*B*a*c^8*d^4*f^4 + 32*B*a*c^10*d^2*f^4 + 256*C*a*c^3*d^9*f^4 + 3
84*C*a*c^5*d^7*f^4 + 256*C*a*c^7*d^5*f^4 + 64*C*a*c^9*d^3*f^4 - 64*A*a*c*d^
11*f^4 + 64*C*a*c*d^11*f^4))*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*
C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*
f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 +
48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (
16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a
```

^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f ^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2* d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^ 2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16 *(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 + 16*C^2*a^2*d^1 0*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*A^2*a^2*c^8*d^ 2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*B^2*a^2*c^8*d^ 2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*C^2*a^2*c^8*d^ 2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C*a^2*c^7*d^3*f ^3) - ((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A* B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2* f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3* a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2 *C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c* d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^ 2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3 *c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2 *a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^ 2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^ 2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f ^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a ^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3 *f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^ 2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2* c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 32*B*a*d^12*f^4 + 256*A*a*c^3*d^9*f^4 + 384*A*a*c^5*d^7*f^4 + 256*A*a*c^7*d^5*f^4 + 64*A*a*c^ 9*d^3*f^4 + 96*B*a*c^2*d^10*f^4 + 64*B*a*c^4*d^8*f^4 - 64*B*a*c^6*d^6*f^4 - 96*B*a*c^8*d^4*f^4 - 32*B*a*c^10*d^2*f^4 - 256*C*a*c^3*d^9*f^4 - 384*C*a*c ^5*d^7*f^4 - 256*C*a*c^7*d^5*f^4 - 64*C*a*c^9*d^3*f^4 + 64*A*a*c*d^11*f^4 - 64*C*a*c*d^11*f^4))*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2* c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24 *A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a ^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f

^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4
*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^
2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c^3*f^2 - 4*C
^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*a^2*d^3*f^2
+ 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^2*f^2 - 24*A
*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^
4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - 16*A^3*a^3*d^9*f^2 +
16*C^3*a^3*d^9*f^2 - 48*A^3*a^3*c^2*d^7*f^2 - 48*A^3*a^3*c^4*d^5*f^2 - 16*
A^3*a^3*c^6*d^3*f^2 + 48*B^3*a^3*c^3*d^6*f^2 + 48*B^3*a^3*c^5*d^4*f^2 + 16*
B^3*a^3*c^7*d^2*f^2 + 48*C^3*a^3*c^2*d^7*f^2 + 48*C^3*a^3*c^4*d^5*f^2 + 16*
C^3*a^3*c^6*d^3*f^2 - 16*A*B^2*a^3*d^9*f^2 - 48*A*C^2*a^3*d^9*f^2 + 48*A^2*
C*a^3*d^9*f^2 + 16*B^2*C*a^3*d^9*f^2 + 16*B^3*a^3*c*d^8*f^2 - 48*A*B^2*a^3*
c^2*d^7*f^2 - 48*A*B^2*a^3*c^4*d^5*f^2 - 16*A*B^2*a^3*c^6*d^3*f^2 + 48*A^2*
B*a^3*c^3*d^6*f^2 + 48*A^2*B*a^3*c^5*d^4*f^2 + 16*A^2*B*a^3*c^7*d^2*f^2 - 1
44*A*C^2*a^3*c^2*d^7*f^2 - 144*A*C^2*a^3*c^4*d^5*f^2 - 48*A*C^2*a^3*c^6*d^3
*f^2 + 144*A^2*C*a^3*c^2*d^7*f^2 + 144*A^2*C*a^3*c^4*d^5*f^2 + 48*A^2*C*a^3
*c^6*d^3*f^2 + 48*B*C^2*a^3*c^3*d^6*f^2 + 48*B*C^2*a^3*c^5*d^4*f^2 + 16*B*C
^2*a^3*c^7*d^2*f^2 + 48*B^2*C*a^3*c^2*d^7*f^2 + 48*B^2*C*a^3*c^4*d^5*f^2 +
16*B^2*C*a^3*c^6*d^3*f^2 + 16*A^2*B*a^3*c*d^8*f^2 + 16*B*C^2*a^3*c*d^8*f^2
- 96*A*B*C*a^3*c^3*d^6*f^2 - 96*A*B*C*a^3*c^5*d^4*f^2 - 32*A*B*C*a^3*c^7*d^
2*f^2 - 32*A*B*C*a^3*c*d^8*f^2))*((((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2
+ 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*
d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^
2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4
- (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B
^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^
4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^3*f^2 + 4*B^2*a^2*c
^3*f^2 - 4*C^2*a^2*c^3*f^2 + 8*A*B*a^2*d^3*f^2 + 8*A*C*a^2*c^3*f^2 - 8*B*C*
a^2*d^3*f^2 + 12*A^2*a^2*c*d^2*f^2 - 12*B^2*a^2*c*d^2*f^2 + 12*C^2*a^2*c*d^
2*f^2 - 24*A*B*a^2*c^2*d*f^2 - 24*A*C*a^2*c*d^2*f^2 + 24*B*C*a^2*c^2*d*f^2)
/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*2i - atan(
((((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^
2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2
+ 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A
*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 4
8*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4
- 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b
^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*
A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*
f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 +
24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2
*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2
*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*
A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*
d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^

2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*A*b*d^12*f^4 + 32*C*b*d^12*f^4 - 96*A*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*d^11*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 - 64*A*B*b^2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192*A*B*b^2*c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 192*B*C*b^2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i - (((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(32*C*b*d^12*f^4 - 32*A*b*d^12*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b

^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2
*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^
2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4
+ B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2
*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^
2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B
*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c
*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f
^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d
^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^
4*f^5 + 64*c^11*d^2*f^5) - 96*A*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*
b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*
f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C
*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*
f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*d^11*f^4) - (c + d*tan(e + f*x))^(1/2)
*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*
b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*
b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*
b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*
b^2*d^10*f^3 - 64*A*B*b^2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^
3*d^7*f^3 - 192*A*B*b^2*c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c
^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*
c^3*d^7*f^3 + 192*B*C*b^2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*((((8*A^2*
b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 -
16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2
*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2
*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^
4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^
4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) +
4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*
f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2
*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c
*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3
*c^4*d^2*f^4)))^(1/2)*1i)/(16*B^3*b^3*d^9*f^2 - (((((8*A^2*b^2*c^3*f^2 - 8*
B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f
^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*
C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^
2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f
^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^
4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f
^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*
c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 -
12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*
C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^
(1/2)*(32*C*b*d^12*f^4 - 32*A*b*d^12*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8

```
*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f
^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^
2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*
c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d
^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3
*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1
/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2
*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 1
2*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*
b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^
4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*
f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*A*b*c^2*d^1
0*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 + 32*A
*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c^7*d
^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4 - 64
*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*d^11
*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^10*f
^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^6*d^4*f^3
- 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^6*d^4*f^3
+ 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^6*d^4*f^3
- 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 - 64*A*B*b^2*c*d^9*f^3 + 64*
B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192*A*B*b^2*c^5*d^5*f^3 - 64*
A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4*f^3 + 32*
A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 192*B*C*b^2*c^5*d^5*f^3 + 6
4*B*C*b^2*c^7*d^3*f^3))*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b
^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 -
 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*
B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^
6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 +
C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2
*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 +
4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f
^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 2
4*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6
*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - (((((8*A^2*b^2*c^
3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C
*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2
*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 -
 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48
*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*
A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2
*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 -
8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c
*d^2*f^2 - 12*C^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f
^2 - 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d
```

^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b
^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 +
16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b
^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2
*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(
A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6
*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 -
4*B^2*b^2*c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f
^2 + 8*B*C*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C
^2*b^2*c*d^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2
*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)
*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 32
0*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*A*b*d^12*f^4 + 32*C*b*d^12*f^4 - 96*A
*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*
f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c^5*d^7*f^4 + 256
*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4 + 64*C*b*c^4*d^
8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^10*d^2*f^4 + 64*
B*b*c*d^11*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^10*f^3 - 16*B^2*
b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^
6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^
6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^
6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 - 64*A*B*b^2*c*d^9
*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192*A*B*b^2*c^5*d^5
*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4
*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 192*B*C*b^2*c^5*d
^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2
 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2
*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f
^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/
4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 +
B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b
^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*
c^3*f^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C
*b^2*d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d
^2*f^2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2
)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) + 48*A^3*
b^3*c^3*d^6*f^2 + 48*A^3*b^3*c^5*d^4*f^2 + 16*A^3*b^3*c^7*d^2*f^2 + 48*B^3*
b^3*c^2*d^7*f^2 + 48*B^3*b^3*c^4*d^5*f^2 + 16*B^3*b^3*c^6*d^3*f^2 - 48*C^3*
b^3*c^3*d^6*f^2 - 48*C^3*b^3*c^5*d^4*f^2 - 16*C^3*b^3*c^7*d^2*f^2 + 16*A^2*
B*b^3*d^9*f^2 + 16*B*C^2*b^3*d^9*f^2 + 16*A^3*b^3*c*d^8*f^2 - 16*C^3*b^3*c*
d^8*f^2 + 48*A*B^2*b^3*c^3*d^6*f^2 + 48*A*B^2*b^3*c^5*d^4*f^2 + 16*A*B^2*b^
3*c^7*d^2*f^2 + 48*A^2*B*b^3*c^2*d^7*f^2 + 48*A^2*B*b^3*c^4*d^5*f^2 + 16*A^
2*B*b^3*c^6*d^3*f^2 + 144*A*C^2*b^3*c^3*d^6*f^2 + 144*A*C^2*b^3*c^5*d^4*f^2
 + 48*A*C^2*b^3*c^7*d^2*f^2 - 144*A^2*C*b^3*c^3*d^6*f^2 - 144*A^2*C*b^3*c^5
*d^4*f^2 - 48*A^2*C*b^3*c^7*d^2*f^2 + 48*B*C^2*b^3*c^2*d^7*f^2 + 48*B*C^2*b

^3*c^4*d^5*f^2 + 16*B*C^2*b^3*c^6*d^3*f^2 - 48*B^2*C*b^3*c^3*d^6*f^2 - 48*B
^2*C*b^3*c^5*d^4*f^2 - 16*B^2*C*b^3*c^7*d^2*f^2 - 32*A*B*C*b^3*d^9*f^2 + 16
*A*B^2*b^3*c*d^8*f^2 + 48*A*C^2*b^3*c*d^8*f^2 - 48*A^2*C*b^3*c*d^8*f^2 - 16
*B^2*C*b^3*c*d^8*f^2 - 96*A*B*C*b^3*c^2*d^7*f^2 - 96*A*B*C*b^3*c^4*d^5*f^2
- 32*A*B*C*b^3*c^6*d^3*f^2))*((((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*
C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*
f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 +
48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (
16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b
^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 +
2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^3*f^2 - 4*B^2*b^2*c^3*f
^2 + 4*C^2*b^2*c^3*f^2 - 8*A*B*b^2*d^3*f^2 - 8*A*C*b^2*c^3*f^2 + 8*B*C*b^2*
d^3*f^2 - 12*A^2*b^2*c*d^2*f^2 + 12*B^2*b^2*c*d^2*f^2 - 12*C^2*b^2*c*d^2*f^
2 + 24*A*B*b^2*c^2*d*f^2 + 24*A*C*b^2*c*d^2*f^2 - 24*B*C*b^2*c^2*d*f^2)/(16
*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*2i - atan((((-
(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d
^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 2
4*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*
b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c
^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4
*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4)
)^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B
*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2
 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*
A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^
4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c
^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*
C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^
2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2
- 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 4
8*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2
*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^
2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 +
 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*
c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*
f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*
d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*
c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*A*b*d^12*f^4 + 32*C*b
*d^12*f^4 - 96*A*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 +
 96*A*b*c^8*d^4*f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c
^5*d^7*f^4 + 256*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4
 + 64*C*b*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^
10*d^2*f^4 + 64*B*b*c*d^11*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^
10*f^3 - 16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3
 - 32*A^2*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3

+ 32*B^2*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3
 - 32*C^2*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 -
64*A*B*b^2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192
*A*B*b^2*c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64
*A*C*b^2*c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 1
92*B*C*b^2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*(-(((8*A^2*b^2*c^3*f^2 -
8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3
*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 2
4*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*
b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2
f^4)(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*
b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3
*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^
2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2
 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*
B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
)^(1/2)*1i - ((-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^
2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b
^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2
*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 1
6*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 -
 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4
 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2
*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A
^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2
*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^
6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(32*C*b*d^12*f^4 - 32*A*b*d^
12*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*
f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*
b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^
2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)
^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4
 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^
2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b
^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*
B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*
c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*
f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*
d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d
^4*f^5 + 64*c^11*d^2*f^5) - 96*A*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A
*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9
*f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*
C*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4
*f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*d^11*f^4) - (c + d*tan(e + f*x))^(1/2
)*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2

*b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2
*b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2
*b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C
*b^2*d^10*f^3 - 64*A*B*b^2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c
^3*d^7*f^3 - 192*A*B*b^2*c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*
c^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2
*c^3*d^7*f^3 + 192*B*C*b^2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*(-(((8*A^
2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2
- 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b
^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d
^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*
f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*
b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2)
- 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^
3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B
^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2
*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 +
3*c^4*d^2*f^4)))^(1/2)*1i)/(16*B^3*b^3*d^9*f^2 - ((-(((8*A^2*b^2*c^3*f^2 -
8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^
3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 -
24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C
*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^
2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2
*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^
3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b
^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^
2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24
*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)
))^(1/2)*(32*C*b*d^12*f^4 - 32*A*b*d^12*f^4 - (c + d*tan(e + f*x))^(1/2)*(-
(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d
^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 2
4*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*
b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c
^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4
*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4)
)^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B
*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2
- 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*
A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^
4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*
d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 96*A*b*c^2
*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 +
32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c
^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4 + 64*C*b*c^4*d^8*f^4
- 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^10*d^2*f^4 + 64*B*b*c*

d^11*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^10*f^3 - 16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3 - 32*A^2*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3 + 32*B^2*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3 - 32*C^2*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 - 64*A*B*b^2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192*A*B*b^2*c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64*A*C*b^2*c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 192*B*C*b^2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*(-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32*A*b*d^12*f^4 + 32*C*b*d^12*f^4 - 96*A*b*c^2*d^10*f^4 - 64*A*b*c^4*d^8*f^4 + 64*A*b*c^6*d^6*f^4 + 96*A*b*c^8*d^4*f^4 + 32*A*b*c^10*d^2*f^4 + 256*B*b*c^3*d^9*f^4 + 384*B*b*c^5*d^7*f^4 + 256*B*b*c^7*d^5*f^4 + 64*B*b*c^9*d^3*f^4 + 96*C*b*c^2*d^10*f^4 + 64*C*b

```
*c^4*d^8*f^4 - 64*C*b*c^6*d^6*f^4 - 96*C*b*c^8*d^4*f^4 - 32*C*b*c^10*d^2*f^
4 + 64*B*b*c*d^11*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^10*f^3 -
16*B^2*b^2*d^10*f^3 + 16*C^2*b^2*d^10*f^3 + 32*A^2*b^2*c^2*d^8*f^3 - 32*A^2
*b^2*c^6*d^4*f^3 - 16*A^2*b^2*c^8*d^2*f^3 - 32*B^2*b^2*c^2*d^8*f^3 + 32*B^2
*b^2*c^6*d^4*f^3 + 16*B^2*b^2*c^8*d^2*f^3 + 32*C^2*b^2*c^2*d^8*f^3 - 32*C^2
*b^2*c^6*d^4*f^3 - 16*C^2*b^2*c^8*d^2*f^3 - 32*A*C*b^2*d^10*f^3 - 64*A*B*b^
2*c*d^9*f^3 + 64*B*C*b^2*c*d^9*f^3 - 192*A*B*b^2*c^3*d^7*f^3 - 192*A*B*b^2*
c^5*d^5*f^3 - 64*A*B*b^2*c^7*d^3*f^3 - 64*A*C*b^2*c^2*d^8*f^3 + 64*A*C*b^2*
c^6*d^4*f^3 + 32*A*C*b^2*c^8*d^2*f^3 + 192*B*C*b^2*c^3*d^7*f^3 + 192*B*C*b^
2*c^5*d^5*f^3 + 64*B*C*b^2*c^7*d^3*f^3))*(-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2
*c^3*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16
*B*C*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2
*c*d^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d
*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^
4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A
^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*
B^2*b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2
 - 8*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2
*b^2*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c
^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) +
 48*A^3*b^3*c^3*d^6*f^2 + 48*A^3*b^3*c^5*d^4*f^2 + 16*A^3*b^3*c^7*d^2*f^2 +
 48*B^3*b^3*c^2*d^7*f^2 + 48*B^3*b^3*c^4*d^5*f^2 + 16*B^3*b^3*c^6*d^3*f^2 -
 48*C^3*b^3*c^3*d^6*f^2 - 48*C^3*b^3*c^5*d^4*f^2 - 16*C^3*b^3*c^7*d^2*f^2 +
 16*A^2*B*b^3*d^9*f^2 + 16*B*C^2*b^3*d^9*f^2 + 16*A^3*b^3*c*d^8*f^2 - 16*C^
3*b^3*c*d^8*f^2 + 48*A*B^2*b^3*c^3*d^6*f^2 + 48*A*B^2*b^3*c^5*d^4*f^2 + 16*
A*B^2*b^3*c^7*d^2*f^2 + 48*A^2*B*b^3*c^2*d^7*f^2 + 48*A^2*B*b^3*c^4*d^5*f^2
 + 16*A^2*B*b^3*c^6*d^3*f^2 + 144*A*C^2*b^3*c^3*d^6*f^2 + 144*A*C^2*b^3*c^5
*d^4*f^2 + 48*A*C^2*b^3*c^7*d^2*f^2 - 144*A^2*C*b^3*c^3*d^6*f^2 - 144*A^2*C
*b^3*c^5*d^4*f^2 - 48*A^2*C*b^3*c^7*d^2*f^2 + 48*B*C^2*b^3*c^2*d^7*f^2 + 48
*B*C^2*b^3*c^4*d^5*f^2 + 16*B*C^2*b^3*c^6*d^3*f^2 - 48*B^2*C*b^3*c^3*d^6*f^
2 - 48*B^2*C*b^3*c^5*d^4*f^2 - 16*B^2*C*b^3*c^7*d^2*f^2 - 32*A*B*C*b^3*d^9*
f^2 + 16*A*B^2*b^3*c*d^8*f^2 + 48*A*C^2*b^3*c*d^8*f^2 - 48*A^2*C*b^3*c*d^8*
f^2 - 16*B^2*C*b^3*c*d^8*f^2 - 96*A*B*C*b^3*c^2*d^7*f^2 - 96*A*B*C*b^3*c^4*
d^5*f^2 - 32*A*B*C*b^3*c^6*d^3*f^2))*(-(((8*A^2*b^2*c^3*f^2 - 8*B^2*b^2*c^3
*f^2 + 8*C^2*b^2*c^3*f^2 - 16*A*B*b^2*d^3*f^2 - 16*A*C*b^2*c^3*f^2 + 16*B*C
*b^2*d^3*f^2 - 24*A^2*b^2*c*d^2*f^2 + 24*B^2*b^2*c*d^2*f^2 - 24*C^2*b^2*c*d
^2*f^2 + 48*A*B*b^2*c^2*d*f^2 + 48*A*C*b^2*c*d^2*f^2 - 48*B*C*b^2*c^2*d*f^2
)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*b^
4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C
^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^3*f^2 + 4*B^2*
b^2*c^3*f^2 - 4*C^2*b^2*c^3*f^2 + 8*A*B*b^2*d^3*f^2 + 8*A*C*b^2*c^3*f^2 - 8
*B*C*b^2*d^3*f^2 + 12*A^2*b^2*c*d^2*f^2 - 12*B^2*b^2*c*d^2*f^2 + 12*C^2*b^2
*c*d^2*f^2 - 24*A*B*b^2*c^2*d*f^2 - 24*A*C*b^2*c*d^2*f^2 + 24*B*C*b^2*c^2*d
*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*2i +
atan((((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^
```

```
3 + 16*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 -
 16*A^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 +
 16*B^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 -
 16*C^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B
*C*a^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A
*B*a^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A
*C*a^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64
*B*C*a^2*c^7*d^3*f^3) - (-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*
a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2
- 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A
*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c
^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 +
 C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^
2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 +
 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*
f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 +
24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^
6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*
x))^(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 -
16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c
*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f
^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^
6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A
*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*
A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3
*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a
^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2
*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^
4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^
5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5)
- 32*B*a*d^12*f^4 - 256*A*a*c^3*d^9*f^4 - 384*A*a*c^5*d^7*f^4 - 256*A*a*c^7
*d^5*f^4 - 64*A*a*c^9*d^3*f^4 - 96*B*a*c^2*d^10*f^4 - 64*B*a*c^4*d^8*f^4 +
64*B*a*c^6*d^6*f^4 + 96*B*a*c^8*d^4*f^4 + 32*B*a*c^10*d^2*f^4 + 256*C*a*c^3
*d^9*f^4 + 384*C*a*c^5*d^7*f^4 + 256*C*a*c^7*d^5*f^4 + 64*C*a*c^9*d^3*f^4 -
 64*A*a*c*d^11*f^4 + 64*C*a*c*d^11*f^4))*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2
*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16
*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2
*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d
*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^
4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A
^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*
B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2
 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2
*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c
^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1
```

i + ((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3
+ 16*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 1
6*A^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 1
6*B^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 1
6*C^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C
*a^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B
*a^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C
*a^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B
*C*a^2*c^7*d^3*f^3) - (-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^
2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 -
24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B
*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6
*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C
^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*
C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4
*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^
2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24
*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*
f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x)
)^(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16
*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d
^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2
+ 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*
f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C
^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*
B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f
^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2
*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d
*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4
+ 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5
+ 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) +
32*B*a*d^12*f^4 + 256*A*a*c^3*d^9*f^4 + 384*A*a*c^5*d^7*f^4 + 256*A*a*c^7*d
^5*f^4 + 64*A*a*c^9*d^3*f^4 + 96*B*a*c^2*d^10*f^4 + 64*B*a*c^4*d^8*f^4 - 64
*B*a*c^6*d^6*f^4 - 96*B*a*c^8*d^4*f^4 - 32*B*a*c^10*d^2*f^4 - 256*C*a*c^3*d
^9*f^4 - 384*C*a*c^5*d^7*f^4 - 256*C*a*c^7*d^5*f^4 - 64*C*a*c^9*d^3*f^4 + 6
4*A*a*c*d^11*f^4 - 64*C*a*c*d^11*f^4))*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c
^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B
*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c
*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f
^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*
a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2
*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^
2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 +
8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a
^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2

*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*1i)
/(((c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 +
16*C^2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*
A^2*a^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*
B^2*a^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*
C^2*a^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a
^2*c*d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a
^2*c^7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a
^2*c^8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C
*a^2*c^7*d^3*f^3) - (-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*
c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24
*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a
^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f
^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4
*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^
2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C
^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2
- 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A
*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^
4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^
(1/2)*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A
*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2
*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 +
48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^
4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3
*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^
2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2
- 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c
*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f
^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 +
3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 +
640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32
*B*a*d^12*f^4 - 256*A*a*c^3*d^9*f^4 - 384*A*a*c^5*d^7*f^4 - 256*A*a*c^7*d^5
*f^4 - 64*A*a*c^9*d^3*f^4 - 96*B*a*c^2*d^10*f^4 - 64*B*a*c^4*d^8*f^4 + 64*B
*a*c^6*d^6*f^4 + 96*B*a*c^8*d^4*f^4 + 32*B*a*c^10*d^2*f^4 + 256*C*a*c^3*d^9
*f^4 + 384*C*a*c^5*d^7*f^4 + 256*C*a*c^7*d^5*f^4 + 64*C*a*c^9*d^3*f^4 - 64*
A*a*c*d^11*f^4 + 64*C*a*c*d^11*f^4))*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3
*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C
*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d
^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2
)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^
4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C
^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*
a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8
*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2

```
*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d
*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - ((c
 + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^10*f^3 - 16*B^2*a^2*d^10*f^3 + 16*C^
2*a^2*d^10*f^3 + 32*A^2*a^2*c^2*d^8*f^3 - 32*A^2*a^2*c^6*d^4*f^3 - 16*A^2*a
^2*c^8*d^2*f^3 - 32*B^2*a^2*c^2*d^8*f^3 + 32*B^2*a^2*c^6*d^4*f^3 + 16*B^2*a
^2*c^8*d^2*f^3 + 32*C^2*a^2*c^2*d^8*f^3 - 32*C^2*a^2*c^6*d^4*f^3 - 16*C^2*a
^2*c^8*d^2*f^3 - 32*A*C*a^2*d^10*f^3 - 64*A*B*a^2*c*d^9*f^3 + 64*B*C*a^2*c*
d^9*f^3 - 192*A*B*a^2*c^3*d^7*f^3 - 192*A*B*a^2*c^5*d^5*f^3 - 64*A*B*a^2*c^
7*d^3*f^3 - 64*A*C*a^2*c^2*d^8*f^3 + 64*A*C*a^2*c^6*d^4*f^3 + 32*A*C*a^2*c^
8*d^2*f^3 + 192*B*C*a^2*c^3*d^7*f^3 + 192*B*C*a^2*c^5*d^5*f^3 + 64*B*C*a^2*
c^7*d^3*f^3) - (-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f
^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*
a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^
2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 +
16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4
- 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4
 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^
2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*
A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^
2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d
^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)
*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^
2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2
+ 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A
*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 4
8*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4
- 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a
^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*
A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*
f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 +
24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2
*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c
^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 32*B*a*
d^12*f^4 + 256*A*a*c^3*d^9*f^4 + 384*A*a*c^5*d^7*f^4 + 256*A*a*c^7*d^5*f^4
+ 64*A*a*c^9*d^3*f^4 + 96*B*a*c^2*d^10*f^4 + 64*B*a*c^4*d^8*f^4 - 64*B*a*c^
6*d^6*f^4 - 96*B*a*c^8*d^4*f^4 - 32*B*a*c^10*d^2*f^4 - 256*C*a*c^3*d^9*f^4
- 384*C*a*c^5*d^7*f^4 - 256*C*a*c^7*d^5*f^4 - 64*C*a*c^9*d^3*f^4 + 64*A*a*c
*d^11*f^4 - 64*C*a*c*d^11*f^4))*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2*a^2*c^3*f^2
+ 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2 + 16*B*C*a^2*
d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2*a^2*c*d^2*f^
2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c^2*d*f^2)^2/4
 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)*(A^4*a^4 + B
^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^
4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2 - 4*B^2*a^2*c
^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3*f^2 + 8*B*C*
```

```
a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12*C^2*a^2*c*d^
2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a^2*c^2*d*f^2)
/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/2) - 16*A^3*a
^3*d^9*f^2 + 16*C^3*a^3*d^9*f^2 - 48*A^3*a^3*c^2*d^7*f^2 - 48*A^3*a^3*c^4*d
^5*f^2 - 16*A^3*a^3*c^6*d^3*f^2 + 48*B^3*a^3*c^3*d^6*f^2 + 48*B^3*a^3*c^5*d
^4*f^2 + 16*B^3*a^3*c^7*d^2*f^2 + 48*C^3*a^3*c^2*d^7*f^2 + 48*C^3*a^3*c^4*d
^5*f^2 + 16*C^3*a^3*c^6*d^3*f^2 - 16*A*B^2*a^3*d^9*f^2 - 48*A*C^2*a^3*d^9*f
^2 + 48*A^2*C*a^3*d^9*f^2 + 16*B^2*C*a^3*d^9*f^2 + 16*B^3*a^3*c*d^8*f^2 - 4
8*A*B^2*a^3*c^2*d^7*f^2 - 48*A*B^2*a^3*c^4*d^5*f^2 - 16*A*B^2*a^3*c^6*d^3*f
^2 + 48*A^2*B*a^3*c^3*d^6*f^2 + 48*A^2*B*a^3*c^5*d^4*f^2 + 16*A^2*B*a^3*c^7
*d^2*f^2 - 144*A*C^2*a^3*c^2*d^7*f^2 - 144*A*C^2*a^3*c^4*d^5*f^2 - 48*A*C^2
*a^3*c^6*d^3*f^2 + 144*A^2*C*a^3*c^2*d^7*f^2 + 144*A^2*C*a^3*c^4*d^5*f^2 +
48*A^2*C*a^3*c^6*d^3*f^2 + 48*B*C^2*a^3*c^3*d^6*f^2 + 48*B*C^2*a^3*c^5*d^4*
f^2 + 16*B*C^2*a^3*c^7*d^2*f^2 + 48*B^2*C*a^3*c^2*d^7*f^2 + 48*B^2*C*a^3*c^
4*d^5*f^2 + 16*B^2*C*a^3*c^6*d^3*f^2 + 16*A^2*B*a^3*c*d^8*f^2 + 16*B*C^2*a^
3*c*d^8*f^2 - 96*A*B*C*a^3*c^3*d^6*f^2 - 96*A*B*C*a^3*c^5*d^4*f^2 - 32*A*B*
C*a^3*c^7*d^2*f^2 - 32*A*B*C*a^3*c*d^8*f^2))*(-(((8*A^2*a^2*c^3*f^2 - 8*B^2
*a^2*c^3*f^2 + 8*C^2*a^2*c^3*f^2 - 16*A*B*a^2*d^3*f^2 - 16*A*C*a^2*c^3*f^2
+ 16*B*C*a^2*d^3*f^2 - 24*A^2*a^2*c*d^2*f^2 + 24*B^2*a^2*c*d^2*f^2 - 24*C^2
*a^2*c*d^2*f^2 + 48*A*B*a^2*c^2*d*f^2 + 48*A*C*a^2*c*d^2*f^2 - 48*B*C*a^2*c
^2*d*f^2)^2/4 - (16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4)
*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 +
 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^3*f^2
- 4*B^2*a^2*c^3*f^2 + 4*C^2*a^2*c^3*f^2 - 8*A*B*a^2*d^3*f^2 - 8*A*C*a^2*c^3
*f^2 + 8*B*C*a^2*d^3*f^2 - 12*A^2*a^2*c*d^2*f^2 + 12*B^2*a^2*c*d^2*f^2 - 12
*C^2*a^2*c*d^2*f^2 + 24*A*B*a^2*c^2*d*f^2 + 24*A*C*a^2*c*d^2*f^2 - 24*B*C*a
^2*c^2*d*f^2)/(16*(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4)))^(1/
2)*2i + (2*(C*b*c^3 + A*b*c*d^2 - B*b*c^2*d))/(d^2*f*(c^2 + d^2)*(c + d*tan
(e + f*x))^(1/2)) - (2*(A*a*d^2 + C*a*c^2 - B*a*c*d))/(d*f*(c^2 + d^2)*(c +
 d*tan(e + f*x))^(1/2)) + (2*C*b*(c + d*tan(e + f*x))^(1/2))/(d^2*f)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a + b\tan\left(e + fx\right)\right)\left(A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)\right)}{\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
)**(3/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(c +
 d*tan(e + f*x))**(3/2), x)
```

3.119 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=157

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)}{df\left(c^2+d^2\right)\sqrt{c+d\tan(e+fx)}}-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{3/2}}-\frac{(B-i(A-C))\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(c+id)^{3/2}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d)^(3/2)/f-
(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(c+I*d)^(3/2)/f-2
*(A*d^2-B*c*d+C*c^2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] time = 0.29, antiderivative size = 157, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 5, integrand size = 35, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.143, Rules used = {3628, 3539, 3537, 63, 208}

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)}{df\left(c^2+d^2\right)\sqrt{c+d\tan(e+fx)}}-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{3/2}}-\frac{(B-i(A-C))\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{f(c+id)^{3/2}}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((c - I
*d)^(3/2)*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c +
I*d]])/((c + I*d)^(3/2)*f) - (2*(c^2*C - B*c*d + A*d^2))/(d*(c^2 + d^2)*f*S
qrt[c + d*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$\begin{aligned}
\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{3/2}}\,dx &= -\frac{2\left(c^2C-Bcd+Ad^2\right)}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+\frac{\int \frac{Ac-cC+Bd+(Bc-(A-C)d)\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,}{c^2+d^2} \\
&= -\frac{2\left(c^2C-Bcd+Ad^2\right)}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+\frac{(A-iB-C)\int \frac{1+i\tan(e+fx)}{\sqrt{c+d\tan(e+fx)}}\,dx}{2(c-id)} \\
&= -\frac{2\left(c^2C-Bcd+Ad^2\right)}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+\frac{(iA+B-iC)\,\mathrm{Subst}\left(\int \frac{1}{(-1+x)\sqrt{}}\right.}{2(c-id)} \\
&= -\frac{2\left(c^2C-Bcd+Ad^2\right)}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}-\frac{(A-iB-C)\,\mathrm{Subst}\left(\int \frac{1}{-1-\frac{ic}{d}+\frac{ix^2}{d}}\right.}{(c-id)} \\
&= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{3/2}f}-\frac{(B-i(A-C))\tanh^{-1}\left(\frac{\sqrt{c}}{}\right.}{(c+id)^{3/2}f}
\end{aligned}$$

Mathematica [C] time = 1.01, size = 218, normalized size = 1.39

$$\frac{\frac{(d(C-A)+Bc)\left((d-ic)\,{}_2F_1\left(-\frac{1}{2},1;\frac{1}{2};\frac{c+d\tan(e+fx)}{c-id}\right)+(d+ic)\,{}_2F_1\left(-\frac{1}{2},1;\frac{1}{2};\frac{c+d\tan(e+fx)}{c+id}\right)\right)}{(c^2+d^2)\sqrt{c+d\tan(e+fx)}}-iB\left(\frac{\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}}-\frac{\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c+id}}\right)}{\sqrt{c+id}}\right)-}{df}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^(3/2
),x]
```

```
[Out] ((-I)*B*(ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]]/Sqrt[c - I*d] - Ar
cTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]]/Sqrt[c + I*d]) - (2*C)/Sqrt[c
 + d*Tan[e + f*x]] + ((B*c + (-A + C)*d)*(((-I)*c + d)*Hypergeometric2F1[-1
/2, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)] + (I*c + d)*Hypergeometric2F1[-
1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)]))/((c^2 + d^2)*Sqrt[c + d*Tan[
e + f*x]]))/(d*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/2),x, algorit
hm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/2),x, algorit
hm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.36, size = 11427, normalized size = 72.78

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(3/2),x, algorit
hm="maxima")
```

```
[Out] Timed out
```

mupad [B] time = 19.61, size = 8588, normalized size = 54.70

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x))^(3/2),x)
```

```
[Out] (log((((((((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2)
 - 4*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3
*c^4*d^2*f^4))^(1/2)*(64*C*c*d^11*f^4 - ((c + d*tan(e + f*x))^(1/2)*(((96*C
^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) - 4*C^2*c^3*f^
2 + 12*C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^
(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5
 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5))/4 + 256*C*c^3*d^9*f^4 + 384*C*c^5*d^
7*f^4 + 256*C*c^7*d^5*f^4 + 64*C*c^9*d^3*f^4))/4 + (c + d*tan(e + f*x))^(1/
2)*(16*C^2*d^10*f^3 + 32*C^2*c^2*d^8*f^3 - 32*C^2*c^6*d^4*f^3 - 16*C^2*c^8*
d^2*f^3))*(((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/
2) - 4*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 +
 3*c^4*d^2*f^4))^(1/2))/4 - 8*C^3*d^9*f^2 - 24*C^3*c^2*d^7*f^2 - 24*C^3*c^4
*d^5*f^2 - 8*C^3*c^6*d^3*f^2)*(((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*
C^4*c^4*d^2*f^4)^(1/2) - 4*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f
^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/4 + (log(((((-((96*C^4*c^2*d^4*
f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*C^2*
c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(64*C
*c*d^11*f^4 - ((c + d*tan(e + f*x))^(1/2)*(-((96*C^4*c^2*d^4*f^4 - 16*C^4*d
^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*C^2*c*d^2*f^2)/(c^
6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(64*c*d^12*f^5 + 32
0*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c
^11*d^2*f^5))/4 + 256*C*c^3*d^9*f^4 + 384*C*c^5*d^7*f^4 + 256*C*c^7*d^5*f^4
 + 64*C*c^9*d^3*f^4))/4 + (c + d*tan(e + f*x))^(1/2)*(16*C^2*d^10*f^3 + 32*
C^2*c^2*d^8*f^3 - 32*C^2*c^6*d^4*f^3 - 16*C^2*c^8*d^2*f^3))*(-((96*C^4*c^2*
```

d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*
C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/
4 - 8*C^3*d^9*f^2 - 24*C^3*c^2*d^7*f^2 - 24*C^3*c^4*d^5*f^2 - 8*C^3*c^6*d^3
f^2)(-((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2)
+ 4*C^2*c^3*f^2 - 12*C^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*
c^4*d^2*f^4))^(1/2))/4 - log(((((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*
C^4*c^4*d^2*f^4)^(1/2) - 4*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(16*c^6*f^4 + 16
*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1
/2)*(((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) - 4
*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4
+ 48*c^4*d^2*f^4))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^
5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 64*C*c*d^11*f^4
+ 256*C*c^3*d^9*f^4 + 384*C*c^5*d^7*f^4 + 256*C*c^7*d^5*f^4 + 64*C*c^9*d^3*
f^4) - (c + d*tan(e + f*x))^(1/2)*(16*C^2*d^10*f^3 + 32*C^2*c^2*d^8*f^3 - 3
2*C^2*c^6*d^4*f^3 - 16*C^2*c^8*d^2*f^3))*(((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6
*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) - 4*C^2*c^3*f^2 + 12*C^2*c*d^2*f^2)/(16*c
^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2) - 8*C^3*d^9*f
^2 - 24*C^3*c^2*d^7*f^2 - 24*C^3*c^4*d^5*f^2 - 8*C^3*c^6*d^3*f^2)*(((96*C^4
*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) - 4*C^2*c^3*f^2
+ 12*C^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*
f^4))^(1/2) - log(((-((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^
2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*C^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 +
48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-((96
*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*
f^2 - 12*C^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*
d^2*f^4))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c
^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) + 64*C*c*d^11*f^4 + 256*C*c
^3*d^9*f^4 + 384*C*c^5*d^7*f^4 + 256*C*c^7*d^5*f^4 + 64*C*c^9*d^3*f^4) - (c
+ d*tan(e + f*x))^(1/2)*(16*C^2*d^10*f^3 + 32*C^2*c^2*d^8*f^3 - 32*C^2*c^6
*d^4*f^3 - 16*C^2*c^8*d^2*f^3))*(-((96*C^4*c^2*d^4*f^4 - 16*C^4*d^6*f^4 - 1
44*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*C^2*c*d^2*f^2)/(16*c^6*f^4 +
16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2) - 8*C^3*d^9*f^2 - 24*
C^3*c^2*d^7*f^2 - 24*C^3*c^4*d^5*f^2 - 8*C^3*c^6*d^3*f^2)*(-((96*C^4*c^2*d^
4*f^4 - 16*C^4*d^6*f^4 - 144*C^4*c^4*d^2*f^4)^(1/2) + 4*C^2*c^3*f^2 - 12*C^
2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(
1/2) + (log(8*A^3*d^9*f^2 - ((((((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144
*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^6*
f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(((((96*A^4*c^2*d^4*f^4 - 16*A^
4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^2*f^2)/
(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f
x))^(1/2)(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^
6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5))/4 + 64*A*c*d^11*f^4 + 256*A*c^3
*d^9*f^4 + 384*A*c^5*d^7*f^4 + 256*A*c^7*d^5*f^4 + 64*A*c^9*d^3*f^4))/4 - (
c + d*tan(e + f*x))^(1/2)*(16*A^2*d^10*f^3 + 32*A^2*c^2*d^8*f^3 - 32*A^2*c^
6*d^4*f^3 - 16*A^2*c^8*d^2*f^3))*(((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 1

```
44*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^
6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/4 + 24*A^3*c^2*d^7*f^2 + 24*
A^3*c^4*d^5*f^2 + 8*A^3*c^6*d^3*f^2)*(((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4
 - 144*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^2*f^2)/(c^6*f^4
+ d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/4 + (log(8*A^3*d^9*f^2 -
 ((((-((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) +
4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^
4*d^2*f^4))^(1/2)*(((-((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d
^2*f^4)^(1/2) + 4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^
2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5
 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 +
 64*c^11*d^2*f^5))/4 + 64*A*c*d^11*f^4 + 256*A*c^3*d^9*f^4 + 384*A*c^5*d^7*
f^4 + 256*A*c^7*d^5*f^4 + 64*A*c^9*d^3*f^4))/4 - (c + d*tan(e + f*x))^(1/2)
*(16*A^2*d^10*f^3 + 32*A^2*c^2*d^8*f^3 - 32*A^2*c^6*d^4*f^3 - 16*A^2*c^8*d^
2*f^3))*(-((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2
) + 4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 +
3*c^4*d^2*f^4))^(1/2))/4 + 24*A^3*c^2*d^7*f^2 + 24*A^3*c^4*d^5*f^2 + 8*A^3*
c^6*d^3*f^2)*(-((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)
^(1/2) + 4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f
^4 + 3*c^4*d^2*f^4))^(1/2))/4 - log(8*A^3*d^9*f^2 - ((((96*A^4*c^2*d^4*f^4
- 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^
2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*(
64*A*c*d^11*f^4 - (((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*
f^4)^(1/2) - 4*A^2*c^3*f^2 + 12*A^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 4
8*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^1
2*f^5 + 320*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*
f^5 + 64*c^11*d^2*f^5) + 256*A*c^3*d^9*f^4 + 384*A*c^5*d^7*f^4 + 256*A*c^7*
d^5*f^4 + 64*A*c^9*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*d^10*f^3 +
 32*A^2*c^2*d^8*f^3 - 32*A^2*c^6*d^4*f^3 - 16*A^2*c^8*d^2*f^3))*(((96*A^4*c
^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c^3*f^2 +
12*A^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^
4))^(1/2) + 24*A^3*c^2*d^7*f^2 + 24*A^3*c^4*d^5*f^2 + 8*A^3*c^6*d^3*f^2)*((
(96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) - 4*A^2*c
^3*f^2 + 12*A^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c
^4*d^2*f^4))^(1/2) - log(8*A^3*d^9*f^2 - ((-((96*A^4*c^2*d^4*f^4 - 16*A^4*d
^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) + 4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(16
*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*(64*A*c*d^1
1*f^4 - (-((96*A^4*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2
) + 4*A^2*c^3*f^2 - 12*A^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4
*f^4 + 48*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5 + 3
20*c^3*d^10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*
c^11*d^2*f^5) + 256*A*c^3*d^9*f^4 + 384*A*c^5*d^7*f^4 + 256*A*c^7*d^5*f^4 +
 64*A*c^9*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*d^10*f^3 + 32*A^2*c
^2*d^8*f^3 - 32*A^2*c^6*d^4*f^3 - 16*A^2*c^8*d^2*f^3))*(-((96*A^4*c^2*d^4*f
^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) + 4*A^2*c^3*f^2 - 12*A^2*c
```

```
*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2
) + 24*A^3*c^2*d^7*f^2 + 24*A^3*c^4*d^5*f^2 + 8*A^3*c^6*d^3*f^2)*(-((96*A^4
*c^2*d^4*f^4 - 16*A^4*d^6*f^4 - 144*A^4*c^4*d^2*f^4)^(1/2) + 4*A^2*c^3*f^2
- 12*A^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*
f^4))^(1/2) + (log(- (((c + d*tan(e + f*x))^(1/2)*(16*B^2*d^10*f^3 + 32*B^2
*c^2*d^8*f^3 - 32*B^2*c^6*d^4*f^3 - 16*B^2*c^8*d^2*f^3) + ((((96*B^4*c^2*d^
4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 12*B^
2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(((
((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*
c^3*f^2 - 12*B^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*
f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 +
640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5))/4 +
 32*B*d^12*f^4 + 96*B*c^2*d^10*f^4 + 64*B*c^4*d^8*f^4 - 64*B*c^6*d^6*f^4 -
96*B*c^8*d^4*f^4 - 32*B*c^10*d^2*f^4))/4)*(((96*B^4*c^2*d^4*f^4 - 16*B^4*d^
6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 12*B^2*c*d^2*f^2)/(c^6
*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/4 - 24*B^3*c^3*d^6*
f^2 - 24*B^3*c^5*d^4*f^2 - 8*B^3*c^7*d^2*f^2 - 8*B^3*c*d^8*f^2)*(((96*B^4*c
^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 -
12*B^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2
))/4 + (log(- (((c + d*tan(e + f*x))^(1/2)*(16*B^2*d^10*f^3 + 32*B^2*c^2*d^
8*f^3 - 32*B^2*c^6*d^4*f^3 - 16*B^2*c^8*d^2*f^3) + ((-((96*B^4*c^2*d^4*f^4
- 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*B^2*c*d^
2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2)*(((-((96*
B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f
^2 + 12*B^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))
^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 + 640*c
^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5))/4 + 32*B
*d^12*f^4 + 96*B*c^2*d^10*f^4 + 64*B*c^4*d^8*f^4 - 64*B*c^6*d^6*f^4 - 96*B*
c^8*d^4*f^4 - 32*B*c^10*d^2*f^4))/4)*(-((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^
4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*B^2*c*d^2*f^2)/(c^6*f^4
 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/4 - 24*B^3*c^3*d^6*f^2
- 24*B^3*c^5*d^4*f^2 - 8*B^3*c^7*d^2*f^2 - 8*B^3*c*d^8*f^2)*(-((96*B^4*c^2*
d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*
B^2*c*d^2*f^2)/(c^6*f^4 + d^6*f^4 + 3*c^2*d^4*f^4 + 3*c^4*d^2*f^4))^(1/2))/
4 - log(((c + d*tan(e + f*x))^(1/2)*(16*B^2*d^10*f^3 + 32*B^2*c^2*d^8*f^3 -
 32*B^2*c^6*d^4*f^3 - 16*B^2*c^8*d^2*f^3) + (((96*B^4*c^2*d^4*f^4 - 16*B^4*
d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 12*B^2*c*d^2*f^2)/(1
6*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*((((96*B^4
*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2
- 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*
f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^10*f^5 +
640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*f^5) - 32
*B*d^12*f^4 - 96*B*c^2*d^10*f^4 - 64*B*c^4*d^8*f^4 + 64*B*c^6*d^6*f^4 + 96*
B*c^8*d^4*f^4 + 32*B*c^10*d^2*f^4))*(((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4
- 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 12*B^2*c*d^2*f^2)/(16*c^6*f^
```

```
4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2) - 24*B^3*c^3*d^6*f
^2 - 24*B^3*c^5*d^4*f^2 - 8*B^3*c^7*d^2*f^2 - 8*B^3*c*d^8*f^2)*(((96*B^4*c^
2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) + 4*B^2*c^3*f^2 - 1
2*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4
))^(1/2) - log(((c + d*tan(e + f*x))^(1/2)*(16*B^2*d^10*f^3 + 32*B^2*c^2*d^
8*f^3 - 32*B^2*c^6*d^4*f^3 - 16*B^2*c^8*d^2*f^3) + (-((96*B^4*c^2*d^4*f^4 -
 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*B^2*c*d^2
*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2)*((
-((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2
*c^3*f^2 + 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48
*c^4*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^12*f^5 + 320*c^3*d^
10*f^5 + 640*c^5*d^8*f^5 + 640*c^7*d^6*f^5 + 320*c^9*d^4*f^5 + 64*c^11*d^2*
f^5) - 32*B*d^12*f^4 - 96*B*c^2*d^10*f^4 - 64*B*c^4*d^8*f^4 + 64*B*c^6*d^6*
f^4 + 96*B*c^8*d^4*f^4 + 32*B*c^10*d^2*f^4))*(-((96*B^4*c^2*d^4*f^4 - 16*B^
4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*c^3*f^2 + 12*B^2*c*d^2*f^2)/
(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*c^4*d^2*f^4))^(1/2) - 24*B^3
*c^3*d^6*f^2 - 24*B^3*c^5*d^4*f^2 - 8*B^3*c^7*d^2*f^2 - 8*B^3*c*d^8*f^2)*(-
((96*B^4*c^2*d^4*f^4 - 16*B^4*d^6*f^4 - 144*B^4*c^4*d^2*f^4)^(1/2) - 4*B^2*
c^3*f^2 + 12*B^2*c*d^2*f^2)/(16*c^6*f^4 + 16*d^6*f^4 + 48*c^2*d^4*f^4 + 48*
c^4*d^2*f^4))^(1/2) - (2*A*d)/(f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2)) +
(2*B*c)/(f*(c^2 + d^2)*(c + d*tan(e + f*x))^(1/2)) - (2*C*c^2)/(d*f*(c^2 +
d^2)*(c + d*tan(e + f*x))^(1/2))
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(c + d*tan(e + f*x))**(3/
2), x)
```

3.120 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=262

$$-\frac{2\sqrt{b}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{f\left(a^2+b^2\right)(bc-ad)^{3/2}}+\frac{2\left(Ad^2-Bcd+c^2C\right)}{f\left(c^2+d^2\right)(bc-ad)\sqrt{c+d\tan(e+fx)}}+\frac{(A-iB-C)\,\text{ta}}{f(b+}$$

```
[Out] (A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(I*a+b)/(c-I*d)^(3/
2)/f+(I*A-B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(a+I*b)/(c+I
*d)^(3/2)/f-2*(A*b^2-a*(B*b-C*a))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)/(-
a*d+b*c)^(1/2))*b^(1/2)/(a^2+b^2)/(-a*d+b*c)^(3/2)/f+2*(A*d^2-B*c*d+C*c^2)/
(-a*d+b*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] time = 1.28, antiderivative size = 262, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3649, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{2\sqrt{b}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{f\left(a^2+b^2\right)(bc-ad)^{3/2}}+\frac{2\left(Ad^2-Bcd+c^2C\right)}{f\left(c^2+d^2\right)(bc-ad)\sqrt{c+d\tan(e+fx)}}+\frac{(A-iB-C)\,\text{ta}}{f(b+}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c + d*Ta
n[e + f*x])^(3/2)),x]
```

```
[Out] ((A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((I*a + b)*
(c - I*d)^(3/2)*f) + ((I*A - B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt
[c + I*d]])/((a + I*b)*(c + I*d)^(3/2)*f) - (2*Sqrt[b]*(A*b^2 - a*(b*B - a*
C))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/((a^2 + b^
2)*(b*c - a*d)^(3/2)*f) + (2*(c^2*C - B*c*d + A*d^2))/((b*c - a*d)*(c^2 + d
^2)*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
```

```
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}\,dx = \frac{2\left(c^2C-Bcd+Ad^2\right)}{(bc-ad)\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{2\int \frac{\frac{1}{2}\left(-aAcd+ad(cC-B\right.}{}}{}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)}{(bc-ad)\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{\left(b\left(Ab^2-a(bB-a\right.\right.}{\left(\right.}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)}{(bc-ad)\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\int \frac{1+}{\sqrt{c+}}}{2(a-ib)(c}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)}{(bc-ad)\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{(iA+B-iC)\,\mathrm{Subs}}{}$$

$$= -\frac{2\sqrt{b}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\left(a^2+b^2\right)(bc-ad)^{3/2}f} + \frac{}{(bc-}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)(c-id)^{3/2}f} - \frac{(A+iB-C)\tanh^{-1}}{(ia-b)(c+}$$

Mathematica [A] time = 4.95, size = 296, normalized size = 1.13

$$\frac{\frac{2\sqrt{b}\left(c^2+d^2\right)\left(a(aC-bB)+Ab^2\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\left(a^2+b^2\right)\sqrt{bc-ad}} - \frac{i\left(\frac{(a+ib)(c+id)(A-iB-C)(ad-bc)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{\sqrt{c-id}} + \frac{(a-ib)(c-id)(A+iB-C)(bc-ad)\tanh^{-1}\left(\sqrt{\right.}}{\sqrt{c+id}}\right.}{a^2+b^2}}{f\left(c^2+d^2\right)(ad-bc)}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c
+ d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] (((-I)*(((a + I*b)*(A - I*B - C)*(c + I*d)*(-(b*c) + a*d)*ArcTanh[Sqrt[c +
d*Tan[e + f*x]]/Sqrt[c - I*d]])/Sqrt[c - I*d] + ((a - I*b)*(A + I*B - C)*(c
 - I*d)*(b*c - a*d)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/Sqrt[c
 + I*d]))/(a^2 + b^2) + (2*Sqrt[b]*(A*b^2 + a*(-(b*B) + a*C))*(c^2 + d^2)*A
rcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/((a^2 + b^2)*Sq
rt[b*c - a*d]) - (2*(c^2*C - B*c*d + A*d^2))/Sqrt[c + d*Tan[e + f*x]])/((-(
b*c) + a*d)*(c^2 + d^2)*f)
```

fricas [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [**B**] time = 0.62, size = 26343, normalized size = 100.55

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))^(3/2)
,x)
```

```
[Out] result too large to display
```

maxima [**F(-2)**] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(3/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))^(3/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)
)**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))*(c
+ d*tan(e + f*x))**(3/2)), x)
```

3.121 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=447

$$-\frac{d\left(A\left(2a^2d^2+b^2\left(c^2+3d^2\right)\right)+a^2\left(-2Bcd+3c^2C+Cd^2\right)-abB\left(c^2+d^2\right)+2b^2c(cC-Bd)\right)}{f\left(a^2+b^2\right)\left(c^2+d^2\right)(bc-ad)^2\sqrt{c+d\tan(e+fx)}}-\frac{}{f\left(a^2+b^2\right)(bc-ad}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a-I*b)^2/(c-I*d
)^(3/2)/f-(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(a+I*b)
^2/(c+I*d)^(3/2)/f-(5*a^3*b*B*d-3*a^4*C*d+b^4*(-3*A*d+2*B*c)+a*b^3*(4*A*c+B
*d-4*C*c)-a^2*b^2*(2*B*c+(7*A-C)*d))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^(1/2)
/(-a*d+b*c)^(1/2))*b^(1/2)/(a^2+b^2)^2/(-a*d+b*c)^(5/2)/f-d*(2*b^2*c*(-B*d+
C*c)-a*b*B*(c^2+d^2)+a^2*(-2*B*c*d+3*C*c^2+C*d^2)+A*(2*a^2*d^2+b^2*(c^2+3*d
^2)))/(a^2+b^2)/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)+(-A*b^2+a*(
B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
```

Rubi [A] time = 2.88, antiderivative size = 446, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3649, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{d\left(2a^2Ad^2+a^2\left(-2Bcd+3c^2C+Cd^2\right)-abB\left(c^2+d^2\right)+Ab^2\left(c^2+3d^2\right)+2b^2c(cC-Bd)\right)}{f\left(a^2+b^2\right)\left(c^2+d^2\right)(bc-ad)^2\sqrt{c+d\tan(e+fx)}}-\frac{}{f\left(a^2+b^2\right)(bc-ad}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(c + d*
Tan[e + f*x])^(3/2)),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((a - I
*b)^2*(c - I*d)^(3/2)*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x
]]/Sqrt[c + I*d]])/((a + I*b)^2*(c + I*d)^(3/2)*f) - (Sqrt[b]*(5*a^3*b*B*d
- 3*a^4*C*d + b^4*(2*B*c - 3*A*d) + a*b^3*(4*A*c - 4*c*C + B*d) - a^2*b^2*(
2*B*c + (7*A - C)*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c -
 a*d]])/((a^2 + b^2)^2*(b*c - a*d)^(5/2)*f) - (d*(2*a^2*A*d^2 + 2*b^2*c*(c*
C - B*d) - a*b*B*(c^2 + d^2) + A*b^2*(c^2 + 3*d^2) + a^2*(3*c^2*C - 2*B*c*d
 + C*d^2)))/((a^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x
]]) - (A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x
])*Sqrt[c + d*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
```

```
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{Ab^2-a(bB-aC)}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{d\left(2a^2Ad^2+2b^2c(cC-Bd)-abB\left(c^2+d^2\right)+Ab^2\left(c^2+3d\right.\right.}{\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f\sqrt{c+d\,\mathrm{ta}}}$$

$$= -\frac{d\left(2a^2Ad^2+2b^2c(cC-Bd)-abB\left(c^2+d^2\right)+Ab^2\left(c^2+3d\right.\right.}{\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f\sqrt{c+d\,\mathrm{ta}}}$$

$$= -\frac{d\left(2a^2Ad^2+2b^2c(cC-Bd)-abB\left(c^2+d^2\right)+Ab^2\left(c^2+3d\right.\right.}{\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f\sqrt{c+d\,\mathrm{ta}}}$$

$$= -\frac{d\left(2a^2Ad^2+2b^2c(cC-Bd)-abB\left(c^2+d^2\right)+Ab^2\left(c^2+3d\right.\right.}{\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f\sqrt{c+d\,\mathrm{ta}}}$$

$$= -\frac{\sqrt{b}\left(5a^3bBd-3a^4Cd+b^4(2Bc-3Ad)+ab^3(4Ac-4cC+\right.}{\left(a^2+b^2\right)^2(bc}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2(c-id)^{3/2}f} - \frac{(B-i(A-C))\,\mathrm{tanh}}{(a+ib)^2(c}$$

Mathematica [B] time = 6.29, size = 2078, normalized size = 4.65

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] -((A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])*S
qrt[c + d*Tan[e + f*x]])) - ((-2*(((I*Sqrt[c - I*d]*((b*(-(b*c) + a*d)*((-3
*(A*b^2 - a*(b*B - a*C))*d^2)/2 - c*(A*b - a*B - b*C)*(b*c - a*d) + (d*(3*A
*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2))/2 + a*(-1/2*(a
```

```
*d*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d)))
+ (((b*d^2)/2 - (c*(-(b*c) + a*d))/2)*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B
 - a*C)*(2*b*c + a*d)))/2 - (b*(-(c*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (
A*b - a*B - b*C)*d*(b*c - a*d))) + (d^2*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b
*B - a*C)*(2*b*c + a*d)))/2))/2) - I*((a*(-(b*c) + a*d)*((-3*(A*b^2 - a*(b*
B - a*C))*d^2)/2 - c*(A*b - a*B - b*C)*(b*c - a*d) + (d*(3*A*b^2*d - 2*a*A*
(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2))/2 - b*(-1/2*(a*d*((-3*c*(A*b^
2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d))) + (((b*d^2)/2 -
 (c*(-(b*c) + a*d))/2)*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c
+ a*d)))/2 - (b*(-(c*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C
)*d*(b*c - a*d))) + (d^2*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*
c + a*d)))/2))/2)))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((-c +
 I*d)*f) - (I*Sqrt[c + I*d]*((b*(-(b*c) + a*d)*((-3*(A*b^2 - a*(b*B - a*C))
*d^2)/2 - c*(A*b - a*B - b*C)*(b*c - a*d) + (d*(3*A*b^2*d - 2*a*A*(b*c - a*
d) - (b*B - a*C)*(2*b*c + a*d)))/2))/2 + a*(-1/2*(a*d*((-3*c*(A*b^2 - a*(b*
B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d))) + (((b*d^2)/2 - (c*(-(b*
c) + a*d))/2)*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/
2 - (b*(-(c*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c
- a*d))) + (d^2*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d))
)/2))/2) + I*((a*(-(b*c) + a*d)*((-3*(A*b^2 - a*(b*B - a*C))*d^2)/2 - c*(A*
b - a*B - b*C)*(b*c - a*d) + (d*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C
)*(2*b*c + a*d)))/2))/2 - b*(-1/2*(a*d*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2
+ (A*b - a*B - b*C)*d*(b*c - a*d))) + (((b*d^2)/2 - (c*(-(b*c) + a*d))/2)*(
3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2 - (b*(-(c*((-
3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d))) + (d^2
*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2))/2)))*ArcT
anh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c + I*d]])/((-c - I*d)*f))/(a^2 + b^2) +
(2*Sqrt[b*c - a*d]*(-1/2*(a*b*(-(b*c) + a*d)*((-3*(A*b^2 - a*(b*B - a*C))*d
^2)/2 - c*(A*b - a*B - b*C)*(b*c - a*d) + (d*(3*A*b^2*d - 2*a*A*(b*c - a*d)
 - (b*B - a*C)*(2*b*c + a*d)))/2)) + (a^2*b*(-(c*((-3*c*(A*b^2 - a*(b*B - a
*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d))) + (d^2*(3*A*b^2*d - 2*a*A*(b*
c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2))/2 + b^2*(-1/2*(a*d*((-3*c*(A*b^2
 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d))) + (((b*d^2)/2 -
(c*(-(b*c) + a*d))/2)*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c +
 a*d)))/2))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/(S
qrt[b]*(a^2 + b^2)*(-(b*c) + a*d)*f)))/((-(b*c) + a*d)*(c^2 + d^2)) - (2*(-
(c*((-3*c*(A*b^2 - a*(b*B - a*C))*d)/2 + (A*b - a*B - b*C)*d*(b*c - a*d)))
+ (d^2*(3*A*b^2*d - 2*a*A*(b*c - a*d) - (b*B - a*C)*(2*b*c + a*d)))/2))/((-
(b*c) + a*d)*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]]))/((a^2 + b^2)*(b*c - a
*d))
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.72, size = 40619, normalized size = 90.87

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e))^(3/
2),x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(3/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))^(3/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)^2 \left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**2/(c+d*tan(f*x
+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**2*
(c + d*tan(e + f*x))**(3/2)), x)
```

3.122 $\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=585

$$\frac{2b\sqrt{c+d\tan(e+fx)}\left(6a^2d^3\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+3abd\left(-c^2d^2(A-17C)+d^4(5A+3C)-2Bc^3d-8B\right.\right.}{3d^4f\left(c^2+d^2\right)^2}$$

```
[Out] -(a-I*b)^3*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d
)^(5/2)/f-(I*a-b)^3*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))
/(c+I*d)^(5/2)/f+2/3*b*(3*a*b*d*(8*c^4*C-2*B*c^3*d-c^2*(A-17*C)*d^2-8*B*c*d
^3+(5*A+3*C)*d^4)-b^2*(16*c^5*C-8*B*c^4*d+2*c^3*(A+15*C)*d^2-17*B*c^2*d^3+8
*c*(A+C)*d^4-3*B*d^5)+6*a^2*d^3*(2*c*(A-C)*d-B*(c^2-d^2)))*(c+d*tan(f*x+e))
^(1/2)/d^4/(c^2+d^2)^2/f+2/3*b^2*(b*(8*c^4*C-4*B*c^3*d+c^2*(A+15*C)*d^2-10*
B*c*d^3+(7*A+C)*d^4)+3*a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(c+d*tan(f*x+e))^(1
/2)*tan(f*x+e)/d^3/(c^2+d^2)^2/f-2*(b*(2*A*d^4-B*c^3*d-3*B*c*d^3+2*C*c^4+4*
C*c^2*d^2)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*x+e))^2/d^2/(c^2+d^2
)^2/f/(c+d*tan(f*x+e))^(1/2)-2/3*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^3/d/(
c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] time = 2.97, antiderivative size = 585, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}} = 0.149$, Rules used = {3645, 3637, 3630, 3539, 3537, 63, 208}

$$\frac{2b\sqrt{c+d\tan(e+fx)}\left(6a^2d^3\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+3abd\left(-c^2d^2(A-17C)+d^4(5A+3C)-2Bc^3d-8B\right.\right.}{3d^4f\left(c^2+d^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((a - I*b)^3*(I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/((c - I*d)^(5/2)*f)) - ((I*a - b)^3*(A + I*B - C)*ArcTanh[Sqrt[c + d*T
an[e + f*x]]/Sqrt[c + I*d]])/((c + I*d)^(5/2)*f) - (2*(c^2*C - B*c*d + A*d^
2)*(a + b*Tan[e + f*x])^3)/(3*d*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2)) -
 (2*(b*(2*c^4*C - B*c^3*d + 4*c^2*C*d^2 - 3*B*c*d^3 + 2*A*d^4) + a*d^2*(2*c
*(A - C)*d - B*(c^2 - d^2)))*(a + b*Tan[e + f*x])^2)/(d^2*(c^2 + d^2)^2*f*S
qrt[c + d*Tan[e + f*x]]) + (2*b*(3*a*b*d*(8*c^4*C - 2*B*c^3*d - c^2*(A - 17
*C)*d^2 - 8*B*c*d^3 + (5*A + 3*C)*d^4) - b^2*(16*c^5*C - 8*B*c^4*d + 2*c^3*
(A + 15*C)*d^2 - 17*B*c^2*d^3 + 8*c*(A + C)*d^4 - 3*B*d^5) + 6*a^2*d^3*(2*c
*(A - C)*d - B*(c^2 - d^2)))*Sqrt[c + d*Tan[e + f*x]])/(3*d^4*(c^2 + d^2)^2
```

```
*f) + (2*b^2*(b*(8*c^4*C - 4*B*c^3*d + c^2*(A + 15*C)*d^2 - 10*B*c*d^3 + (7
*A + C)*d^4) + 3*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Tan[e + f*x]*Sqrt[c
 + d*Tan[e + f*x]])/(3*d^3*(c^2 + d^2)^2*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
```

```
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}-$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}-$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}-$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}-$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}-$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^3}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}-$$

$$=-\frac{(a-ib)^3(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right.}{(c-id)^{5/2}f}$$

Mathematica [C] time = 6.91, size = 670, normalized size = 1.15

$$\frac{2C(a+b\tan(e+fx))^3}{3df(c+d\tan(e+fx))^{3/2}}+\frac{2\left(-\frac{3(-2aCd-bBd+2bcC)(a+b\tan(e+fx))^2}{df(c+d\tan(e+fx))^{3/2}}+\frac{2\left(-\frac{3(a+b\tan(e+fx))\left(bd^2(aB+Ab-bC)+4(bc-ad)(-2aCd-bBd+2bcC)\right)}{2df(c+d\tan(e+fx))^{3/2}}-\frac{3\ldots}{\ldots}\right)}{\ldots}\right)}{\ldots}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^3)/(3*d*f*(c + d*Tan[e + f*x])^(3/2)) + (2*((-3*(
2*b*c*C - b*B*d - 2*a*C*d)*(a + b*Tan[e + f*x])^2)/(d*f*(c + d*Tan[e + f*x]
)^(3/2)) + (2*((-3*(b*(A*b + a*B - b*C)*d^2 + 4*(b*c - a*d)*(2*b*c*C - b*B*
d - 2*a*C*d))*(a + b*Tan[e + f*x]))/(2*d*f*(c + d*Tan[e + f*x])^(3/2)) - (3
*((-2*(-16*b^3*c^3*C + 8*b^3*B*c^2*d + 48*a*b^2*c^2*C*d - 2*A*b^3*c*d^2 - 1
8*a*b^2*B*c*d^2 - 48*a^2*b*c*C*d^2 + 2*b^3*c*C*d^2 + 9*a^2*b*B*d^3 + b^3*B*
d^3 + 16*a^3*C*d^3))/(3*d*(c + d*Tan[e + f*x])^(3/2)) + (2*((((3*c*(a^3*B -
 3*a*b^2*B + 3*a^2*b*(A - C) - b^3*(A - C))*d^4)/2 + (3*(3*a^2*b*B - b^3*B
- a^3*(A - C) + 3*a*b^2*(A - C))*d^5)/2)*(-1/3*Hypergeometric2F1[-3/2, 1, -
1/2, (c + d*Tan[e + f*x])/(c - I*d)]/((I*c + d)*(c + d*Tan[e + f*x])^(3/2))
 + Hypergeometric2F1[-3/2, 1, -1/2, (c + d*Tan[e + f*x])/(c + I*d)]/(3*(I*c
 - d)*(c + d*Tan[e + f*x])^(3/2))))/d - (3*(a^3*B - 3*a*b^2*B + 3*a^2*b*(A
- C) - b^3*(A - C))*d^3*(-(Hypergeometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f
*x])/(c - I*d)]/((I*c + d)*Sqrt[c + d*Tan[e + f*x]])) + Hypergeometric2F1[-
1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)]/((I*c - d)*Sqrt[c + d*Tan[e +
f*x]])))/2))/(3*d)))/(4*d*f)))/d))/(3*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.62, size = 85156, normalized size = 145.57

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/
2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(5/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a + b\tan\left(e + fx\right)\right)^{3}\left(A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)\right)}{\left(c + d\tan\left(e + fx\right)\right)^{\frac{5}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(5/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**3*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x))**(5/2), x)
```

3.123 $\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=358

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)(a+b\tan(e+fx))^2}{3df\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}}+\frac{2(bc-ad)\left(3ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-2c^2d^2(A-5C)+\right.\right.}{3d^3f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(a-I*b)^2*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d
)^(5/2)/f-(a+I*b)^2*(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2
))/(c+I*d)^(5/2)/f+2/3*(-a*d+b*c)*(b*(4*c^4*C-B*c^3*d-2*c^2*(A-5*C)*d^2-7*B
*c*d^3+4*A*d^4)+3*a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))/d^3/(c^2+d^2)^2/f/(c+d*t
an(f*x+e))^(1/2)+2/3*b^2*(4*c^2*C-B*c*d+(A+3*C)*d^2)*(c+d*tan(f*x+e))^(1/2)
/d^3/(c^2+d^2)/f-2/3*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^2/d/(c^2+d^2)/f/(
c+d*tan(f*x+e))^(3/2)
```

Rubi [A] time = 1.55, antiderivative size = 358, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3645, 3635, 3630, 3539, 3537, 63, 208}

$$\frac{2(bc-ad)\left(3ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-2c^2d^2(A-5C)+4Ad^4-Bc^3d-7Bcd^3+4c^4C\right)\right)}{3d^3f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}}-\frac{2\left(Ad^2-Bc\right.}{3df\left(c^2\right.}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((a - I*b)^2*(I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*
d]])/((c - I*d)^(5/2)*f)) - ((a + I*b)^2*(B - I*(A - C))*ArcTanh[Sqrt[c + d
*Tan[e + f*x]]/Sqrt[c + I*d]])/((c + I*d)^(5/2)*f) - (2*(c^2*C - B*c*d + A*
d^2)*(a + b*Tan[e + f*x])^2)/(3*d*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2))
 + (2*(b*c - a*d)*(b*(4*c^4*C - B*c^3*d - 2*c^2*(A - 5*C)*d^2 - 7*B*c*d^3 +
 4*A*d^4) + 3*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2))))/(3*d^3*(c^2 + d^2)^2*
f*Sqrt[c + d*Tan[e + f*x]]) + (2*b^2*(4*c^2*C - B*c*d + (A + 3*C)*d^2)*Sqrt
[c + d*Tan[e + f*x]])/(3*d^3*(c^2 + d^2)*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
```

```
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$=-\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^2}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$=-\frac{(a-ib)^2(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right.}{(c-id)^{5/2}f}$$

Mathematica [C] time = 6.56, size = 502, normalized size = 1.40

$$\frac{2C(a+b\tan(e+fx))^2}{df(c+d\tan(e+fx))^{3/2}}+\frac{2\left(-\frac{(4aCd+bBd-4bcC)(a+b\tan(e+fx))}{df(c+d\tan(e+fx))^{3/2}}-\frac{-\frac{2(8a^2Cd^2+abBd^2-16abcCd-Ab^2d^2-2b^2Bcd+8b^2c^2C+b^2Cd^2)}{3d(c+d\tan(e+fx))^{3/2}}+\frac{2\left(\frac{3}{2}cd^3(a^2B\cdots\right.}{\cdots}}{\cdots}\right)}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] (2*C*(a + b*Tan[e + f*x])^2)/(d*f*(c + d*Tan[e + f*x])^(3/2)) + (2*(-(((-4*
b*c*C + b*B*d + 4*a*C*d)*(a + b*Tan[e + f*x]))/(d*f*(c + d*Tan[e + f*x])^(3
/2))) - ((-2*(8*b^2*c^2*C - 2*b^2*B*c*d - 16*a*b*c*C*d - A*b^2*d^2 + a*b*B*
d^2 + 8*a^2*C*d^2 + b^2*C*d^2))/(3*d*(c + d*Tan[e + f*x])^(3/2)) + (2*((((3
*c*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3)/2 + (3*(2*a*b*B - a^2*(A - C) + b^2
*(A - C))*d^4)/2)*(-1/3*Hypergeometric2F1[-3/2, 1, -1/2, (c + d*Tan[e + f*x
])/(c - I*d)]/((I*c + d)*(c + d*Tan[e + f*x])^(3/2)) + Hypergeometric2F1[-3
/2, 1, -1/2, (c + d*Tan[e + f*x])/(c + I*d)]/(3*(I*c - d)*(c + d*Tan[e + f*
x])^(3/2))))/d - (3*(a^2*B - b^2*B + 2*a*b*(A - C))*d^2*(-(Hypergeometric2F
1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)]/((I*c + d)*Sqrt[c + d*Tan[e
 + f*x]])) + Hypergeometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)
]/((I*c - d)*Sqrt[c + d*Tan[e + f*x]])))/2))/(3*d))/(2*d*f)))/d
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.52, size = 61833, normalized size = 172.72

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/
2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [B] time = 116.90, size = 88684, normalized size = 247.72

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^(5/2),x)
```

```
[Out] atan((((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^1
8*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d^14*f^3 + 1024*A^2*a^4*c^6*d
^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*a^4*c^10*d^8*f^3 + 320*A^2*a
^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320*A^2*b^4*c^4*d^14*f^3 + 1024
*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f^3 + 1024*A^2*b^4*c^10*d^8*f
^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16*d^2*f^3 - 256*A^2*a*b^3*c*d
^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a*b^3*c^3*d^15*f^3 - 2304*A^2
*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f^3 + 1280*A^2*a*b^3*c^9*d^9*
f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*b^3*c^13*d^5*f^3 + 256*A^2*a
*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3 + 2304*A^2*a^3*b*c^5*d^13*f
```

^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) + ((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*A*b^2*d^21*f^4 - 32*A*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*a^2*c^2*d^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4480*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4864*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 + 6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4 - 320*A*a*b*c^17*d^4*f^4 - 64

```
*A*a*b*c^19*d^2*f^4))*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^
2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*
b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^
4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c
^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^
2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4
*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4
*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4
*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2
*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4
*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c
^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*
a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 +
 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1
i + ((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^18*
f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d^14*f^3 + 1024*A^2*a^4*c^6*d^1
2*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*a^4*c^10*d^8*f^3 + 320*A^2*a^4
*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320*A^2*b^4*c^4*d^14*f^3 + 1024*A
^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f^3 + 1024*A^2*b^4*c^10*d^8*f^3
 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16*d^2*f^3 - 256*A^2*a*b^3*c*d^1
7*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a*b^3*c^3*d^15*f^3 - 2304*A^2*a
*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f^3 + 1280*A^2*a*b^3*c^9*d^9*f^
3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*b^3*c^13*d^5*f^3 + 256*A^2*a*b
^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3 + 2304*A^2*a^3*b*c^5*d^13*f^3
 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*
b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3 - 256*A^2*a^3*b*c^15*d^3*f^3 -
 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2*b^2*c^6*d^12*f^3 - 8640*A^2*a
^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d^8*f^3 - 1920*A^2*a^2*b^2*c^12
*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) - ((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4
*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3
*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f
^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f
^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*
c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4
*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c
^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4
*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*
c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*
d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*
f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*
c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c
^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^
8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c^5*f^2 + 8*A^
2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^
4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*
```

d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*A*a^2*d^21*f^4 + 32*A*b^2*d^21*f^4 - 160*A*a^2*c^2*d^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4480*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4864*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 + 6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^19*d^2*f^4))*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a*b^3*c^3*d^15*f^3 - 2304*A^2*a*b

^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*b^3*c^13*d^5*f^3 + 256*A^2*a*b^3 *c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b* c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1 920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2 *b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d ^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) + ((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c ^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d ^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^ 2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a ^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2 *d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A ^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^ 3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^ 5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^ 2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c* d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^1 0*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8* d^2*f^4)))^(1/2)*(32*A*b^2*d^21*f^4 - 32*A*a^2*d^21*f^4 - (c + d*tan(e + f* x))^(1/2)*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^ 2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160* A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^ 3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4 *a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 1 60*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5 *f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2 *f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 1 60*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^ 3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f ^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^ 5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^ 14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 28 80*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*a^2*c^2*d^19* f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13 *f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14 *d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d ^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8* d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4480*A*b^2*c^12*d^9*f^4 - 2432*A*b^2* c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c *d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4864*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 + 6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^19*d^2*f^4))*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a*b^3*c^3*d^15*f^3 - 2304*A^2*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*b^3*c^13*d^5*f^3 + 256*A^2*a*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) - ((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f

^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c
^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d
^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2
 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2
 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^
2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a
^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2
*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A
^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^
3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^
5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^
2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*
d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^1
0*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*
d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 76
80*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^1
0*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*
d^2*f^5) - 32*A*a^2*d^21*f^4 + 32*A*b^2*d^21*f^4 - 160*A*a^2*c^2*d^19*f^4 -
 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4
+ 4928*A*a^2*c^10*d^11*f^4 + 4480*A*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*
f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f
^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*
f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4480*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*
d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20
*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4864*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d
^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 + 6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c
^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^
19*d^2*f^4))*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5
*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f
^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 1
60*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^
2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2
*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*
A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4
+ 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*
c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*
d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^
2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2
- 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2
*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^
8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 64*A^3*a
^3*b^3*d^16*f^2 - 192*A^3*a^6*c^3*d^13*f^2 - 480*A^3*a^6*c^5*d^11*f^2 - 640
*A^3*a^6*c^7*d^9*f^2 - 480*A^3*a^6*c^9*d^7*f^2 - 192*A^3*a^6*c^11*d^5*f^2 -
 32*A^3*a^6*c^13*d^3*f^2 + 192*A^3*b^6*c^3*d^13*f^2 + 480*A^3*b^6*c^5*d^11*
f^2 + 640*A^3*b^6*c^7*d^9*f^2 + 480*A^3*b^6*c^9*d^7*f^2 + 192*A^3*b^6*c^11*

d^5*f^2 + 32*A^3*b^6*c^13*d^3*f^2 - 32*A^3*a*b^5*d^16*f^2 - 32*A^3*a^5*b*d^16*f^2 - 32*A^3*a^6*c*d^15*f^2 + 32*A^3*b^6*c*d^15*f^2 - 160*A^3*a*b^5*c^2*d^14*f^2 - 288*A^3*a*b^5*c^4*d^12*f^2 - 160*A^3*a*b^5*c^6*d^10*f^2 + 160*A^3*a*b^5*c^8*d^8*f^2 + 288*A^3*a*b^5*c^10*d^6*f^2 + 160*A^3*a*b^5*c^12*d^4*f^2 + 32*A^3*a*b^5*c^14*d^2*f^2 + 32*A^3*a^2*b^4*c*d^15*f^2 - 32*A^3*a^4*b^2*c*d^15*f^2 - 160*A^3*a^5*b*c^2*d^14*f^2 - 288*A^3*a^5*b*c^4*d^12*f^2 - 160*A^3*a^5*b*c^6*d^10*f^2 + 160*A^3*a^5*b*c^8*d^8*f^2 + 288*A^3*a^5*b*c^10*d^6*f^2 + 160*A^3*a^5*b*c^12*d^4*f^2 + 32*A^3*a^5*b*c^14*d^2*f^2 + 192*A^3*a^2*b^4*c^3*d^13*f^2 + 480*A^3*a^2*b^4*c^5*d^11*f^2 + 640*A^3*a^2*b^4*c^7*d^9*f^2 + 480*A^3*a^2*b^4*c^9*d^7*f^2 + 192*A^3*a^2*b^4*c^11*d^5*f^2 + 32*A^3*a^2*b^4*c^13*d^3*f^2 - 320*A^3*a^3*b^3*c^2*d^14*f^2 - 576*A^3*a^3*b^3*c^4*d^12*f^2 - 320*A^3*a^3*b^3*c^6*d^10*f^2 + 320*A^3*a^3*b^3*c^8*d^8*f^2 + 576*A^3*a^3*b^3*c^10*d^6*f^2 + 320*A^3*a^3*b^3*c^12*d^4*f^2 + 64*A^3*a^3*b^3*c^14*d^2*f^2 - 192*A^3*a^4*b^2*c^3*d^13*f^2 - 480*A^3*a^4*b^2*c^5*d^11*f^2 - 640*A^3*a^4*b^2*c^7*d^9*f^2 - 480*A^3*a^4*b^2*c^9*d^7*f^2 - 192*A^3*a^4*b^2*c^11*d^5*f^2 - 32*A^3*a^4*b^2*c^13*d^3*f^2))*((((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*A^2*a^4*c^5*f^2 - 4*A^2*b^4*c^5*f^2 + 24*A^2*a^2*b^2*c^5*f^2 + 40*A^2*a^4*c^3*d^2*f^2 + 40*A^2*b^4*c^3*d^2*f^2 + 16*A^2*a*b^3*d^5*f^2 - 16*A^2*a^3*b*d^5*f^2 - 20*A^2*a^4*c*d^4*f^2 - 20*A^2*b^4*c*d^4*f^2 + 80*A^2*a*b^3*c^4*d*f^2 - 80*A^2*a^3*b*c^4*d*f^2 - 160*A^2*a*b^3*c^2*d^3*f^2 + 120*A^2*a^2*b^2*c*d^4*f^2 + 160*A^2*a^3*b*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i + atan((((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a*b^3*c^3*d^15*f^3 - 2304*A^2*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*b^3*c^13*d^5*f^3 + 256*A^2*a*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) + (-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a

```
^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3
*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^
4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2
*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2
/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(1
6*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f
^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^
2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^
2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*
c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3
*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*
A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*
d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*A*b^2*d^21*f^4 - 32*A
*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b
^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c
^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4
*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d
*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*
b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A
^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80
*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) +
 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^
4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*
b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*
d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^
2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*
(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*
c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5
+ 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13
*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c
^21*d^2*f^5) - 160*A*a^2*c^2*d^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*
c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*
a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*
a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*
b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 448
0*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 9
6*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 486
4*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 +
6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4
 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^19*d^2*f^4))*(-(((8*A^2*a^4*c^5*f^2
+ 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*
A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*
a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^
3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320
*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*
```

b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10
*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)
)^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 -
40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16
*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a
*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A
^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*
f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4
*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*
b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*
d^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2
*a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 32
0*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*
f^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^1
6*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*
a*b^3*c^3*d^15*f^3 - 2304*A^2*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*
f^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a
*b^3*c^13*d^5*f^3 + 256*A^2*a*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^
3 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^
3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3
 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^
2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*
d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) - (-
(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*
a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^
3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c
^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^
2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^
2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(
16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*
f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A
^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A
^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4
*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^
3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240
*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4
*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1
/2)*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 8
0*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*
A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a
*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*
A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2
*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*
b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^
6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2

- 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2
- 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A
^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^
2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2
 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 +
10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 6
40*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^
5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^
17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*A*a^2*d^21*f^4 + 32*A
*b^2*d^21*f^4 - 160*A*a^2*c^2*d^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2
*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A
*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A
*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A
*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 44
80*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 -
96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 48
64*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 +
 6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^
4 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^19*d^2*f^4))*(-(((8*A^2*a^4*c^5*f^2
 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80
*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2
*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a
^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 32
0*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4
*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^1
0*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4
))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 -
 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 1
6*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*
a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*
A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2
f^2)/(16(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^
4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2
*b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4
*d^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^
2*a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 3
20*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10
*f^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^
16*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2
*a*b^3*c^3*d^15*f^3 - 2304*A^2*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11
*f^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*
a*b^3*c^13*d^5*f^3 + 256*A^2*a*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f
^3 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a
^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^
3 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a

^2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10
*d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) + (
-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2
*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a
^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*
c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a
^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)
^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*
(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4
*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*
A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*
A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^
4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b
^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 24
0*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^
4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*A*b^2*d^21*f^4 - 32
*A*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2
*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4
*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d
^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4
*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^
3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4
*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 +
80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)
 + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*
a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^
3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^
4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*
b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(1
6*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 +
5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^
5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^
13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64
*c^21*d^2*f^5) - 160*A*a^2*c^2*d^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^
2*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*
A*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*
A*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*
A*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4
480*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 -
 96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4
864*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4
+ 6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f
^4 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^19*d^2*f^4))*(-(((8*A^2*a^4*c^5*f^
2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 8
0*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^

2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(96*A^2*a^2*b^2*d^18*f^3 - 16*A^2*b^4*d^18*f^3 - 16*A^2*a^4*d^18*f^3 + 320*A^2*a^4*c^4*d^14*f^3 + 1024*A^2*a^4*c^6*d^12*f^3 + 1440*A^2*a^4*c^8*d^10*f^3 + 1024*A^2*a^4*c^10*d^8*f^3 + 320*A^2*a^4*c^12*d^6*f^3 - 16*A^2*a^4*c^16*d^2*f^3 + 320*A^2*b^4*c^4*d^14*f^3 + 1024*A^2*b^4*c^6*d^12*f^3 + 1440*A^2*b^4*c^8*d^10*f^3 + 1024*A^2*b^4*c^10*d^8*f^3 + 320*A^2*b^4*c^12*d^6*f^3 - 16*A^2*b^4*c^16*d^2*f^3 - 256*A^2*a*b^3*c*d^17*f^3 + 256*A^2*a^3*b*c*d^17*f^3 - 1280*A^2*a*b^3*c^3*d^15*f^3 - 2304*A^2*a*b^3*c^5*d^13*f^3 - 1280*A^2*a*b^3*c^7*d^11*f^3 + 1280*A^2*a*b^3*c^9*d^9*f^3 + 2304*A^2*a*b^3*c^11*d^7*f^3 + 1280*A^2*a*b^3*c^13*d^5*f^3 + 256*A^2*a*b^3*c^15*d^3*f^3 + 1280*A^2*a^3*b*c^3*d^15*f^3 + 2304*A^2*a^3*b*c^5*d^13*f^3 + 1280*A^2*a^3*b*c^7*d^11*f^3 - 1280*A^2*a^3*b*c^9*d^9*f^3 - 2304*A^2*a^3*b*c^11*d^7*f^3 - 1280*A^2*a^3*b*c^13*d^5*f^3 - 256*A^2*a^3*b*c^15*d^3*f^3 - 1920*A^2*a^2*b^2*c^4*d^14*f^3 - 6144*A^2*a^2*b^2*c^6*d^12*f^3 - 8640*A^2*a^2*b^2*c^8*d^10*f^3 - 6144*A^2*a^2*b^2*c^10*d^8*f^3 - 1920*A^2*a^2*b^2*c^12*d^6*f^3 + 96*A^2*a^2*b^2*c^16*d^2*f^3) - (-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2* f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*A*a^2*d^21*f^4 + 32*A*b^2*d^21*f^4 - 160*A*a^2*c^2*d^19*f^4 - 128*A*a^2*c^4*d^17*f^4 + 896*A*a^2*c^6*d^15*f^4 + 3136*A*a^2*c^8*d^13*f^4 + 4928*A*a^2*c^10*d^11*f^4 + 4480*A*a^2*c^12*d^9*f^4 + 2432*A*a^2*c^14*d^7*f^4 + 736*A*a^2*c^16*d^5*f^4 + 96*A*a^2*c^18*d^3*f^4 + 160*A*b^2*c^2*d^19*f^4 + 128*A*b^2*c^4*d^17*f^4 - 896*A*b^2*c^6*d^15*f^4 - 3136*A*b^2*c^8*d^13*f^4 - 4928*A*b^2*c^10*d^11*f^4 - 4480*A*b^2*c^12*d^9*f^4 - 2432*A*b^2*c^14*d^7*f^4 - 736*A*b^2*c^16*d^5*f^4 - 96*A*b^2*c^18*d^3*f^4 + 192*A*a*b*c*d^20*f^4 + 1472*A*a*b*c^3*d^18*f^4 + 4864*A*a*b*c^5*d^16*f^4 + 8960*A*a*b*c^7*d^14*f^4 + 9856*A*a*b*c^9*d^12*f^4 + 6272*A*a*b*c^11*d^10*f^4 + 1792*A*a*b*c^13*d^8*f^4 - 256*A*a*b*c^15*d^6*f^4 - 320*A*a*b*c^17*d^4*f^4 - 64*A*a*b*c^19*d^2*f^4))*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 64*A^3*a^3*b^3*d^16*f^2 - 192*A^3*a^6*c^3*d^13*f^2 - 480*A^3*a^6*c^5*d^11*f^2 - 640*A^3*a^6*c^7*d^9*f^2 - 480*A^3*a^6*c^9*d^7*f^2 - 192*A^3*a^6*c^11*d^5*f^2 - 32*A^3*a^6*c^13*d^3*f^2 + 192*A^3*b^6*c^3*d^13*f^2 + 480*A^3*b^6*c^5*d^11*f^2 + 640*A^3*b^6*c^7*d^9*f^2 + 480*A^3*b^6*c^9*d^7*f^2 + 192*A^3*b^6*c^11*d^5*f^2 + 32*A^3*b^6*c^13*d^3*f^2 - 32*A^3*a*b^5*d^16*f^2 - 32*A^3*a^5*b*d^16*f^2 - 32*A^3*a^6*c*d^15*f^2 + 32*A^3*b^6*c*d^15*f^2 - 160*A^3*a*b^5*c^2*d^14*f^2 - 288*A^3*a*b^5*c^4*d^12*f^2 - 160*A^3*a*b^5*c^6*d^10*f^2 + 160*A^3*a*b^5*c^8*d^8*f^2 + 288*A^3*a*b^5*c^10*d^6*f^2 + 160*A^3*a*b^5*c^12*d^4*f^2 + 32*A^3*a*b^5*c^14*d^2*f^2 + 32*A^3*a^2*b^4*c*d^15*f^2 - 32*A^3*a^4*b^2*c*d^15*f^2 - 160*A^3*a^5*b*c^2*d^1

4*f^2 - 288*A^3*a^5*b*c^4*d^12*f^2 - 160*A^3*a^5*b*c^6*d^10*f^2 + 160*A^3*a^5*b*c^8*d^8*f^2 + 288*A^3*a^5*b*c^10*d^6*f^2 + 160*A^3*a^5*b*c^12*d^4*f^2 + 32*A^3*a^5*b*c^14*d^2*f^2 + 192*A^3*a^2*b^4*c^3*d^13*f^2 + 480*A^3*a^2*b^4*c^5*d^11*f^2 + 640*A^3*a^2*b^4*c^7*d^9*f^2 + 480*A^3*a^2*b^4*c^9*d^7*f^2 + 192*A^3*a^2*b^4*c^11*d^5*f^2 + 32*A^3*a^2*b^4*c^13*d^3*f^2 - 320*A^3*a^3*b^3*c^2*d^14*f^2 - 576*A^3*a^3*b^3*c^4*d^12*f^2 - 320*A^3*a^3*b^3*c^6*d^10*f^2 + 320*A^3*a^3*b^3*c^8*d^8*f^2 + 576*A^3*a^3*b^3*c^10*d^6*f^2 + 320*A^3*a^3*b^3*c^12*d^4*f^2 + 64*A^3*a^3*b^3*c^14*d^2*f^2 - 192*A^3*a^4*b^2*c^3*d^13*f^2 - 480*A^3*a^4*b^2*c^5*d^11*f^2 - 640*A^3*a^4*b^2*c^7*d^9*f^2 - 480*A^3*a^4*b^2*c^9*d^7*f^2 - 192*A^3*a^4*b^2*c^11*d^5*f^2 - 32*A^3*a^4*b^2*c^13*d^3*f^2))*(-(((8*A^2*a^4*c^5*f^2 + 8*A^2*b^4*c^5*f^2 - 48*A^2*a^2*b^2*c^5*f^2 - 80*A^2*a^4*c^3*d^2*f^2 - 80*A^2*b^4*c^3*d^2*f^2 - 32*A^2*a*b^3*d^5*f^2 + 32*A^2*a^3*b*d^5*f^2 + 40*A^2*a^4*c*d^4*f^2 + 40*A^2*b^4*c*d^4*f^2 - 160*A^2*a*b^3*c^4*d*f^2 + 160*A^2*a^3*b*c^4*d*f^2 + 320*A^2*a*b^3*c^2*d^3*f^2 - 240*A^2*a^2*b^2*c*d^4*f^2 - 320*A^2*a^3*b*c^2*d^3*f^2 + 480*A^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (A^4*a^8 + A^4*b^8 + 4*A^4*a^2*b^6 + 6*A^4*a^4*b^4 + 4*A^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*A^2*a^4*c^5*f^2 + 4*A^2*b^4*c^5*f^2 - 24*A^2*a^2*b^2*c^5*f^2 - 40*A^2*a^4*c^3*d^2*f^2 - 40*A^2*b^4*c^3*d^2*f^2 - 16*A^2*a*b^3*d^5*f^2 + 16*A^2*a^3*b*d^5*f^2 + 20*A^2*a^4*c*d^4*f^2 + 20*A^2*b^4*c*d^4*f^2 - 80*A^2*a*b^3*c^4*d*f^2 + 80*A^2*a^3*b*c^4*d*f^2 + 160*A^2*a*b^3*c^2*d^3*f^2 - 120*A^2*a^2*b^2*c*d^4*f^2 - 160*A^2*a^3*b*c^2*d^3*f^2 + 240*A^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - atan(-((((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160* c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*C*a^2*d^21*f^4 + 32*C*b^2*d^21*f^4 - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*

f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b
*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^
2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i - (
((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2
*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a
^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*
c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a
^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)
^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*
(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4
*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*
C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*
C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^
4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b
^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 24
0*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^
4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*b^2*d^21*f^4 - 32
*C*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c^5*f^2 + 8*C^2*
b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*
c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^
4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*
d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3
*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*
C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 8
0*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)
- 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a
^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3
*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4
*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b
^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16
*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5
*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5
 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^1
3*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*
c^21*d^2*f^5) - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2
*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C
*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C
*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C
*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 44
80*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 -
96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 48
64*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 +
 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^
4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) + (c + d*tan(e + f*x))^
(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3

+ 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d
^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4
*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440
*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^
3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^
17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C
^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^
7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2
*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11
*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*
a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14
*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144
*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*
c^16*d^2*f^3))*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c
^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5
*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 -
160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*
f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b
^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 +
4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^
4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^
4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^
3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*
f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^
2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c
^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*
d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/((((
((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a
^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3
*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^
4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2
*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2
/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(1
6*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f
^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^
2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^
2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*
c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3
*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*
C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*
d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/
2)*(((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*
C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^
2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b
^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^

2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*C*a^2*d^21*f^4 + 32*C*b^2*d^21*f^4 - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*

c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*
d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^
2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2
- 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2
*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^
8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) + (((((8*C
^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^
3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^
5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f
^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*
c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 -
(C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^1
0*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 +
80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2
*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b
^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4
*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*
d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a
^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f
^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*b^2*d^21*f^4 - 32*C*a^2*
d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5
*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2
*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 +
40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 +
320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*
d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2
*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d
^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2
*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*
d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*
f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2
- 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^
4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*
f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^
2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680
*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*
f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^
2*f^5) - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^
15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^
12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^
18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^
6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^
2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^
2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*
b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C

*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320
*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) + (c + d*tan(e + f*x))^(1/2)*(
96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C
^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3
 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d
^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^
4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*
C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3
 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^
3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 +
 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*
c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 -
1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c
^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 -
6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^
2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^
2*f^3))*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2
 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 +
32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^
2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 2
40*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*
d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a
^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160
*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f
^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f
^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 2
0*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160
*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*
f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4
 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 64*C^3*a^3*b^
3*d^16*f^2 - 192*C^3*a^6*c^3*d^13*f^2 - 480*C^3*a^6*c^5*d^11*f^2 - 640*C^3*
a^6*c^7*d^9*f^2 - 480*C^3*a^6*c^9*d^7*f^2 - 192*C^3*a^6*c^11*d^5*f^2 - 32*C
^3*a^6*c^13*d^3*f^2 + 192*C^3*b^6*c^3*d^13*f^2 + 480*C^3*b^6*c^5*d^11*f^2 +
 640*C^3*b^6*c^7*d^9*f^2 + 480*C^3*b^6*c^9*d^7*f^2 + 192*C^3*b^6*c^11*d^5*f
^2 + 32*C^3*b^6*c^13*d^3*f^2 - 32*C^3*a*b^5*d^16*f^2 - 32*C^3*a^5*b*d^16*f^
2 - 32*C^3*a^6*c*d^15*f^2 + 32*C^3*b^6*c*d^15*f^2 - 160*C^3*a*b^5*c^2*d^14*
f^2 - 288*C^3*a*b^5*c^4*d^12*f^2 - 160*C^3*a*b^5*c^6*d^10*f^2 + 160*C^3*a*b
^5*c^8*d^8*f^2 + 288*C^3*a*b^5*c^10*d^6*f^2 + 160*C^3*a*b^5*c^12*d^4*f^2 +
32*C^3*a*b^5*c^14*d^2*f^2 + 32*C^3*a^2*b^4*c*d^15*f^2 - 32*C^3*a^4*b^2*c*d^
15*f^2 - 160*C^3*a^5*b*c^2*d^14*f^2 - 288*C^3*a^5*b*c^4*d^12*f^2 - 160*C^3*
a^5*b*c^6*d^10*f^2 + 160*C^3*a^5*b*c^8*d^8*f^2 + 288*C^3*a^5*b*c^10*d^6*f^2
 + 160*C^3*a^5*b*c^12*d^4*f^2 + 32*C^3*a^5*b*c^14*d^2*f^2 + 192*C^3*a^2*b^4
*c^3*d^13*f^2 + 480*C^3*a^2*b^4*c^5*d^11*f^2 + 640*C^3*a^2*b^4*c^7*d^9*f^2
+ 480*C^3*a^2*b^4*c^9*d^7*f^2 + 192*C^3*a^2*b^4*c^11*d^5*f^2 + 32*C^3*a^2*b
^4*c^13*d^3*f^2 - 320*C^3*a^3*b^3*c^2*d^14*f^2 - 576*C^3*a^3*b^3*c^4*d^12*f

^2 - 320*C^3*a^3*b^3*c^6*d^10*f^2 + 320*C^3*a^3*b^3*c^8*d^8*f^2 + 576*C^3*a
^3*b^3*c^10*d^6*f^2 + 320*C^3*a^3*b^3*c^12*d^4*f^2 + 64*C^3*a^3*b^3*c^14*d^
2*f^2 - 192*C^3*a^4*b^2*c^3*d^13*f^2 - 480*C^3*a^4*b^2*c^5*d^11*f^2 - 640*C
^3*a^4*b^2*c^7*d^9*f^2 - 480*C^3*a^4*b^2*c^9*d^7*f^2 - 192*C^3*a^4*b^2*c^11
*d^5*f^2 - 32*C^3*a^4*b^2*c^13*d^3*f^2))*((((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*
c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*
d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^
2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^
2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c
^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*
a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^
2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*
C^2*a^4*c^5*f^2 - 4*C^2*b^4*c^5*f^2 + 24*C^2*a^2*b^2*c^5*f^2 + 40*C^2*a^4*c
^3*d^2*f^2 + 40*C^2*b^4*c^3*d^2*f^2 + 16*C^2*a*b^3*d^5*f^2 - 16*C^2*a^3*b*d
^5*f^2 - 20*C^2*a^4*c*d^4*f^2 - 20*C^2*b^4*c*d^4*f^2 + 80*C^2*a*b^3*c^4*d*f
^2 - 80*C^2*a^3*b*c^4*d*f^2 - 160*C^2*a*b^3*c^2*d^3*f^2 + 120*C^2*a^2*b^2*c
*d^4*f^2 + 160*C^2*a^3*b*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^
10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8
*d^2*f^4)))^(1/2)*2i - atan(-(((-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 -
48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 -
32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^
2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C
^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^
2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 +
6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4
+ 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c
^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^
2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 +
20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C
^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2
- 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 +
d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)
))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5
*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2
*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 +
40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 +
320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*
d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2
*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d
^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2
*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*
d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*
f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2
+ 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^
4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*

f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*C*a^2*d^21*f^4 + 32*C*b^2*d^21*f^4 - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i - ((-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C

^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*b^2*d^21*f^4 - 32*C*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 23

04*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(((-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^

5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*C*a^2*d^21*f^4 + 32*C*b^2*d^21*f^4 - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 + 1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f

^4 + 5*c^8*d^2*f^4)))^(1/2) + ((-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*b^2*d^21*f^4 - 32*C*a^2*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*C*a^2*c^2*d^19*f^4 - 128*C*a^2*c^4*d^17*f^4 + 896*C*a^2*c^6*d^15*f^4 + 3136*C*a^2*c^8*d^13*f^4 + 4928*C*a^2*c^10*d^11*f^4 + 4480*C*a^2*c^12*d^9*f^4 + 2432*C*a^2*c^14*d^7*f^4 + 736*C*a^2*c^16*d^5*f^4 + 96*C*a^2*c^18*d^3*f^4 + 160*C*b^2*c^2*d^19*f^4 + 128*C*b^2*c^4*d^17*f^4 - 896*C*b^2*c^6*d^15*f^4 - 3136*C*b^2*c^8*d^13*f^4 - 4928*C*b^2*c^10*d^11*f^4 - 4480*C*b^2*c^12*d^9*f^4 - 2432*C*b^2*c^14*d^7*f^4 - 736*C*b^2*c^16*d^5*f^4 - 96*C*b^2*c^18*d^3*f^4 + 192*C*a*b*c*d^20*f^4 + 1472*C*a*b*c^3*d^18*f^4 + 4864*C*a*b*c^5*d^16*f^4 + 8960*C*a*b*c^7*d^14*f^4 + 9856*C*a*b*c^9*d^12*f^4 + 6272*C*a*b*c^11*d^10*f^4 + 1792*C*a*b*c^13*d^8*f^4 - 256*C*a*b*c^15*d^6*f^4 - 320*C*a*b*c^17*d^4*f^4 - 64*C*a*b*c^19*d^2*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*C^2*a^2*b^2*d^18*f^3 - 16*C^2*b^4*d^18*f^3 - 16*C^2*a^4*d^18*f^3 + 320*C^2*a^4*c^4*d^14*f^3 + 1024*C^2*a^4*c^6*d^12*f^3 + 1440*C^2*a^4*c^8*d^10*f^3 + 1024*C^2*a^4*c^10*d^8*f^3 + 320*C^2*a^4*c^12*d^6*f^3 - 16*C^2*a^4*c^16*d^2*f^3 + 320*C^2*b^4*c^4*d^14*f^3 +

1024*C^2*b^4*c^6*d^12*f^3 + 1440*C^2*b^4*c^8*d^10*f^3 + 1024*C^2*b^4*c^10*d^8*f^3 + 320*C^2*b^4*c^12*d^6*f^3 - 16*C^2*b^4*c^16*d^2*f^3 - 256*C^2*a*b^3*c*d^17*f^3 + 256*C^2*a^3*b*c*d^17*f^3 - 1280*C^2*a*b^3*c^3*d^15*f^3 - 2304*C^2*a*b^3*c^5*d^13*f^3 - 1280*C^2*a*b^3*c^7*d^11*f^3 + 1280*C^2*a*b^3*c^9*d^9*f^3 + 2304*C^2*a*b^3*c^11*d^7*f^3 + 1280*C^2*a*b^3*c^13*d^5*f^3 + 256*C^2*a*b^3*c^15*d^3*f^3 + 1280*C^2*a^3*b*c^3*d^15*f^3 + 2304*C^2*a^3*b*c^5*d^13*f^3 + 1280*C^2*a^3*b*c^7*d^11*f^3 - 1280*C^2*a^3*b*c^9*d^9*f^3 - 2304*C^2*a^3*b*c^11*d^7*f^3 - 1280*C^2*a^3*b*c^13*d^5*f^3 - 256*C^2*a^3*b*c^15*d^3*f^3 - 1920*C^2*a^2*b^2*c^4*d^14*f^3 - 6144*C^2*a^2*b^2*c^6*d^12*f^3 - 8640*C^2*a^2*b^2*c^8*d^10*f^3 - 6144*C^2*a^2*b^2*c^10*d^8*f^3 - 1920*C^2*a^2*b^2*c^12*d^6*f^3 + 96*C^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2 - 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 64*C^3*a^3*b^3*d^16*f^2 - 192*C^3*a^6*c^3*d^13*f^2 - 480*C^3*a^6*c^5*d^11*f^2 - 640*C^3*a^6*c^7*d^9*f^2 - 480*C^3*a^6*c^9*d^7*f^2 - 192*C^3*a^6*c^11*d^5*f^2 - 32*C^3*a^6*c^13*d^3*f^2 + 192*C^3*b^6*c^3*d^13*f^2 + 480*C^3*b^6*c^5*d^11*f^2 + 640*C^3*b^6*c^7*d^9*f^2 + 480*C^3*b^6*c^9*d^7*f^2 + 192*C^3*b^6*c^11*d^5*f^2 + 32*C^3*b^6*c^13*d^3*f^2 - 32*C^3*a*b^5*d^16*f^2 - 32*C^3*a^5*b*d^16*f^2 - 32*C^3*a^6*c*d^15*f^2 + 32*C^3*b^6*c*d^15*f^2 - 160*C^3*a*b^5*c^2*d^14*f^2 - 288*C^3*a*b^5*c^4*d^12*f^2 - 160*C^3*a*b^5*c^6*d^10*f^2 + 160*C^3*a*b^5*c^8*d^8*f^2 + 288*C^3*a*b^5*c^10*d^6*f^2 + 160*C^3*a*b^5*c^12*d^4*f^2 + 32*C^3*a*b^5*c^14*d^2*f^2 + 32*C^3*a^2*b^4*c*d^15*f^2 - 32*C^3*a^4*b^2*c*d^15*f^2 - 160*C^3*a^5*b*c^2*d^14*f^2 - 288*C^3*a^5*b*c^4*d^12*f^2 - 160*C^3*a^5*b*c^6*d^10*f^2 + 160*C^3*a^5*b*c^8*d^8*f^2 + 288*C^3*a^5*b*c^10*d^6*f^2 + 160*C^3*a^5*b*c^12*d^4*f^2 + 32*C^3*a^5*b*c^14*d^2*f^2 + 192*C^3*a^2*b^4*c^3*d^13*f^2 + 480*C^3*a^2*b^4*c^5*d^11*f^2 + 640*C^3*a^2*b^4*c^7*d^9*f^2 + 480*C^3*a^2*b^4*c^9*d^7*f^2 + 192*C^3*a^2*b^4*c^11*d^5*f^2 + 32*C^3*a^2*b^4*c^13*d^3*f^2 - 320*C^3*a^3*b^3*c^2*d^14*f^2 - 576*C^3*a^3*b^3*c^4*d^12*f^2 - 320*C^3*a^3*b^3*c^6*d^10*f^2 + 320*C^3*a^3*b^3*c^8*d^8*f^2 + 576*C^3*a^3*b^3*c^10*d^6*f^2 + 320*C^3*a^3*b^3*c^12*d^4*f^2 + 64*C^3*a^3*b^3*c^14*d^2*f^2 - 192*C^3*a^4*b^2*c^3*d^13*f^2 - 480*C^3*a^4*b^2*c^5*d^11*f^2 - 640*C^3*a^4*b^2*c^7*d^9*f^2 - 480*C^3*a^4*b^2*c^9*d^7*f^2 - 192*C^3*a^4*b^2*c^11*d^5*f^2 - 32*C^3*a^4*b^2*c^13*d^3*f^2))*(-(((8*C^2*a^4*c^5*f^2 + 8*C^2*b^4*c^5*f^2 - 48*C^2*a^2*b^2*c^5*f^2

- 80*C^2*a^4*c^3*d^2*f^2 - 80*C^2*b^4*c^3*d^2*f^2 - 32*C^2*a*b^3*d^5*f^2 + 32*C^2*a^3*b*d^5*f^2 + 40*C^2*a^4*c*d^4*f^2 + 40*C^2*b^4*c*d^4*f^2 - 160*C^2*a*b^3*c^4*d*f^2 + 160*C^2*a^3*b*c^4*d*f^2 + 320*C^2*a*b^3*c^2*d^3*f^2 - 240*C^2*a^2*b^2*c*d^4*f^2 - 320*C^2*a^3*b*c^2*d^3*f^2 + 480*C^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (C^4*a^8 + C^4*b^8 + 4*C^4*a^2*b^6 + 6*C^4*a^4*b^4 + 4*C^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*C^2*a^4*c^5*f^2 + 4*C^2*b^4*c^5*f^2 - 24*C^2*a^2*b^2*c^5*f^2 - 40*C^2*a^4*c^3*d^2*f^2 - 40*C^2*b^4*c^3*d^2*f^2 - 16*C^2*a*b^3*d^5*f^2 + 16*C^2*a^3*b*d^5*f^2 + 20*C^2*a^4*c*d^4*f^2 + 20*C^2*b^4*c*d^4*f^2 - 80*C^2*a*b^3*c^4*d*f^2 + 80*C^2*a^3*b*c^4*d*f^2 + 160*C^2*a*b^3*c^2*d^3*f^2 - 120*C^2*a^2*b^2*c*d^4*f^2 - 160*C^2*a^3*b*c^2*d^3*f^2 + 240*C^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - atan((((-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 96*B*a^2*c*d^20*f^4 + 96*B*b^2*c*d^20*f^4 - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*

B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 8
96*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 3
2*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4
480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4
 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4
 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256
*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9
856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4
 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) + (c + d*tan(e + f*x))
^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3
 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*
d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^
4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 144
0*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f
^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d
^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*
B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d
^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^
2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^1
1*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2
*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*d^1
4*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 614
4*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2
*c^16*d^2*f^3))*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2
*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d
^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2
 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^
3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2
*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4
+ 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*
f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*
b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*
c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^
4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*
f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b
*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^
2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i - (
(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^
2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*
a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3
*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*
a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2
)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)
*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^
4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24

*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(96*B*b^2*c*d^20*f^4 - 96*B*a^2*c*d^20*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4* d^14*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(16*B^3*b^6*d^16*f^2 - ((-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(96*B*b^2*c*d^20*f^4 - 96*B*a^2*c*d^20*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c

^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 16*B^3*a^6*d^16*f^2 - ((-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2* a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 96*B*a^2*c*d^20*f^4 + 96*B*b^2*c*d^20*f^4 - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c

^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3*d^15*f^3 + 23
04*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 1280*B^2*a^3*b*c^
9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13*d^5*f^3 - 256
*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 6144*B^2*a^2*b^2*
c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b^2*c^10*d^8*f^
3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f^3))*(-(((8*B^
2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3
*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5
*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^
2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c
*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (
B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10
*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 8
0*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*
b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^
3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*
f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d
^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^
2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^
4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) + 16*B^3*a^2*b^4*d^16*f^2 - 16*
B^3*a^4*b^2*d^16*f^2 - 80*B^3*a^6*c^2*d^14*f^2 - 144*B^3*a^6*c^4*d^12*f^2 -
 80*B^3*a^6*c^6*d^10*f^2 + 80*B^3*a^6*c^8*d^8*f^2 + 144*B^3*a^6*c^10*d^6*f^
2 + 80*B^3*a^6*c^12*d^4*f^2 + 16*B^3*a^6*c^14*d^2*f^2 + 80*B^3*b^6*c^2*d^14
*f^2 + 144*B^3*b^6*c^4*d^12*f^2 + 80*B^3*b^6*c^6*d^10*f^2 - 80*B^3*b^6*c^8*
d^8*f^2 - 144*B^3*b^6*c^10*d^6*f^2 - 80*B^3*b^6*c^12*d^4*f^2 - 16*B^3*b^6*c
^14*d^2*f^2 + 64*B^3*a*b^5*c*d^15*f^2 + 64*B^3*a^5*b*c*d^15*f^2 + 384*B^3*a
*b^5*c^3*d^13*f^2 + 960*B^3*a*b^5*c^5*d^11*f^2 + 1280*B^3*a*b^5*c^7*d^9*f^2
 + 960*B^3*a*b^5*c^9*d^7*f^2 + 384*B^3*a*b^5*c^11*d^5*f^2 + 64*B^3*a*b^5*c^
13*d^3*f^2 + 128*B^3*a^3*b^3*c*d^15*f^2 + 384*B^3*a^5*b*c^3*d^13*f^2 + 960*
B^3*a^5*b*c^5*d^11*f^2 + 1280*B^3*a^5*b*c^7*d^9*f^2 + 960*B^3*a^5*b*c^9*d^7
*f^2 + 384*B^3*a^5*b*c^11*d^5*f^2 + 64*B^3*a^5*b*c^13*d^3*f^2 + 80*B^3*a^2*
b^4*c^2*d^14*f^2 + 144*B^3*a^2*b^4*c^4*d^12*f^2 + 80*B^3*a^2*b^4*c^6*d^10*f
^2 - 80*B^3*a^2*b^4*c^8*d^8*f^2 - 144*B^3*a^2*b^4*c^10*d^6*f^2 - 80*B^3*a^2
*b^4*c^12*d^4*f^2 - 16*B^3*a^2*b^4*c^14*d^2*f^2 + 768*B^3*a^3*b^3*c^3*d^13*
f^2 + 1920*B^3*a^3*b^3*c^5*d^11*f^2 + 2560*B^3*a^3*b^3*c^7*d^9*f^2 + 1920*B
^3*a^3*b^3*c^9*d^7*f^2 + 768*B^3*a^3*b^3*c^11*d^5*f^2 + 128*B^3*a^3*b^3*c^1
3*d^3*f^2 - 80*B^3*a^4*b^2*c^2*d^14*f^2 - 144*B^3*a^4*b^2*c^4*d^12*f^2 - 80
*B^3*a^4*b^2*c^6*d^10*f^2 + 80*B^3*a^4*b^2*c^8*d^8*f^2 + 144*B^3*a^4*b^2*c^
10*d^6*f^2 + 80*B^3*a^4*b^2*c^12*d^4*f^2 + 16*B^3*a^4*b^2*c^14*d^2*f^2))*(-
(((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*
a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^
3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c
^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^
2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^
2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(

16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - 4*B^2*a^4*c^5*f^2 - 4*B^2*b^4*c^5*f^2 + 24*B^2*a^2*b^2*c^5*f^2 + 40*B^2*a^4*c^3*d^2*f^2 + 40*B^2*b^4*c^3*d^2*f^2 + 16*B^2*a*b^3*d^5*f^2 - 16*B^2*a^3*b*d^5*f^2 - 20*B^2*a^4*c*d^4*f^2 - 20*B^2*b^4*c*d^4*f^2 + 80*B^2*a*b^3*c^4*d*f^2 - 80*B^2*a^3*b*c^4*d*f^2 - 160*B^2*a*b^3*c^2*d^3*f^2 + 120*B^2*a^2*b^2*c*d^4*f^2 + 160*B^2*a^3*b*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - atan((((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 96*B*a^2*c*d^20*f^4 + 96*B*b^2*c*d^20*f^4 - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2

*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*
d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c
^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a
*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*
B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*
a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 +
1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*
f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c
^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2
*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1
280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c
^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 12
80*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3
*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 128
0*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13
*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 614
4*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b
^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f
^3))*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 8
0*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*
B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a
*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*
B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2
*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*
b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^
6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2
- 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2
- 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B
^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^
2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2
 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 +
10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i - (((((8*B^2*a^
4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2
*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2
 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 +
160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4
*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*
a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4
 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^
8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*
c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^
5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2
- 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f
^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^
2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 +

10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(96*B*b^2*c*d^20*f^4 - 96*B*a^2*c*d^20*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f^3))*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a *b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(16*B^3*b^6*d^16*f^2 - (((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(96*B*b^2*c*d^20*f^4 - 96*B*a^2*c*d^20*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 736*B*a^2*c^3*d^18*f^4 - 2432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4 - 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^4 + 160*B*a^2*c^17* d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4 - 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d^3*f^4) - (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^13*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*f^3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f^3))*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 16*B^3*a^6*d^16*f^2 - (((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2 + 20*

B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80*B^2*
a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^2 - 1
60*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4 + d^1
0*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^
(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*B^2*a^4*c^5*f^2 + 8*B^2*b^4*c^5*f^2
 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4*c^3*d^2*f^2
 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d^4*f^2 + 40*
B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4*d*f^2 + 320
*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^3*b*c^2*d^3*
f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4*B^4*a^2*b^6
 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f
^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 4*B^2*a^4
*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*a^4*c^3*d^2*
f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^3*b*d^5*f^2
+ 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^4*d*f^2 + 80
*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*b^2*c*d^4*f^
2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(16*(c^10*f^4
+ d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^
4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7
*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5
+ 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^
5) - 96*B*a^2*c*d^20*f^4 + 96*B*b^2*c*d^20*f^4 - 736*B*a^2*c^3*d^18*f^4 - 2
432*B*a^2*c^5*d^16*f^4 - 4480*B*a^2*c^7*d^14*f^4 - 4928*B*a^2*c^9*d^12*f^4
- 3136*B*a^2*c^11*d^10*f^4 - 896*B*a^2*c^13*d^8*f^4 + 128*B*a^2*c^15*d^6*f^
4 + 160*B*a^2*c^17*d^4*f^4 + 32*B*a^2*c^19*d^2*f^4 + 736*B*b^2*c^3*d^18*f^4
 + 2432*B*b^2*c^5*d^16*f^4 + 4480*B*b^2*c^7*d^14*f^4 + 4928*B*b^2*c^9*d^12*
f^4 + 3136*B*b^2*c^11*d^10*f^4 + 896*B*b^2*c^13*d^8*f^4 - 128*B*b^2*c^15*d^
6*f^4 - 160*B*b^2*c^17*d^4*f^4 - 32*B*b^2*c^19*d^2*f^4 - 64*B*a*b*d^21*f^4
- 320*B*a*b*c^2*d^19*f^4 - 256*B*a*b*c^4*d^17*f^4 + 1792*B*a*b*c^6*d^15*f^4
 + 6272*B*a*b*c^8*d^13*f^4 + 9856*B*a*b*c^10*d^11*f^4 + 8960*B*a*b*c^12*d^9
*f^4 + 4864*B*a*b*c^14*d^7*f^4 + 1472*B*a*b*c^16*d^5*f^4 + 192*B*a*b*c^18*d
^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(96*B^2*a^2*b^2*d^18*f^3 - 16*B^2*b^4*
d^18*f^3 - 16*B^2*a^4*d^18*f^3 + 320*B^2*a^4*c^4*d^14*f^3 + 1024*B^2*a^4*c^
6*d^12*f^3 + 1440*B^2*a^4*c^8*d^10*f^3 + 1024*B^2*a^4*c^10*d^8*f^3 + 320*B^
2*a^4*c^12*d^6*f^3 - 16*B^2*a^4*c^16*d^2*f^3 + 320*B^2*b^4*c^4*d^14*f^3 + 1
024*B^2*b^4*c^6*d^12*f^3 + 1440*B^2*b^4*c^8*d^10*f^3 + 1024*B^2*b^4*c^10*d^
8*f^3 + 320*B^2*b^4*c^12*d^6*f^3 - 16*B^2*b^4*c^16*d^2*f^3 - 256*B^2*a*b^3*
c*d^17*f^3 + 256*B^2*a^3*b*c*d^17*f^3 - 1280*B^2*a*b^3*c^3*d^15*f^3 - 2304*
B^2*a*b^3*c^5*d^13*f^3 - 1280*B^2*a*b^3*c^7*d^11*f^3 + 1280*B^2*a*b^3*c^9*d
^9*f^3 + 2304*B^2*a*b^3*c^11*d^7*f^3 + 1280*B^2*a*b^3*c^13*d^5*f^3 + 256*B^
2*a*b^3*c^15*d^3*f^3 + 1280*B^2*a^3*b*c^3*d^15*f^3 + 2304*B^2*a^3*b*c^5*d^1
3*f^3 + 1280*B^2*a^3*b*c^7*d^11*f^3 - 1280*B^2*a^3*b*c^9*d^9*f^3 - 2304*B^2
*a^3*b*c^11*d^7*f^3 - 1280*B^2*a^3*b*c^13*d^5*f^3 - 256*B^2*a^3*b*c^15*d^3*
f^3 - 1920*B^2*a^2*b^2*c^4*d^14*f^3 - 6144*B^2*a^2*b^2*c^6*d^12*f^3 - 8640*
B^2*a^2*b^2*c^8*d^10*f^3 - 6144*B^2*a^2*b^2*c^10*d^8*f^3 - 1920*B^2*a^2*b^2

*c^12*d^6*f^3 + 96*B^2*a^2*b^2*c^16*d^2*f^3))*((((8*B^2*a^4*c^5*f^2 + 8*B^2
*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B^2*b^4
*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a^4*c*d
^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3*b*c^4
*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*B^2*a^
3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b^8 + 4
*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*f^4 +
80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)
 + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 40*B^2*
a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*B^2*a^
3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*b^3*c^
4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^2*a^2*
b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f^2)/(1
6*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 +
5*c^8*d^2*f^4)))^(1/2) + 16*B^3*a^2*b^4*d^16*f^2 - 16*B^3*a^4*b^2*d^16*f^2
- 80*B^3*a^6*c^2*d^14*f^2 - 144*B^3*a^6*c^4*d^12*f^2 - 80*B^3*a^6*c^6*d^10*
f^2 + 80*B^3*a^6*c^8*d^8*f^2 + 144*B^3*a^6*c^10*d^6*f^2 + 80*B^3*a^6*c^12*d
^4*f^2 + 16*B^3*a^6*c^14*d^2*f^2 + 80*B^3*b^6*c^2*d^14*f^2 + 144*B^3*b^6*c^
4*d^12*f^2 + 80*B^3*b^6*c^6*d^10*f^2 - 80*B^3*b^6*c^8*d^8*f^2 - 144*B^3*b^6
*c^10*d^6*f^2 - 80*B^3*b^6*c^12*d^4*f^2 - 16*B^3*b^6*c^14*d^2*f^2 + 64*B^3*
a*b^5*c*d^15*f^2 + 64*B^3*a^5*b*c*d^15*f^2 + 384*B^3*a*b^5*c^3*d^13*f^2 + 9
60*B^3*a*b^5*c^5*d^11*f^2 + 1280*B^3*a*b^5*c^7*d^9*f^2 + 960*B^3*a*b^5*c^9*
d^7*f^2 + 384*B^3*a*b^5*c^11*d^5*f^2 + 64*B^3*a*b^5*c^13*d^3*f^2 + 128*B^3*
a^3*b^3*c*d^15*f^2 + 384*B^3*a^5*b*c^3*d^13*f^2 + 960*B^3*a^5*b*c^5*d^11*f^
2 + 1280*B^3*a^5*b*c^7*d^9*f^2 + 960*B^3*a^5*b*c^9*d^7*f^2 + 384*B^3*a^5*b*
c^11*d^5*f^2 + 64*B^3*a^5*b*c^13*d^3*f^2 + 80*B^3*a^2*b^4*c^2*d^14*f^2 + 14
4*B^3*a^2*b^4*c^4*d^12*f^2 + 80*B^3*a^2*b^4*c^6*d^10*f^2 - 80*B^3*a^2*b^4*c
^8*d^8*f^2 - 144*B^3*a^2*b^4*c^10*d^6*f^2 - 80*B^3*a^2*b^4*c^12*d^4*f^2 - 1
6*B^3*a^2*b^4*c^14*d^2*f^2 + 768*B^3*a^3*b^3*c^3*d^13*f^2 + 1920*B^3*a^3*b^
3*c^5*d^11*f^2 + 2560*B^3*a^3*b^3*c^7*d^9*f^2 + 1920*B^3*a^3*b^3*c^9*d^7*f^
2 + 768*B^3*a^3*b^3*c^11*d^5*f^2 + 128*B^3*a^3*b^3*c^13*d^3*f^2 - 80*B^3*a^
4*b^2*c^2*d^14*f^2 - 144*B^3*a^4*b^2*c^4*d^12*f^2 - 80*B^3*a^4*b^2*c^6*d^10
*f^2 + 80*B^3*a^4*b^2*c^8*d^8*f^2 + 144*B^3*a^4*b^2*c^10*d^6*f^2 + 80*B^3*a
^4*b^2*c^12*d^4*f^2 + 16*B^3*a^4*b^2*c^14*d^2*f^2))*((((8*B^2*a^4*c^5*f^2 +
 8*B^2*b^4*c^5*f^2 - 48*B^2*a^2*b^2*c^5*f^2 - 80*B^2*a^4*c^3*d^2*f^2 - 80*B
^2*b^4*c^3*d^2*f^2 - 32*B^2*a*b^3*d^5*f^2 + 32*B^2*a^3*b*d^5*f^2 + 40*B^2*a
^4*c*d^4*f^2 + 40*B^2*b^4*c*d^4*f^2 - 160*B^2*a*b^3*c^4*d*f^2 + 160*B^2*a^3
*b*c^4*d*f^2 + 320*B^2*a*b^3*c^2*d^3*f^2 - 240*B^2*a^2*b^2*c*d^4*f^2 - 320*
B^2*a^3*b*c^2*d^3*f^2 + 480*B^2*a^2*b^2*c^3*d^2*f^2)^2/4 - (B^4*a^8 + B^4*b
^8 + 4*B^4*a^2*b^6 + 6*B^4*a^4*b^4 + 4*B^4*a^6*b^2)*(16*c^10*f^4 + 16*d^10*
f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))
^(1/2) + 4*B^2*a^4*c^5*f^2 + 4*B^2*b^4*c^5*f^2 - 24*B^2*a^2*b^2*c^5*f^2 - 4
0*B^2*a^4*c^3*d^2*f^2 - 40*B^2*b^4*c^3*d^2*f^2 - 16*B^2*a*b^3*d^5*f^2 + 16*
B^2*a^3*b*d^5*f^2 + 20*B^2*a^4*c*d^4*f^2 + 20*B^2*b^4*c*d^4*f^2 - 80*B^2*a*
b^3*c^4*d*f^2 + 80*B^2*a^3*b*c^4*d*f^2 + 160*B^2*a*b^3*c^2*d^3*f^2 - 120*B^

```
2*a^2*b^2*c*d^4*f^2 - 160*B^2*a^3*b*c^2*d^3*f^2 + 240*B^2*a^2*b^2*c^3*d^2*f
^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*
f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - ((2*(A*a^2*d^2 + A*b^2*c^2 - 2*A*a*b*c*d)
)/(3*(c^2 + d^2)) - (4*d*(c + d*tan(e + f*x))*(A*a*b*c^2 - A*a*b*d^2 - A*a^
2*c*d + A*b^2*c*d))/(c^2 + d^2)^2)/(d*f*(c + d*tan(e + f*x))^(3/2)) - ((2*(
C*b^2*c^4 + C*a^2*c^2*d^2 - 2*C*a*b*c^3*d))/(3*(c^2 + d^2)) - (4*(c + d*tan
(e + f*x))*(C*b^2*c^5 + C*a^2*c*d^4 + 2*C*b^2*c^3*d^2 - C*a*b*c^4*d - 3*C*a
*b*c^2*d^3))/(c^2 + d^2)^2)/(d^3*f*(c + d*tan(e + f*x))^(3/2)) + ((2*(B*b^2
*c^3 + B*a^2*c*d^2 - 2*B*a*b*c^2*d))/(3*(c^2 + d^2)) - (2*(c + d*tan(e + f*
x))*(B*a^2*d^4 + B*b^2*c^4 - B*a^2*c^2*d^2 + 3*B*b^2*c^2*d^2 - 4*B*a*b*c*d^
3))/(c^2 + d^2)^2)/(d^2*f*(c + d*tan(e + f*x))^(3/2)) + (2*C*b^2*(c + d*tan
(e + f*x))^(1/2))/(d^3*f)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a + b\tan\left(e + fx\right)\right)^2 \left(A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)\right)}{\left(c + d\tan\left(e + fx\right)\right)^{\frac{5}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**(5/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**2*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x))**(5/2), x)
```

3.124 $\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=273

$$\frac{2(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{3d^2f\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}}-\frac{2\left(ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c\right.\right.}{d^2f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(a-I*b)*(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d)^
(5/2)/f+(I*a-b)*(A+I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(c+
I*d)^(5/2)/f-2*(b*(c^4*C-c^2*(A-3*C)*d^2-2*B*c*d^3+A*d^4)+a*d^2*(2*c*(A-C)*
d-B*(c^2-d^2)))/d^2/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)+2/3*(-a*d+b*c)*(A*
d^2-B*c*d+C*c^2)/d^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] time = 0.80, antiderivative size = 271, normalized size of antiderivative = 0.99, number of steps used = 9, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3635, 3628, 3539, 3537, 63, 208}

$$\frac{2(bc-ad)\left(Ad^2-Bcd+c^2C\right)}{3d^2f\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}}-\frac{2\left(ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c\right.\right.}{d^2f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d*T
an[e + f*x])^(5/2),x]
```

```
[Out] -(((I*a + b)*(A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])
/((c - I*d)^(5/2)*f)) + ((I*a - b)*(A + I*B - C)*ArcTanh[Sqrt[c + d*Tan[e +
 f*x]]/Sqrt[c + I*d]])/((c + I*d)^(5/2)*f) + (2*(b*c - a*d)*(c^2*C - B*c*d
+ A*d^2))/(3*d^2*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2)) - (2*(b*(c^4*C -
 c^2*(A - 3*C)*d^2 - 2*B*c*d^3 + A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d
^2))))/(d^2*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rule 3635

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.
)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> -Simp[((b*c - a*d)*(c^2*C - B*c*d + A*d^2)*(c +
d*Tan[e + f*x])^(n + 1))/(d^2*f*(n + 1)*(c^2 + d^2)), x] + Dist[1/(d*(c^2 +
 d^2)), Int[(c + d*Tan[e + f*x])^(n + 1)*Simp[a*d*(A*c - c*C + B*d) + b*(c^
2*C - B*c*d + A*d^2) + d*(A*b*c + a*B*c - b*c*C - a*A*d + b*B*d + a*C*d)*Ta
n[e + f*x] + b*C*(c^2 + d^2)*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c,
d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] && LtQ[n, -
1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{\int \frac{ad(A}{}}{}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2\left(b\left(c\right.\right.}{}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2\left(b\left(c\right.\right.}{}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2\left(b\left(c\right.\right.}{}$$

$$= \frac{2(bc-ad)\left(c^2C-Bcd+Ad^2\right)}{3d^2\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2\left(b\left(c\right.\right.}{}$$

$$= -\frac{(ia+b)(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{5/2}f}$$

Mathematica [C] time = 3.02, size = 300, normalized size = 1.10

$$-\frac{d(-aAd+aBc+aCd+Abc+bBd-bcC)\left(i(c+id)\,{}_2F_1\left(-\frac{3}{2},1;-\frac{1}{2};\frac{c+d\tan(e+fx)}{c-id}\right)-(d+ic)\,{}_2F_1\left(-\frac{3}{2},1;-\frac{1}{2};\frac{c+d\tan}{c}\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -1/3*(2*(c - I*d)*(c + I*d)*(2*b*c*C + b*B*d - 2*a*C*d) + d*(A*b*c + a*B*c
- b*c*C - a*A*d + b*B*d + a*C*d)*(I*(c + I*d)*Hypergeometric2F1[-3/2, 1, -1
/2, (c + d*Tan[e + f*x])/(c - I*d)] - (I*c + d)*Hypergeometric2F1[-3/2, 1,
-1/2, (c + d*Tan[e + f*x])/(c + I*d)]) + 6*C*(c - I*d)*(c + I*d)*d*(a + b*T
an[e + f*x]) - 3*(A*b + a*B - b*C)*d*(I*(c + I*d)*Hypergeometric2F1[-1/2, 1
, 1/2, (c + d*Tan[e + f*x])/(c - I*d)] - (I*c + d)*Hypergeometric2F1[-1/2,
1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)])*(c + d*Tan[e + f*x]))/(d^2*(c^2 +
d^2)*f*(c + d*Tan[e + f*x])^(3/2))
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.61, size = 40201, normalized size = 147.26

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/2)
,x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [B] time = 88.47, size = 64641, normalized size = 236.78

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d*t
an(e + f*x))^(5/2),x)

[Out] ((2*(C*b*c^3 + A*b*c*d^2 - B*b*c^2*d))/(3*(c^2 + d^2)) - (2*(c + d*tan(e +
f*x))*(A*b*d^4 + C*b*c^4 - 2*B*b*c*d^3 - A*b*c^2*d^2 + 3*C*b*c^2*d^2))/(c^2
 + d^2)^2)/(d^2*f*(c + d*tan(e + f*x))^(3/2)) - atan(-(((c + d*tan(e + f*x)
)^(1/2)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 -
320*A^2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^1
0*f^3 - 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c
^16*d^2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B
^2*b^2*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3
- 16*B^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^1
2*f^3 - 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2
*c^12*d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2
*c*d^17*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*
b^2*c^5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 115
2*A*B*b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^
3 + 640*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8
*d^10*f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b
^2*c^16*d^2*f^3 + 640*B*C*b^2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 64
0*B*C*b^2*c^7*d^11*f^3 - 640*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^
3 - 640*B*C*b^2*c^13*d^5*f^3 - 128*B*C*b^2*c^15*d^3*f^3) + ((((8*A^2*b^2*c^
5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80
*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C
*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4
*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 -
 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 +
 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4
 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 +
 C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^
2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 +
 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C
^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*
f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 +
40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B
*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^
10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8
*d^2*f^4)))^(1/2)*(128*A*b*c^15*d^6*f^4 - 32*B*b*d^21*f^4 - 736*A*b*c^3*d^1
8*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^12*f
^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^8*f^4 - (c + d*tan(e + f*x))^(
1/2)*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2
*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B
*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f
^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 8
0*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*
```

A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10
*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)
*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 +
6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2
- 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b
^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5
*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20
*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b
^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2
*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 1
0*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 +
2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^1
2*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c
^19*d^4*f^5 + 64*c^21*d^2*f^5) + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4
- 160*B*b*c^2*d^19*f^4 - 128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 313
6*B*b*c^8*d^13*f^4 + 4928*B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*
B*b*c^14*d^7*f^4 + 736*B*b*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3
*d^18*f^4 + 2432*C*b*c^5*d^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^
12*f^4 + 3136*C*b*c^11*d^10*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f
^4 - 160*C*b*c^17*d^4*f^4 - 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*
b*c*d^20*f^4))*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^
2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^
2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b
^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4
*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*
f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4
+ 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^
8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2
*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^
2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2
+ 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A
*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^
4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2
- 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 +
80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d
^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))
^(1/2)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 3
20*A^2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10
*f^3 - 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^
16*d^2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^
2*b^2*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 -
16*B^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12
*f^3 - 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*
c^12*d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*
c*d^17*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b

^2*c^5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152
*A*B*b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3
 + 640*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*
d^10*f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^
2*c^16*d^2*f^3 + 640*B*C*b^2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640
*B*C*b^2*c^7*d^11*f^3 - 640*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3
 - 640*B*C*b^2*c^13*d^5*f^3 - 128*B*C*b^2*c^15*d^3*f^3) - ((((8*A^2*b^2*c^5
*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*
B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*
b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*
f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 -
80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 +
160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4
+ 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 +
C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2
*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 +
4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^
2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f
^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 4
0*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*
b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^1
0*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*
d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^5*f^2 - 8*B^2*
b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d
^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 -
 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*
b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^
4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c
^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6
*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A
*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*
A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5
*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2
*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b
^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4
*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f
^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*
f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1
/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f
^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*
c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32
*B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7
*d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d
^8*f^4 + 128*A*b*c^15*d^6*f^4 + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4
- 160*B*b*c^2*d^19*f^4 - 128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136

```
*B*b*c^8*d^13*f^4 + 4928*B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B
*b*c^14*d^7*f^4 + 736*B*b*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*
d^18*f^4 + 2432*C*b*c^5*d^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^1
2*f^4 + 3136*C*b*c^11*d^10*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^
4 - 160*C*b*c^17*d^4*f^4 - 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b
*c*d^20*f^4))*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2
 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2
 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^
2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*
d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f
^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4
+ 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8
*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*
B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2
*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 +
 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*
C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4
*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 -
 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 8
0*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^
6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(((c + d*tan(e + f*x))^
(1/2)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 32
0*A^2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*
f^3 - 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^1
6*d^2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2
*b^2*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 -
16*B^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*
f^3 - 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c
^12*d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c
*d^17*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^
2*c^5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*
A*B*b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3
+ 640*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d
^10*f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2
*c^16*d^2*f^3 + 640*B*C*b^2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*
B*C*b^2*c^7*d^11*f^3 - 640*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3
- 640*B*C*b^2*c^13*d^5*f^3 - 128*B*C*b^2*c^15*d^3*f^3) - ((((8*A^2*b^2*c^5*
f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B
^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b
^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f
^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 8
0*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 1
60*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 +
 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C
^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*
```

C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^8*f^4 + 128*A*b*c^15*d^6*f^4 + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 - 128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^10*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4 - 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 +

40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3 - 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 - 16*B^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3 - 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^17*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 640*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^16*d^2*f^3 + 640*B*C*b^2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*b^2*c^7*d^11*f^3 - 640*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 640*B*C*b^2*c^13*d^5*f^3 - 128*B*C*b^2*c^15*d^3*f^3) + ((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(128*A*b*c^15*d^6*f^4 - 32*B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^8*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^

2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 - 128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^10*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4 - 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 16*A^3*b^3*d^16*f^2 + 16*C^3*b^3*d^16*f^2 - 80*A^3*b^3*c^2*d^14*f^2 - 144*A^3*b^3*c^4*d^12*f^2 - 80*A^3*b^3*c^6*d^10*f^2 + 80*A^3*b^3*c^8*d^8*f^2 + 144*A^3*b^3*c^10*d^6*f^2 + 80*A^3*b^3*c^12*d^4*f^2 + 16*A^3*b^3*c^14*d^2*f^2 + 192*B^3*b^3*c^3*d^13*f^2 + 480*B^3*b^3*c^5*d^11*f^2 + 640*B^3*b^3*c^7*d^9*f^2 + 480*B^3*b^3*c^9*d^7*f^2 + 192*B^3*b^3*c^11*d^5*f^2 + 32*B^3*b^3*c^13*d^3*f^2 + 80*C^3*b^3*c^2*d^14*f^2 + 144*C^3*b^3*c^4*d^12*f^2 + 80*C^3*b^3*c^6*d^10*f^2 - 80*C^3*b^3*c^8*d^8*f^2 - 144*C^3*b^3*c^10*d^6*f^2 - 80*C^3*b^3*c^12*d^4*f^2 - 16*C^3*b^3*c^14*d^2*f^2 - 16*A*B^2*b^3*d^16*f^2 - 48*A*C^2*b^3*d^16*f^2 + 48*A^2*C*b^3*d^16*f^2 + 16*B^2*C*b^3*d^16*f^2 + 32*B^3*b^3*c*d^15*f^2 - 80*A*B^2*b^3*c^2 *d^14*f^2 - 144*A*B^2*b^3*c^4*d^12*f^2 - 80*A*B^2*b^3*c^6*d^10*f^2 + 80*A*B^2*b^3*c^8*d^8*f^2 + 144*A*B^2*b^3*c^10*d^6*f^2 + 80*A*B^2*b^3*c^12*d^4*f^2 + 16*A*B^2*b^3*c^14*d^2*f^2 + 192*A^2*B*b^3*c^3*d^13*f^2 + 480*A^2*B*b^3*c^5*d^11*f^2 + 640*A^2*B*b^3*c^7*d^9*f^2 + 480*A^2*B*b^3*c^9*d^7*f^2 + 192*A^2*B*b^3*c^11*d^5*f^2 + 32*A^2*B*b^3*c^13*d^3*f^2 - 240*A*C^2*b^3*c^2*d^14*f^2 - 432*A*C^2*b^3*c^4*d^12*f^2 - 240*A*C^2*b^3*c^6*d^10*f^2 + 240*A*C^2*b^3*c^8*d^8*f^2 + 432*A*C^2*b^3*c^10*d^6*f^2 + 240*A*C^2*b^3*c^12*d^4*f^2 + 48*A*C^2*b^3*c^14*d^2*f^2 + 240*A^2*C*b^3*c^2*d^14*f^2 + 432*A^2*C*b^3*c^4*d^12*f^2 + 240*A^2*C*b^3*c^6*d^10*f^2 - 240*A^2*C*b^3*c^8*d^8*f^2 - 432*A^2*C*b^3*c^10*d^6*f^2 - 240*A^2*C*b^3*c^12*d^4*f^2 - 48*A^2*C*b^3*c^14*d^2*f^2 + 192*B*C^2*b^3*c^3*d^13*f^2 + 480*B*C^2*b^3*c^5*d^11*f^2 + 640*B*C^2*b^3*c^7*d^9*f^2 + 480*B*C^2*b^3*c^9*d^7*f^2 + 192*B*C^2*b^3*c^11*d^5*f^2 + 32*B*C^2*b^3*c^13*d^3*f^2 + 80*B^2*C*b^3*c^2*d^14*f^2 + 144*B^2*C*b^3*c^4*d^12*f^2 + 80*B^2*C*b^3*c^6*d^10*f^2 - 80*B^2*C*b^3*c^8*d^8*f^2 - 144*B^2*C*b^3*c^10*d^6*f^2 - 80*B^2*C*b^3*c^12*d^4*f^2 - 16*B^2*C*b^3*c^14*d^2*f^2 + 32*A^2*B*b^3*c*d^15*f^2 + 32*B*C^2*b^3*c*d^15*f^2 - 384*A*B*C*b^3*c^3*d^13*f^2 - 960*A*B*C*b^3*c^5*d^11*f^2 - 1280*A*B*C*b^3*c^7*d^9*f^2 - 960*A*B*C*b^3*c^9*d^7*f^2 - 384*A*B*C*b^3*c^11*d^5*f^2 - 64*A*B*C*b^3*c^13*d^3*f^2 - 64*A*B*C*b^3*c*d^15*f^2))*((((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) + 4*A^2*b^2*c^5*f^2 - 4*B^2*b^2*c^5*f^2 + 4*C^2*b^2*c^5*f^2 - 40*A^2*b^2*c^3*d^2*f^2 + 40*B^2*b^2*c^3*d^2*f^2 - 40*C^2*b^2*c^3*d^2*f^2 + 8*A*B*b^2*d^5*f^2 - 8*A*C*b^2*c^5*f^2 - 8*B*C*b^2*d^5*f^2 + 20*A^2*b^2*c*d^4*f^2 - 20*B^2*b^2*c*d^4*f^2 + 20*C^2*b^2*c*d^4*f^2 + 40*A*B*b^2*c^4*d*f^2 - 40*A*C*b^2*c*d^4*f^2 - 40*B*C*b^2*c^4*d*f^2 - 80*A*B*b^2*c^2*d^3*f^2 + 80*A*C*b^2*c^3*d^2*f^2 + 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - atan((((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C

```
*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d
^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2
 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2
)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^
4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^5*f^
2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2
*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2
*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2
 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*
B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160
*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 1
60*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4
*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^
2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C
^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a
^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2
- 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A
*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2
*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f
^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2
*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*
c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f
^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2
*f^5) - 32*A*a*d^21*f^4 + 32*C*a*d^21*f^4 - 160*A*a*c^2*d^19*f^4 - 128*A*a*
c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10
*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^
5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4
+ 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*d^12*f^4 + 3136*B*a*c^11*d^10*f^4 +
896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a
*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d
^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^
9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4
+ 96*B*a*c*d^20*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^18*f^3 - 16
*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A
^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3
 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^1
4*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^
2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C
^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3
 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d
^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*
f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^
7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*
a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 20
48*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8
```

```
*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3
*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*
a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 12
8*B*C*a^2*c^15*d^3*f^3))*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*
a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*
c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2
+ 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A
*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^
2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (1
6*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f
^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a
^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2)
- 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^
3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5
*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^
2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*
c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*
d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4
+ 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i - (((((8*A^2*
a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^
2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 -
16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2
*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4
*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2
*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d
^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4
*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4
+ 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5
*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2
+ 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^
2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*
f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 +
80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(
16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 +
 5*c^8*d^2*f^4)))^(1/2)*(32*C*a*d^21*f^4 - 32*A*a*d^21*f^4 - (c + d*tan(e +
 f*x))^(1/2)*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2
- 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2
+ 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2
*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d
*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^
2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 +
 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*
d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B
^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*
c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 -
```

40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a

^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d
^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2
*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 1
0*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/((((((8*A^2*a^2
*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 +
 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*
A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*
d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^
2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^
2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*
f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^
4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2
*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^
2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 4
0*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d
^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2
 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*
A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*
(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*
c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^5*f^2 - 8*
B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c
^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f
^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*
C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^
2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a
^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4
*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 -
 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4
 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2
*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3
*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A
^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2
*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d
^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d
^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))
)^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^
16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7
680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5)
 - 32*A*a*d^21*f^4 + 32*C*a*d^21*f^4 - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^
17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*
f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4
 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480
*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*
a*c^13*d^8*f^4 - 128*B*a*c^15*d^6*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*
d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^

4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B *a*c*d^20*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^2*a ^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2 *c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320 *A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10 *d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2 *c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 102 4*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11 *f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^ 13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C *a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15* f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^ 9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C* a^2*c^15*d^3*f^3))*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^ 5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^ 2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A ^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2 *c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2* d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10 *f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 8 0*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2 *A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^ 2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2* f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2* c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4* f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^ 2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c ^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) + (((((8*A^2*a^2*c^5*f ^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^ 2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^ 2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^ 2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80 *B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 16 0*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^ 4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C ^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4* C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2* a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2

- 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*a*d^21*f^4 - 32*A*a*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) - (c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A

*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 +
1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*
f^3 - 1152*B*C*a^2*c^11*d^7*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^
15*d^3*f^3))*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2
- 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2
+ 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2
*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d
*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^
2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 +
16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*
d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B
^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*
c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 -
40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C
*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*
f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 +
40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80
*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6
*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 16*B^3*a^3*d^16*f^2 - 192*
A^3*a^3*c^3*d^13*f^2 - 480*A^3*a^3*c^5*d^11*f^2 - 640*A^3*a^3*c^7*d^9*f^2 -
480*A^3*a^3*c^9*d^7*f^2 - 192*A^3*a^3*c^11*d^5*f^2 - 32*A^3*a^3*c^13*d^3*f
^2 - 80*B^3*a^3*c^2*d^14*f^2 - 144*B^3*a^3*c^4*d^12*f^2 - 80*B^3*a^3*c^6*d^
10*f^2 + 80*B^3*a^3*c^8*d^8*f^2 + 144*B^3*a^3*c^10*d^6*f^2 + 80*B^3*a^3*c^1
2*d^4*f^2 + 16*B^3*a^3*c^14*d^2*f^2 + 192*C^3*a^3*c^3*d^13*f^2 + 480*C^3*a^
3*c^5*d^11*f^2 + 640*C^3*a^3*c^7*d^9*f^2 + 480*C^3*a^3*c^9*d^7*f^2 + 192*C^
3*a^3*c^11*d^5*f^2 + 32*C^3*a^3*c^13*d^3*f^2 - 16*A^2*B*a^3*d^16*f^2 - 16*B
*C^2*a^3*d^16*f^2 - 32*A^3*a^3*c*d^15*f^2 + 32*C^3*a^3*c*d^15*f^2 - 192*A*B
^2*a^3*c^3*d^13*f^2 - 480*A*B^2*a^3*c^5*d^11*f^2 - 640*A*B^2*a^3*c^7*d^9*f^
2 - 480*A*B^2*a^3*c^9*d^7*f^2 - 192*A*B^2*a^3*c^11*d^5*f^2 - 32*A*B^2*a^3*c
^13*d^3*f^2 - 80*A^2*B*a^3*c^2*d^14*f^2 - 144*A^2*B*a^3*c^4*d^12*f^2 - 80*A
^2*B*a^3*c^6*d^10*f^2 + 80*A^2*B*a^3*c^8*d^8*f^2 + 144*A^2*B*a^3*c^10*d^6*f
^2 + 80*A^2*B*a^3*c^12*d^4*f^2 + 16*A^2*B*a^3*c^14*d^2*f^2 - 576*A*C^2*a^3*
c^3*d^13*f^2 - 1440*A*C^2*a^3*c^5*d^11*f^2 - 1920*A*C^2*a^3*c^7*d^9*f^2 - 1
440*A*C^2*a^3*c^9*d^7*f^2 - 576*A*C^2*a^3*c^11*d^5*f^2 - 96*A*C^2*a^3*c^13*
d^3*f^2 + 576*A^2*C*a^3*c^3*d^13*f^2 + 1440*A^2*C*a^3*c^5*d^11*f^2 + 1920*A
^2*C*a^3*c^7*d^9*f^2 + 1440*A^2*C*a^3*c^9*d^7*f^2 + 576*A^2*C*a^3*c^11*d^5*
f^2 + 96*A^2*C*a^3*c^13*d^3*f^2 - 80*B*C^2*a^3*c^2*d^14*f^2 - 144*B*C^2*a^3
*c^4*d^12*f^2 - 80*B*C^2*a^3*c^6*d^10*f^2 + 80*B*C^2*a^3*c^8*d^8*f^2 + 144*
B*C^2*a^3*c^10*d^6*f^2 + 80*B*C^2*a^3*c^12*d^4*f^2 + 16*B*C^2*a^3*c^14*d^2*
f^2 + 192*B^2*C*a^3*c^3*d^13*f^2 + 480*B^2*C*a^3*c^5*d^11*f^2 + 640*B^2*C*a
^3*c^7*d^9*f^2 + 480*B^2*C*a^3*c^9*d^7*f^2 + 192*B^2*C*a^3*c^11*d^5*f^2 + 3
2*B^2*C*a^3*c^13*d^3*f^2 + 32*A*B*C*a^3*d^16*f^2 - 32*A*B^2*a^3*c*d^15*f^2
- 96*A*C^2*a^3*c*d^15*f^2 + 96*A^2*C*a^3*c*d^15*f^2 + 32*B^2*C*a^3*c*d^15*f
^2 + 160*A*B*C*a^3*c^2*d^14*f^2 + 288*A*B*C*a^3*c^4*d^12*f^2 + 160*A*B*C*a^
3*c^6*d^10*f^2 - 160*A*B*C*a^3*c^8*d^8*f^2 - 288*A*B*C*a^3*c^10*d^6*f^2 - 1

60*A*B*C*a^3*c^12*d^4*f^2 - 32*A*B*C*a^3*c^14*d^2*f^2))*((((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) - 4*A^2*a^2*c^5*f^2 + 4*B^2*a^2*c^5*f^2 - 4*C^2*a^2*c^5*f^2 + 40*A^2*a^2*c^3*d^2*f^2 - 40*B^2*a^2*c^3*d^2*f^2 + 40*C^2*a^2*c^3*d^2*f^2 - 8*A*B*a^2*d^5*f^2 + 8*A*C*a^2*c^5*f^2 + 8*B*C*a^2*d^5*f^2 - 20*A^2*a^2*c*d^4*f^2 + 20*B^2*a^2*c*d^4*f^2 - 20*C^2*a^2*c*d^4*f^2 - 40*A*B*a^2*c^4*d*f^2 + 40*A*C*a^2*c*d^4*f^2 + 40*B*C*a^2*c^4*d*f^2 + 80*A*B*a^2*c^2*d^3*f^2 - 80*A*C*a^2*c^3*d^2*f^2 - 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - atan((((-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2

 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 +
 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*
d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*
d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 161
28*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*
f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*A*a*d^21*f^4 + 32*C*a*d^21*f
^4 - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3
136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 243
2*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c
^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*
d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6
*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 +
128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*
C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a
*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) + (c + d*tan(e + f
x))^(1/2)(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3
 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*
d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^
2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 144
0*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f
^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*
d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*
a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*
a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A
*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 +
1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3
*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*
c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*
C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 +
 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7
*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*(-(((8*A^2*a^2
*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 +
 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*
A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*
d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^
2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^
2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*
f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^
4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2
*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^
2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 4
0*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d
^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2
 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*
A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*

(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*
c^8*d^2*f^4)))^(1/2)*1i - ((-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C
^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a
^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f
^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 8
0*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B
*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 -
 (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^
4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*
C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/
2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2
*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*
d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20
*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a
^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c
^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f
^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*a*d^21*
f^4 - 32*A*a*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^5*f^2 -
 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^
2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^
5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 +
40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C
*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*
C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*
c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^
4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a
^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*
a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*
c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 2
0*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*
a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^
2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4
+ d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^
4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7
*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5
+ 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^
5) - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3
136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 243
2*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c
^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*
d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6
*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 +
128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*
C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a
*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) - (c + d*tan(e + f

x))^(1/2)(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*(-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(((-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c

^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*A*a*d^21*f^4 + 32*C*a*d^21*f^4 - 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c^3*d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*d^12*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6*f^4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) + (c + d*tan(e + f*x))^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 - 320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^10*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3 - 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^12*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 1152*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 640*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7

*f^3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*(-(((8*A^2*a^2
*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 +
 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*
A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*
d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^
2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^
2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*
f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^
4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2
*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^
2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 4
0*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d
^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2
 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*
A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*
(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*
c^8*d^2*f^4)))^(1/2) + ((-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*
a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*
c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2
+ 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A
*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^
2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (1
6*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f
^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a
^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2)
+ 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^
3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5
*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^
2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*
c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*
d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4
+ 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(32*C*a*d^21*f^4
 - 32*A*a*d^21*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*a^2*c^5*f^2 - 8*
B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c
^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f
^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*
C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^
2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a
^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4
*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 -
 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4
- 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2
*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3
*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A
^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2

```
*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d
^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d
^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))
)^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^
16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7
680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5)
- 160*A*a*c^2*d^19*f^4 - 128*A*a*c^4*d^17*f^4 + 896*A*a*c^6*d^15*f^4 + 3136
*A*a*c^8*d^13*f^4 + 4928*A*a*c^10*d^11*f^4 + 4480*A*a*c^12*d^9*f^4 + 2432*A
*a*c^14*d^7*f^4 + 736*A*a*c^16*d^5*f^4 + 96*A*a*c^18*d^3*f^4 + 736*B*a*c^3*
d^18*f^4 + 2432*B*a*c^5*d^16*f^4 + 4480*B*a*c^7*d^14*f^4 + 4928*B*a*c^9*d^1
2*f^4 + 3136*B*a*c^11*d^10*f^4 + 896*B*a*c^13*d^8*f^4 - 128*B*a*c^15*d^6*f^
4 - 160*B*a*c^17*d^4*f^4 - 32*B*a*c^19*d^2*f^4 + 160*C*a*c^2*d^19*f^4 + 128
*C*a*c^4*d^17*f^4 - 896*C*a*c^6*d^15*f^4 - 3136*C*a*c^8*d^13*f^4 - 4928*C*a
*c^10*d^11*f^4 - 4480*C*a*c^12*d^9*f^4 - 2432*C*a*c^14*d^7*f^4 - 736*C*a*c^
16*d^5*f^4 - 96*C*a*c^18*d^3*f^4 + 96*B*a*c*d^20*f^4) - (c + d*tan(e + f*x)
)^(1/2)*(16*A^2*a^2*d^18*f^3 - 16*B^2*a^2*d^18*f^3 + 16*C^2*a^2*d^18*f^3 -
320*A^2*a^2*c^4*d^14*f^3 - 1024*A^2*a^2*c^6*d^12*f^3 - 1440*A^2*a^2*c^8*d^1
0*f^3 - 1024*A^2*a^2*c^10*d^8*f^3 - 320*A^2*a^2*c^12*d^6*f^3 + 16*A^2*a^2*c
^16*d^2*f^3 + 320*B^2*a^2*c^4*d^14*f^3 + 1024*B^2*a^2*c^6*d^12*f^3 + 1440*B
^2*a^2*c^8*d^10*f^3 + 1024*B^2*a^2*c^10*d^8*f^3 + 320*B^2*a^2*c^12*d^6*f^3
- 16*B^2*a^2*c^16*d^2*f^3 - 320*C^2*a^2*c^4*d^14*f^3 - 1024*C^2*a^2*c^6*d^1
2*f^3 - 1440*C^2*a^2*c^8*d^10*f^3 - 1024*C^2*a^2*c^10*d^8*f^3 - 320*C^2*a^2
*c^12*d^6*f^3 + 16*C^2*a^2*c^16*d^2*f^3 - 32*A*C*a^2*d^18*f^3 - 128*A*B*a^2
*c*d^17*f^3 + 128*B*C*a^2*c*d^17*f^3 - 640*A*B*a^2*c^3*d^15*f^3 - 1152*A*B*
a^2*c^5*d^13*f^3 - 640*A*B*a^2*c^7*d^11*f^3 + 640*A*B*a^2*c^9*d^9*f^3 + 115
2*A*B*a^2*c^11*d^7*f^3 + 640*A*B*a^2*c^13*d^5*f^3 + 128*A*B*a^2*c^15*d^3*f^
3 + 640*A*C*a^2*c^4*d^14*f^3 + 2048*A*C*a^2*c^6*d^12*f^3 + 2880*A*C*a^2*c^8
*d^10*f^3 + 2048*A*C*a^2*c^10*d^8*f^3 + 640*A*C*a^2*c^12*d^6*f^3 - 32*A*C*a
^2*c^16*d^2*f^3 + 640*B*C*a^2*c^3*d^15*f^3 + 1152*B*C*a^2*c^5*d^13*f^3 + 64
0*B*C*a^2*c^7*d^11*f^3 - 640*B*C*a^2*c^9*d^9*f^3 - 1152*B*C*a^2*c^11*d^7*f^
3 - 640*B*C*a^2*c^13*d^5*f^3 - 128*B*C*a^2*c^15*d^3*f^3))*(-(((8*A^2*a^2*c^
5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80
*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C
*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4
*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 -
 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 +
 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4
 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 +
 C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^
2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 +
 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C
^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*
f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 +
40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B
*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^
```

10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 16*B^3*a^3*d^16*f^2 - 192*A^3*a^3*c^3*d^13*f^2 - 480*A^3*a^3*c^5*d^11*f^2 - 640*A^3*a^3*c^7*d^9*f^2 - 480*A^3*a^3*c^9*d^7*f^2 - 192*A^3*a^3*c^11*d^5*f^2 - 32*A^3*a^3*c^13*d^3*f^2 - 80*B^3*a^3*c^2*d^14*f^2 - 144*B^3*a^3*c^4*d^12*f^2 - 80*B^3*a^3*c^6*d^10*f^2 + 80*B^3*a^3*c^8*d^8*f^2 + 144*B^3*a^3*c^10*d^6*f^2 + 80*B^3*a^3*c^12*d^4*f^2 + 16*B^3*a^3*c^14*d^2*f^2 + 192*C^3*a^3*c^3*d^13*f^2 + 480*C^3*a^3*c^5*d^11*f^2 + 640*C^3*a^3*c^7*d^9*f^2 + 480*C^3*a^3*c^9*d^7*f^2 + 192*C^3*a^3*c^11*d^5*f^2 + 32*C^3*a^3*c^13*d^3*f^2 - 16*A^2*B*a^3*d^16*f^2 - 16*B*C^2*a^3*d^16*f^2 - 32*A^3*a^3*c*d^15*f^2 + 32*C^3*a^3*c*d^15*f^2 - 192*A*B^2*a^3*c^3*d^13*f^2 - 480*A*B^2*a^3*c^5*d^11*f^2 - 640*A*B^2*a^3*c^7*d^9*f^2 - 480*A*B^2*a^3*c^9*d^7*f^2 - 192*A*B^2*a^3*c^11*d^5*f^2 - 32*A*B^2*a^3*c^13*d^3*f^2 - 80*A^2*B*a^3*c^2*d^14*f^2 - 144*A^2*B*a^3*c^4*d^12*f^2 - 80*A^2*B*a^3*c^6*d^10*f^2 + 80*A^2*B*a^3*c^8*d^8*f^2 + 144*A^2*B*a^3*c^10*d^6*f^2 + 80*A^2*B*a^3*c^12*d^4*f^2 + 16*A^2*B*a^3*c^14*d^2*f^2 - 576*A*C^2*a^3*c^3*d^13*f^2 - 1440*A*C^2*a^3*c^5*d^11*f^2 - 1920*A*C^2*a^3*c^7*d^9*f^2 - 1440*A*C^2*a^3*c^9*d^7*f^2 - 576*A*C^2*a^3*c^11*d^5*f^2 - 96*A*C^2*a^3*c^13*d^3*f^2 + 576*A^2*C*a^3*c^3*d^13*f^2 + 1440*A^2*C*a^3*c^5*d^11*f^2 + 1920*A^2*C*a^3*c^7*d^9*f^2 + 1440*A^2*C*a^3*c^9*d^7*f^2 + 576*A^2*C*a^3*c^11*d^5*f^2 + 96*A^2*C*a^3*c^13*d^3*f^2 - 80*B*C^2*a^3*c^2*d^14*f^2 - 144*B*C^2*a^3*c^4*d^12*f^2 - 80*B*C^2*a^3*c^6*d^10*f^2 + 80*B*C^2*a^3*c^8*d^8*f^2 + 144*B*C^2*a^3*c^10*d^6*f^2 + 80*B*C^2*a^3*c^12*d^4*f^2 + 16*B*C^2*a^3*c^14*d^2*f^2 + 192*B^2*C*a^3*c^3*d^13*f^2 + 480*B^2*C*a^3*c^5*d^11*f^2 + 640*B^2*C*a^3*c^7*d^9*f^2 + 480*B^2*C*a^3*c^9*d^7*f^2 + 192*B^2*C*a^3*c^11*d^5*f^2 + 32*B^2*C*a^3*c^13*d^3*f^2 + 32*A*B*C*a^3*d^16*f^2 - 32*A*B^2*a^3*c*d^15*f^2 - 96*A*C^2*a^3*c*d^15*f^2 + 96*A^2*C*a^3*c*d^15*f^2 + 32*B^2*C*a^3*c*d^15*f^2 + 160*A*B*C*a^3*c^2*d^14*f^2 + 288*A*B*C*a^3*c^4*d^12*f^2 + 160*A*B*C*a^3*c^6*d^10*f^2 - 160*A*B*C*a^3*c^8*d^8*f^2 - 288*A*B*C*a^3*c^10*d^6*f^2 - 160*A*B*C*a^3*c^12*d^4*f^2 - 32*A*B*C*a^3*c^14*d^2*f^2))*(-(((8*A^2*a^2*c^5*f^2 - 8*B^2*a^2*c^5*f^2 + 8*C^2*a^2*c^5*f^2 - 80*A^2*a^2*c^3*d^2*f^2 + 80*B^2*a^2*c^3*d^2*f^2 - 80*C^2*a^2*c^3*d^2*f^2 + 16*A*B*a^2*d^5*f^2 - 16*A*C*a^2*c^5*f^2 - 16*B*C*a^2*d^5*f^2 + 40*A^2*a^2*c*d^4*f^2 - 40*B^2*a^2*c*d^4*f^2 + 40*C^2*a^2*c*d^4*f^2 + 80*A*B*a^2*c^4*d*f^2 - 80*A*C*a^2*c*d^4*f^2 - 80*B*C*a^2*c^4*d*f^2 - 160*A*B*a^2*c^2*d^3*f^2 + 160*A*C*a^2*c^3*d^2*f^2 + 160*B*C*a^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*a^4 + B^4*a^4 + C^4*a^4 - 4*A*C^3*a^4 - 4*A^3*C*a^4 + 2*A^2*B^2*a^4 + 6*A^2*C^2*a^4 + 2*B^2*C^2*a^4 - 4*A*B^2*C*a^4))^(1/2) + 4*A^2*a^2*c^5*f^2 - 4*B^2*a^2*c^5*f^2 + 4*C^2*a^2*c^5*f^2 - 40*A^2*a^2*c^3*d^2*f^2 + 40*B^2*a^2*c^3*d^2*f^2 - 40*C^2*a^2*c^3*d^2*f^2 + 8*A*B*a^2*d^5*f^2 - 8*A*C*a^2*c^5*f^2 - 8*B*C*a^2*d^5*f^2 + 20*A^2*a^2*c*d^4*f^2 - 20*B^2*a^2*c*d^4*f^2 + 20*C^2*a^2*c*d^4*f^2 + 40*A*B*a^2*c^4*d*f^2 - 40*A*C*a^2*c*d^4*f^2 - 40*B*C*a^2*c^4*d*f^2 - 80*A*B*a^2*c^2*d^3*f^2 + 80*A*C*a^2*c^3*d^2*f^2 + 80*B*C*a^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i - ((2*(A*a*d^2 + C*a*c^2 - B*a*c*d))/(3*(c^2 + d^2)) - (2*d*(c + d*tan(e + f*x))*(B*a

```
*c^2 - B*a*d^2 - 2*A*a*c*d + 2*C*a*c*d))/(c^2 + d^2)^2)/(d*f*(c + d*tan(e +
 f*x))^(3/2)) - atan(-(((c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^18*f^3 - 1
6*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^2*b^2*c^4*d^14*f^3 - 1024*
A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3 - 1024*A^2*b^2*c^10*d^8*f^
3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^2*f^3 + 320*B^2*b^2*c^4*d^
14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2*c^8*d^10*f^3 + 1024*B^2*b
^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 - 16*B^2*b^2*c^16*d^2*f^3 - 320*
C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3 - 1440*C^2*b^2*c^8*d^10*f^
3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*d^6*f^3 + 16*C^2*b^2*c^16*
d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^17*f^3 + 128*B*C*b^2*c*d^17
*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^5*d^13*f^3 - 640*A*B*b^2*c
^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*b^2*c^11*d^7*f^3 + 640*A*B
*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 640*A*C*b^2*c^4*d^14*f^3 + 2
048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*f^3 + 2048*A*C*b^2*c^10*d^
8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^16*d^2*f^3 + 640*B*C*b^2*c^
3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*b^2*c^7*d^11*f^3 - 640*B*C
*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 640*B*C*b^2*c^13*d^5*f^3 - 1
28*B*C*b^2*c^15*d^3*f^3) + (-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C
^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b
^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f
^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 8
0*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B
*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 -
 (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^
4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*
C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/
2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2
*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*
d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20
*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b
^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c
^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f
^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(128*A*b*c^15
*d^6*f^4 - 32*B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 -
 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3136*A*b*c^11*d^10*f^4 - 8
96*A*b*c^13*d^8*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^5*f^2 - 8
*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*
c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*
f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40
*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b
^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*
b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^
4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4
- 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4
 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^
```

2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 - 128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^10*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4 - 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*(-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i + ((c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3 - 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 - 16*B^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3 - 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^17*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 640*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^16*d^2*f^3 + 640*B*C*b^2*

c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*b^2*c^7*d^11*f^3 - 640*B
*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 640*B*C*b^2*c^13*d^5*f^3 -
128*B*C*b^2*c^15*d^3*f^3) - (-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8
*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2
*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5
*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 +
80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A
*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4
- (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*
d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^
3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(
1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b
^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^
2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 +
20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C
*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2
*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8
*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*tan
(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5
*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2
*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^
2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*
c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d
^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*
f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80
*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*
A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2
*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f
^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 +
8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c
*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f
^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2
- 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^
4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^
3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 1
6128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^
6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*B*b*d^21*f^4 - 736*A*b*c^3
*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^
12*f^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^8*f^4 + 128*A*b*c^15*d^6*f
^4 + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 - 12
8*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*B*
b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b*c
^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d^1
6*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^10*
f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4 -

32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*(-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*1i)/(((c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^18*f^3 - 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^2*b^2*c^4*d^14*f^3 - 1024*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3 - 1024*A^2*b^2*c^10*d^8*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^2*f^3 + 320*B^2*b^2*c^4*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2*c^8*d^10*f^3 + 1024*B^2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 - 16*B^2*b^2*c^16*d^2*f^3 - 320*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3 - 1440*C^2*b^2*c^8*d^10*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*d^6*f^3 + 16*C^2*b^2*c^16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^17*f^3 + 128*B*C*b^2*c*d^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^5*d^13*f^3 - 640*A*B*b^2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*b^2*c^11*d^7*f^3 + 640*A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 640*A*C*b^2*c^4*d^14*f^3 + 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*f^3 + 2048*A*C*b^2*c^10*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^16*d^2*f^3 + 640*B*C*b^2*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*b^2*c^7*d^11*f^3 - 640*B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 640*B*C*b^2*c^13*d^5*f^3 - 128*B*C*b^2*c^15*d^3*f^3) - (-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A

```
*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b
^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d
^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*((c + d*t
an(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c
^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d
^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*
A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^
2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2
*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^1
0*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 +
80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 +
2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A
^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2
*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2
+ 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2
*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4
*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f
^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*
c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*
c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 +
 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*
d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*B*b*d^21*f^4 - 736*A*b*c
^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*
d^12*f^4 - 3136*A*b*c^11*d^10*f^4 - 896*A*b*c^13*d^8*f^4 + 128*A*b*c^15*d^6
*f^4 + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 -
128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*
B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b
*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d
^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^1
0*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4
- 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*(-(((8*A^2*
b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^
2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 -
16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2
*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4
*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2
*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d
^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4
*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4
+ 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5
*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2
+ 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^
2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*
f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 +
80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(
```

16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 +
5*c^8*d^2*f^4)))^(1/2) - ((c + d*tan(e + f*x))^(1/2)*(16*A^2*b^2*d^18*f^3
- 16*B^2*b^2*d^18*f^3 + 16*C^2*b^2*d^18*f^3 - 320*A^2*b^2*c^4*d^14*f^3 - 10
24*A^2*b^2*c^6*d^12*f^3 - 1440*A^2*b^2*c^8*d^10*f^3 - 1024*A^2*b^2*c^10*d^8
*f^3 - 320*A^2*b^2*c^12*d^6*f^3 + 16*A^2*b^2*c^16*d^2*f^3 + 320*B^2*b^2*c^4
*d^14*f^3 + 1024*B^2*b^2*c^6*d^12*f^3 + 1440*B^2*b^2*c^8*d^10*f^3 + 1024*B^
2*b^2*c^10*d^8*f^3 + 320*B^2*b^2*c^12*d^6*f^3 - 16*B^2*b^2*c^16*d^2*f^3 - 3
20*C^2*b^2*c^4*d^14*f^3 - 1024*C^2*b^2*c^6*d^12*f^3 - 1440*C^2*b^2*c^8*d^10
*f^3 - 1024*C^2*b^2*c^10*d^8*f^3 - 320*C^2*b^2*c^12*d^6*f^3 + 16*C^2*b^2*c^
16*d^2*f^3 - 32*A*C*b^2*d^18*f^3 - 128*A*B*b^2*c*d^17*f^3 + 128*B*C*b^2*c*d
^17*f^3 - 640*A*B*b^2*c^3*d^15*f^3 - 1152*A*B*b^2*c^5*d^13*f^3 - 640*A*B*b^
2*c^7*d^11*f^3 + 640*A*B*b^2*c^9*d^9*f^3 + 1152*A*B*b^2*c^11*d^7*f^3 + 640*
A*B*b^2*c^13*d^5*f^3 + 128*A*B*b^2*c^15*d^3*f^3 + 640*A*C*b^2*c^4*d^14*f^3
+ 2048*A*C*b^2*c^6*d^12*f^3 + 2880*A*C*b^2*c^8*d^10*f^3 + 2048*A*C*b^2*c^10
*d^8*f^3 + 640*A*C*b^2*c^12*d^6*f^3 - 32*A*C*b^2*c^16*d^2*f^3 + 640*B*C*b^2
*c^3*d^15*f^3 + 1152*B*C*b^2*c^5*d^13*f^3 + 640*B*C*b^2*c^7*d^11*f^3 - 640*
B*C*b^2*c^9*d^9*f^3 - 1152*B*C*b^2*c^11*d^7*f^3 - 640*B*C*b^2*c^13*d^5*f^3
- 128*B*C*b^2*c^15*d^3*f^3) + (-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 +
8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^
2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^
5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2
+ 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*
A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/
4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6
*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A
^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^
(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*
b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b
^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 +
20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*
C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^
2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^
8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(128*A*b*c
^15*d^6*f^4 - 32*B*b*d^21*f^4 - 736*A*b*c^3*d^18*f^4 - 2432*A*b*c^5*d^16*f^
4 - 4480*A*b*c^7*d^14*f^4 - 4928*A*b*c^9*d^12*f^4 - 3136*A*b*c^11*d^10*f^4
- 896*A*b*c^13*d^8*f^4 - (c + d*tan(e + f*x))^(1/2)*(-(((8*A^2*b^2*c^5*f^2
- 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b
^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c
^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 +
40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*
C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B
*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160
*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b
^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*
b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2

*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) + 160*A*b*c^17*d^4*f^4 + 32*A*b*c^19*d^2*f^4 - 160*B*b*c^2*d^19*f^4 - 128*B*b*c^4*d^17*f^4 + 896*B*b*c^6*d^15*f^4 + 3136*B*b*c^8*d^13*f^4 + 4928*B*b*c^10*d^11*f^4 + 4480*B*b*c^12*d^9*f^4 + 2432*B*b*c^14*d^7*f^4 + 736*B*b*c^16*d^5*f^4 + 96*B*b*c^18*d^3*f^4 + 736*C*b*c^3*d^18*f^4 + 2432*C*b*c^5*d^16*f^4 + 4480*C*b*c^7*d^14*f^4 + 4928*C*b*c^9*d^12*f^4 + 3136*C*b*c^11*d^10*f^4 + 896*C*b*c^13*d^8*f^4 - 128*C*b*c^15*d^6*f^4 - 160*C*b*c^17*d^4*f^4 - 32*C*b*c^19*d^2*f^4 - 96*A*b*c*d^20*f^4 + 96*C*b*c*d^20*f^4))*(-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 + 4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2) - 16*A^3*b^3*d^16*f^2 + 16*C^3*b^3*d^16*f^2 - 80*A^3*b^3*c^2*d^14*f^2 - 144*A^3*b^3*c^4*d^12*f^2 - 80*A^3*b^3*c^6*d^10*f^2 + 80*A^3*b^3*c^8*d^8*f^2 + 144*A^3*b^3*c^10*d^6*f^2 + 80*A^3*b^3*c^12*d^4*f^2 + 16*A^3*b^3*c^14*d^2*f^2 + 192*B^3*b^3*c^3*d^13*f^2 + 480*B^3*b^3*c^5*d^11*f^2 + 640*B^3*b^3*c^7*d^9*f^2 + 480*B^3*b^3*c^9*d^7*f^2 + 192*B^3*b^3*c^11*d^5*f^2 + 32*B^3*b^3*c^13*d^3*f^2 + 80*C^3*b^3*c^2*d^14*f^2 + 144*C^3*b^3*c^4*d^12*f^2 + 80*C^3*b^3*c^6*d^10*f^2 - 80*C^3*b^3*c^8*d^8*f^2 - 144*C^3*b^3*c^10*d^6*f^2 - 80*C^3*b^3*c^12*d^4*f^2 - 16*C^3*b^3*c^14*d^2*f^2 - 16*A*B^2*b^3*d^16*f^2 - 48*A*C^2*b^3*d^16*f^2 + 48*A^2*C*b^3*d^16*f^2 + 16*B^2*C*b^3*d^16*f^2 + 32*B^3*b^3*c*d^15*f^2 - 80*A*B^2*b^3*c^2*d^14*f^2 - 144*A*B^2*b^3*c^4*d^12*f^2 - 80*A*B^2*b^3*c^6*d^10*f^2 + 80*A*B^2*b^3*c^8*d^8*f^2 + 144*A*B^2*b^3*c^10*d^6*f^2 + 80*A*B^2*b^3*c^12*d^4*f^2 + 16*A*B^2*b^3*c^14*d^2*f^2 + 192*A^2*B*b^3*c^3*d^13*f^2 + 480*A^2*B*b^3*c^5*d^11*f^2 + 640*A^2*B*b^3*c^7*d^9*f^2 + 480*A^2*B*b^3*c^9*d^7*f^2 + 192*A^2*B*b^3*c^11*d^5*f^

```
2 + 32*A^2*B*b^3*c^13*d^3*f^2 - 240*A*C^2*b^3*c^2*d^14*f^2 - 432*A*C^2*b^3*
c^4*d^12*f^2 - 240*A*C^2*b^3*c^6*d^10*f^2 + 240*A*C^2*b^3*c^8*d^8*f^2 + 432
*A*C^2*b^3*c^10*d^6*f^2 + 240*A*C^2*b^3*c^12*d^4*f^2 + 48*A*C^2*b^3*c^14*d^
2*f^2 + 240*A^2*C*b^3*c^2*d^14*f^2 + 432*A^2*C*b^3*c^4*d^12*f^2 + 240*A^2*C
*b^3*c^6*d^10*f^2 - 240*A^2*C*b^3*c^8*d^8*f^2 - 432*A^2*C*b^3*c^10*d^6*f^2
- 240*A^2*C*b^3*c^12*d^4*f^2 - 48*A^2*C*b^3*c^14*d^2*f^2 + 192*B*C^2*b^3*c^
3*d^13*f^2 + 480*B*C^2*b^3*c^5*d^11*f^2 + 640*B*C^2*b^3*c^7*d^9*f^2 + 480*B
*C^2*b^3*c^9*d^7*f^2 + 192*B*C^2*b^3*c^11*d^5*f^2 + 32*B*C^2*b^3*c^13*d^3*f
^2 + 80*B^2*C*b^3*c^2*d^14*f^2 + 144*B^2*C*b^3*c^4*d^12*f^2 + 80*B^2*C*b^3*
c^6*d^10*f^2 - 80*B^2*C*b^3*c^8*d^8*f^2 - 144*B^2*C*b^3*c^10*d^6*f^2 - 80*B
^2*C*b^3*c^12*d^4*f^2 - 16*B^2*C*b^3*c^14*d^2*f^2 + 32*A^2*B*b^3*c*d^15*f^2
 + 32*B*C^2*b^3*c*d^15*f^2 - 384*A*B*C*b^3*c^3*d^13*f^2 - 960*A*B*C*b^3*c^5
*d^11*f^2 - 1280*A*B*C*b^3*c^7*d^9*f^2 - 960*A*B*C*b^3*c^9*d^7*f^2 - 384*A*
B*C*b^3*c^11*d^5*f^2 - 64*A*B*C*b^3*c^13*d^3*f^2 - 64*A*B*C*b^3*c*d^15*f^2)
)*(-(((8*A^2*b^2*c^5*f^2 - 8*B^2*b^2*c^5*f^2 + 8*C^2*b^2*c^5*f^2 - 80*A^2*b
^2*c^3*d^2*f^2 + 80*B^2*b^2*c^3*d^2*f^2 - 80*C^2*b^2*c^3*d^2*f^2 + 16*A*B*b
^2*d^5*f^2 - 16*A*C*b^2*c^5*f^2 - 16*B*C*b^2*d^5*f^2 + 40*A^2*b^2*c*d^4*f^2
 - 40*B^2*b^2*c*d^4*f^2 + 40*C^2*b^2*c*d^4*f^2 + 80*A*B*b^2*c^4*d*f^2 - 80*
A*C*b^2*c*d^4*f^2 - 80*B*C*b^2*c^4*d*f^2 - 160*A*B*b^2*c^2*d^3*f^2 + 160*A*
C*b^2*c^3*d^2*f^2 + 160*B*C*b^2*c^2*d^3*f^2)^2/4 - (16*c^10*f^4 + 16*d^10*f
^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)*(
A^4*b^4 + B^4*b^4 + C^4*b^4 - 4*A*C^3*b^4 - 4*A^3*C*b^4 + 2*A^2*B^2*b^4 + 6
*A^2*C^2*b^4 + 2*B^2*C^2*b^4 - 4*A*B^2*C*b^4))^(1/2) - 4*A^2*b^2*c^5*f^2 +
4*B^2*b^2*c^5*f^2 - 4*C^2*b^2*c^5*f^2 + 40*A^2*b^2*c^3*d^2*f^2 - 40*B^2*b^2
*c^3*d^2*f^2 + 40*C^2*b^2*c^3*d^2*f^2 - 8*A*B*b^2*d^5*f^2 + 8*A*C*b^2*c^5*f
^2 + 8*B*C*b^2*d^5*f^2 - 20*A^2*b^2*c*d^4*f^2 + 20*B^2*b^2*c*d^4*f^2 - 20*C
^2*b^2*c*d^4*f^2 - 40*A*B*b^2*c^4*d*f^2 + 40*A*C*b^2*c*d^4*f^2 + 40*B*C*b^2
*c^4*d*f^2 + 80*A*B*b^2*c^2*d^3*f^2 - 80*A*C*b^2*c^3*d^2*f^2 - 80*B*C*b^2*c
^2*d^3*f^2)/(16*(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*
c^6*d^4*f^4 + 5*c^8*d^2*f^4)))^(1/2)*2i
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{(a + b\tan(e + fx))\left(A + B\tan(e + fx) + C\tan^2(e + fx)\right)}{(c + d\tan(e + fx))^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e)
)**(5/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(c +
 d*tan(e + f*x))**(5/2), x)
```

3.125 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=209

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)}{3df\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}}-\frac{2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)}{f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}}-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{5/2}}-$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(c-I*d)^(5/2)/f-
(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(c+I*d)^(5/2)/f-2
*(2*c*(A-C)*d-B*(c^2-d^2))/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)-2/3*(A*d^2-
B*c*d+C*c^2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] time = 0.49, antiderivative size = 209, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, integrand size = 35, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.171, Rules used = {3628, 3529, 3539, 3537, 63, 208}

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)}{3df\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}}-\frac{2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)}{f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}}-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{f(c-id)^{5/2}}-$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((c - I
*d)^(5/2)*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c +
I*d]])/((c + I*d)^(5/2)*f) - (2*(c^2*C - B*c*d + A*d^2))/(3*d*(c^2 + d^2)*f
*(c + d*Tan[e + f*x])^(3/2)) - (2*(2*c*(A - C)*d - B*(c^2 - d^2)))/((c^2 +
d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3529

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Simp[((b*c - a*d)*(a + b*Tan[e + f*x])^(m + 1))
/(f*(m + 1)*(a^2 + b^2)), x] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])
^(m + 1)*Simp[a*c + b*d - (b*c - a*d)*Tan[e + f*x], x], x], x] /; FreeQ[{a,
 b, c, d, e, f}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && LtQ[m, -1]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3628

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2
- a*b*B + a^2*C)*(a + b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)*(a^2 + b^2)), x
] + Dist[1/(a^2 + b^2), Int[(a + b*Tan[e + f*x])^(m + 1)*Simp[b*B + a*(A -
C) - (A*b - a*B - b*C)*Tan[e + f*x], x], x], x] /; FreeQ[{a, b, e, f, A, B,
 C}, x] && NeQ[A*b^2 - a*b*B + a^2*C, 0] && LtQ[m, -1] && NeQ[a^2 + b^2, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{2\left(c^2C-Bcd+Ad^2\right)}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{\int \frac{Ac-cC+Bd+(Bc-(A-C)d)\tan(e+}{(c+d\tan(e+fx))^{3/2}}}{c^2+d^2}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2\left(2c(A-C)d-B\left(c^2-d^2\right.\right.}{\left(c^2+d^2\right)^2 f\sqrt{c+d\tan(e+}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2\left(2c(A-C)d-B\left(c^2-d^2\right.\right.}{\left(c^2+d^2\right)^2 f\sqrt{c+d\tan(e+}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2\left(2c(A-C)d-B\left(c^2-d^2\right.\right.}{\left(c^2+d^2\right)^2 f\sqrt{c+d\tan(e+}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} - \frac{2\left(2c(A-C)d-B\left(c^2-d^2\right.\right.}{\left(c^2+d^2\right)^2 f\sqrt{c+d\tan(e+}}$$

$$= -\frac{(B+i(A-C))\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(c-id)^{5/2}f} - \frac{(B-i(A-C))\tanh^{-1}\left(\frac{\sqrt{}}{}\right.}{(c+id)^{5/2}f}$$

Mathematica [C] time = 0.94, size = 223, normalized size = 1.07

$$-\frac{(d(C-A)+Bc)\left(i(c+id)\,{}_2F_1\left(-\frac{3}{2},1;-\frac{1}{2};\frac{c+d\tan(e+fx)}{c-id}\right)-(d+ic)\,{}_2F_1\left(-\frac{3}{2},1;-\frac{1}{2};\frac{c+d\tan(e+fx)}{c+id}\right)\right)-3B(c+d\tan(e+}{3df\left(c^2+d^2\right)(c+d\tan(e}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(c + d*Tan[e + f*x])^(5/2
),x]
```

```
[Out] -1/3*(2*C*(c^2 + d^2) + (B*c + (-A + C)*d)*(I*(c + I*d)*Hypergeometric2F1[-
3/2, 1, -1/2, (c + d*Tan[e + f*x])/(c - I*d)] - (I*c + d)*Hypergeometric2F1
[-3/2, 1, -1/2, (c + d*Tan[e + f*x])/(c + I*d)]) - 3*B*(I*(c + I*d)*Hyperge
ometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c - I*d)] - (I*c + d)*Hyperg
eometric2F1[-1/2, 1, 1/2, (c + d*Tan[e + f*x])/(c + I*d)])*(c + d*Tan[e + f
*x]))/(d*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2))
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/2),x, algorit
hm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/2),x, algorit
hm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.40, size = 20647, normalized size = 98.79

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/2),x)
```

```
[Out] result too large to display
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(5/2),x, algorit
hm="maxima")
```

```
[Out] Timed out
```

mupad [B] time = 37.59, size = 14163, normalized size = 67.77

result too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/(c + d*tan(e + f*x))^(5/2),x)
```

```
[Out] (log(96*A^3*c^3*d^13*f^2 - ((((((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 17
60*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*
```

A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(((((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5))/4 - 32*A*d^21*f^4 - 160*A*c^2*d^19*f^4 - 128*A*c^4*d^17*f^4 + 896*A*c^6*d^15*f^4 + 3136*A*c^8*d^13*f^4 + 4928*A*c^10*d^11*f^4 + 4480*A*c^12*d^9*f^4 + 2432*A*c^14*d^7*f^4 + 736*A*c^16*d^5*f^4 + 96*A*c^18*d^3*f^4))/4 - (c + d*tan(e + f*x))^(1/2)*(320*A^2*c^4*d^14*f^3 - 16*A^2*d^18*f^3 + 1024*A^2*c^6*d^12*f^3 + 1440*A^2*c^8*d^10*f^3 + 1024*A^2*c^10*d^8*f^3 + 320*A^2*c^12*d^6*f^3 - 16*A^2*c^16*d^2*f^3))*(((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 + 240*A^3*c^5*d^11*f^2 + 320*A^3*c^7*d^9*f^2 + 240*A^3*c^9*d^7*f^2 + 96*A^3*c^11*d^5*f^2 + 16*A^3*c^13*d^3*f^2 + 16*A^3*c*d^15*f^2)*(((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 + (log(96*A^3*c^3*d^13*f^2 - ((((-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) + 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(((-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) + 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5))/4 - 32*A*d^21*f^4 - 160*A*c^2*d^19*f^4 - 128*A*c^4*d^17*f^4 + 896*A*c^6*d^15*f^4 + 3136*A*c^8*d^13*f^4 + 4928*A*c^10*d^11*f^4 + 4480*A*c^12*d^9*f^4 + 2432*A*c^14*d^7*f^4 + 736*A*c^16*d^5*f^4 + 96*A*c^18*d^3*f^4))/4 - (c + d*tan(e + f*x))^(1/2)*(320*A^2*c^4*d^14*f^3 - 16*A^2*d^18*f^3 + 1024*A^2*c^6*d^12*f^3 + 1440*A^2*c^8*d^10*f^3 + 1024*A^2*c^10*d^8*f^3 + 320*A^2*c^12*d^6*f^3 - 16*A^2*c^16*d^2*f^3))*(-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) + 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 + 240*A^3*c^5*d^11*f^2 + 320*A^3*c^7*d^9*f^2 + 240*A^3*c^9*d^7*f^2 + 96*A^3*c^11*d^5*f^2 + 16*A^3*c^13*d^3*f^2 + 16*A^3*c*d^15*f^2)*(-((320*A^4

```
*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^
4 - 400*A^4*c^8*d^2*f^4)^(1/2) + 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^
2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6
*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 - log(96*A^3*c^3*d^13*f^2 - ((((320*A^4
*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^
4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^
2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4
+ 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*(896*A*c^6*d^15*f^4 - (((320*A^4
*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^
4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^
2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4
+ 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c
*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 1344
0*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*
f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*c^2*d
^19*f^4 - 128*A*c^4*d^17*f^4 - 32*A*d^21*f^4 + 3136*A*c^8*d^13*f^4 + 4928*A
*c^10*d^11*f^4 + 4480*A*c^12*d^9*f^4 + 2432*A*c^14*d^7*f^4 + 736*A*c^16*d^5
*f^4 + 96*A*c^18*d^3*f^4) + (c + d*tan(e + f*x))^(1/2)*(320*A^2*c^4*d^14*f^
3 - 16*A^2*d^18*f^3 + 1024*A^2*c^6*d^12*f^3 + 1440*A^2*c^8*d^10*f^3 + 1024*
A^2*c^10*d^8*f^3 + 320*A^2*c^12*d^6*f^3 - 16*A^2*c^16*d^2*f^3))*(((320*A^4*
c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4
 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^2
*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 +
 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 240*A^3*c^5*d^11*f^2 + 320*A^3*
c^7*d^9*f^2 + 240*A^3*c^9*d^7*f^2 + 96*A^3*c^11*d^5*f^2 + 16*A^3*c^13*d^3*f
^2 + 16*A^3*c*d^15*f^2)*(((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4
*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) - 4*A^2*c^
5*f^2 + 40*A^2*c^3*d^2*f^2 - 20*A^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 +
 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2
) - log(96*A^3*c^3*d^13*f^2 - ((-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 -
1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) +
4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^
10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^
4))^(1/2)*(896*A*c^6*d^15*f^4 - (-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 -
 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) +
 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d
^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f
^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2
880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12
*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^
19*d^4*f^5 + 64*c^21*d^2*f^5) - 160*A*c^2*d^19*f^4 - 128*A*c^4*d^17*f^4 - 3
2*A*d^21*f^4 + 3136*A*c^8*d^13*f^4 + 4928*A*c^10*d^11*f^4 + 4480*A*c^12*d^9
*f^4 + 2432*A*c^14*d^7*f^4 + 736*A*c^16*d^5*f^4 + 96*A*c^18*d^3*f^4) + (c +
 d*tan(e + f*x))^(1/2)*(320*A^2*c^4*d^14*f^3 - 16*A^2*d^18*f^3 + 1024*A^2*c
^6*d^12*f^3 + 1440*A^2*c^8*d^10*f^3 + 1024*A^2*c^10*d^8*f^3 + 320*A^2*c^12*
```

d^6*f^3 - 16*A^2*c^16*d^2*f^3))*(-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) + 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + 240*A^3*c^5*d^11*f^2 + 320*A^3*c^7*d^9*f^2 + 240*A^3*c^9*d^7*f^2 + 96*A^3*c^11*d^5*f^2 + 16*A^3*c^13*d^3*f^2 + 16*A^3*c*d^15*f^2)*(-((320*A^4*c^2*d^8*f^4 - 16*A^4*d^10*f^4 - 1760*A^4*c^4*d^6*f^4 + 1600*A^4*c^6*d^4*f^4 - 400*A^4*c^8*d^2*f^4)^(1/2) + 4*A^2*c^5*f^2 - 40*A^2*c^3*d^2*f^2 + 20*A^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + (log(((((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*C^2*c^4*d^14*f^3 - 16*C^2*d^18*f^3 + 1024*C^2*c^6*d^12*f^3 + 1440*C^2*c^8*d^10*f^3 + 1024*C^2*c^10*d^8*f^3 + 320*C^2*c^12*d^6*f^3 - 16*C^2*c^16*d^2*f^3) + ((((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(896*C*c^6*d^15*f^4 - ((((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5))/4 - 160*C*c^2*d^19*f^4 - 128*C*c^4*d^17*f^4 - 32*C*d^21*f^4 + 3136*C*c^8*d^13*f^4 + 4928*C*c^10*d^11*f^4 + 4480*C*c^12*d^9*f^4 + 2432*C*c^14*d^7*f^4 + 736*C*c^16*d^5*f^4 + 96*C*c^18*d^3*f^4))/4))/4 - 96*C^3*c^3*d^13*f^2 - 240*C^3*c^5*d^11*f^2 - 320*C^3*c^7*d^9*f^2 - 240*C^3*c^9*d^7*f^2 - 96*C^3*c^11*d^5*f^2 - 16*C^3*c^13*d^3*f^2 - 16*C^3*c*d^15*f^2)*(((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 + (log(((-((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*C^2*c^4*d^14*f^3 - 16*C^2*d^18*f^3 + 1024*C^2*c^6*d^12*f^3 + 1440*C^2*c^8*d^10*f^3 + 1024*C^2*c^10*d^8*f^3 + 320*C^2*c^12*d^6*f^3 - 16*C^2*c^16*d^2*f^3) + ((-((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(896*C*c^6*d^15*f^4 - ((-(

(320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5))/4 - 160*C*c^2*d^19*f^4 - 128*C*c^4*d^17*f^4 - 32*C*d^21*f^4 + 3136*C*c^8*d^13*f^4 + 4928*C*c^10*d^11*f^4 + 4480*C*c^12*d^9*f^4 + 2432*C*c^14*d^7*f^4 + 736*C*c^16*d^5*f^4 + 96*C*c^18*d^3*f^4))/4))/4 - 96*C^3*c^3*d^13*f^2 - 240*C^3*c^5*d^11*f^2 - 320*C^3*c^7*d^9*f^2 - 240*C^3*c^9*d^7*f^2 - 96*C^3*c^11*d^5*f^2 - 16*C^3*c^13*d^3*f^2 - 16*C^3*c*d^15*f^2)*(-((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 - log(- (((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*C^2*c^4*d^14*f^3 - 16*C^2*d^18*f^3 + 1024*C^2*c^6*d^12*f^3 + 1440*C^2*c^8*d^10*f^3 + 1024*C^2*c^10*d^8*f^3 + 320*C^2*c^12*d^6*f^3 - 16*C^2*c^16*d^2*f^3) - (((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*C*d^21*f^4 - 160*C*c^2*d^19*f^4 - 128*C*c^4*d^17*f^4 + 896*C*c^6*d^15*f^4 + 3136*C*c^8*d^13*f^4 + 4928*C*c^10*d^11*f^4 + 4480*C*c^12*d^9*f^4 + 2432*C*c^14*d^7*f^4 + 736*C*c^16*d^5*f^4 + 96*C*c^18*d^3*f^4)) - 96*C^3*c^3*d^13*f^2 - 240*C^3*c^5*d^11*f^2 - 320*C^3*c^7*d^9*f^2 - 240*C^3*c^9*d^7*f^2 - 96*C^3*c^11*d^5*f^2 - 16*C^3*c^13*d^3*f^2 - 16*C^3*c*d^15*f^2)*(((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) - 4*C^2*c^5*f^2 + 40*C^2*c^3*d^2*f^2 - 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) - log(- (-((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*C^

```
2*c^4*d^14*f^3 - 16*C^2*d^18*f^3 + 1024*C^2*c^6*d^12*f^3 + 1440*C^2*c^8*d^1
0*f^3 + 1024*C^2*c^10*d^8*f^3 + 320*C^2*c^12*d^6*f^3 - 16*C^2*c^16*d^2*f^3)
 - (-((320*C^4*c^2*d^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*
C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d
^2*f^2 + 20*C^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 16
0*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((-((320*C^4*c^2*d
^8*f^4 - 16*C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 40
0*C^4*c^8*d^2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c*d^
4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*
c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*
f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*
d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 +
2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) - 32*C*d^21*f^4 - 1
60*C*c^2*d^19*f^4 - 128*C*c^4*d^17*f^4 + 896*C*c^6*d^15*f^4 + 3136*C*c^8*d^
13*f^4 + 4928*C*c^10*d^11*f^4 + 4480*C*c^12*d^9*f^4 + 2432*C*c^14*d^7*f^4 +
 736*C*c^16*d^5*f^4 + 96*C*c^18*d^3*f^4)) - 96*C^3*c^3*d^13*f^2 - 240*C^3*c
^5*d^11*f^2 - 320*C^3*c^7*d^9*f^2 - 240*C^3*c^9*d^7*f^2 - 96*C^3*c^11*d^5*f
^2 - 16*C^3*c^13*d^3*f^2 - 16*C^3*c*d^15*f^2)*(-((320*C^4*c^2*d^8*f^4 - 16*
C^4*d^10*f^4 - 1760*C^4*c^4*d^6*f^4 + 1600*C^4*c^6*d^4*f^4 - 400*C^4*c^8*d^
2*f^4)^(1/2) + 4*C^2*c^5*f^2 - 40*C^2*c^3*d^2*f^2 + 20*C^2*c*d^4*f^2)/(16*c
^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4
+ 80*c^8*d^2*f^4))^(1/2) + (log(8*B^3*d^16*f^2 - ((((320*B^4*c^2*d^8*f^4 -
16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8
*d^2*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(c
^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^
8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*B^2*c^4*d^14*f^3 - 16*B^
2*d^18*f^3 + 1024*B^2*c^6*d^12*f^3 + 1440*B^2*c^8*d^10*f^3 + 1024*B^2*c^10*
d^8*f^3 + 320*B^2*c^12*d^6*f^3 - 16*B^2*c^16*d^2*f^3) + ((((320*B^4*c^2*d^8
*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*
B^4*c^8*d^2*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*
f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4
 + 5*c^8*d^2*f^4))^(1/2)*(((((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*
B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) + 4*B^2
*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*
c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c +
d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5
 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^1
3*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*
c^21*d^2*f^5))/4 + 96*B*c*d^20*f^4 + 736*B*c^3*d^18*f^4 + 2432*B*c^5*d^16*f
^4 + 4480*B*c^7*d^14*f^4 + 4928*B*c^9*d^12*f^4 + 3136*B*c^11*d^10*f^4 + 896
*B*c^13*d^8*f^4 - 128*B*c^15*d^6*f^4 - 160*B*c^17*d^4*f^4 - 32*B*c^19*d^2*f
^4))/4))/4 + 40*B^3*c^2*d^14*f^2 + 72*B^3*c^4*d^12*f^2 + 40*B^3*c^6*d^10*f^
2 - 40*B^3*c^8*d^8*f^2 - 72*B^3*c^10*d^6*f^2 - 40*B^3*c^12*d^4*f^2 - 8*B^3*
c^14*d^2*f^2)*(((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f
^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40
```

*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 + (log(8*B^3*d^16*f^2 - ((-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*B^2*c^4*d^14*f^3 - 16*B^2*d^18*f^3 + 1024*B^2*c^6*d^12*f^3 + 1440*B^2*c^8*d^10*f^3 + 1024*B^2*c^10*d^8*f^3 + 320*B^2*c^12*d^6*f^3 - 16*B^2*c^16*d^2*f^3) + ((-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(((-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5))/4 + 96*B*c*d^20*f^4 + 736*B*c^3*d^18*f^4 + 2432*B*c^5*d^16*f^4 + 4480*B*c^7*d^14*f^4 + 4928*B*c^9*d^12*f^4 + 3136*B*c^11*d^10*f^4 + 896*B*c^13*d^8*f^4 - 128*B*c^15*d^6*f^4 - 160*B*c^17*d^4*f^4 - 32*B*c^19*d^2*f^4))/4))/4 + 40*B^3*c^2*d^14*f^2 + 72*B^3*c^4*d^12*f^2 + 40*B^3*c^6*d^10*f^2 - 40*B^3*c^8*d^8*f^2 - 72*B^3*c^10*d^6*f^2 - 40*B^3*c^12*d^4*f^2 - 8*B^3*c^14*d^2*f^2)*(-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(c^10*f^4 + d^10*f^4 + 5*c^2*d^8*f^4 + 10*c^4*d^6*f^4 + 10*c^6*d^4*f^4 + 5*c^8*d^2*f^4))^(1/2))/4 - log((((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((c + d*tan(e + f*x))^(1/2)*(320*B^2*c^4*d^14*f^3 - 16*B^2*d^18*f^3 + 1024*B^2*c^6*d^12*f^3 + 1440*B^2*c^8*d^10*f^3 + 1024*B^2*c^10*d^8*f^3 + 320*B^2*c^12*d^6*f^3 - 16*B^2*c^16*d^2*f^3) - (((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*(96*B*c*d^20*f^4 - (((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) + 4*B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*d^4*f^5 + 64*c^21*d^2*f^5) + 736*B*c^3*d^18*f^4 + 2432*B*c^5*d

```
^16*f^4 + 4480*B*c^7*d^14*f^4 + 4928*B*c^9*d^12*f^4 + 3136*B*c^11*d^10*f^4
+ 896*B*c^13*d^8*f^4 - 128*B*c^15*d^6*f^4 - 160*B*c^17*d^4*f^4 - 32*B*c^19*
d^2*f^4)) + 8*B^3*d^16*f^2 + 40*B^3*c^2*d^14*f^2 + 72*B^3*c^4*d^12*f^2 + 40
*B^3*c^6*d^10*f^2 - 40*B^3*c^8*d^8*f^2 - 72*B^3*c^10*d^6*f^2 - 40*B^3*c^12*
d^4*f^2 - 8*B^3*c^14*d^2*f^2)*(((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 17
60*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) + 4*
B^2*c^5*f^2 - 40*B^2*c^3*d^2*f^2 + 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10
*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)
)^(1/2) - log((-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d^6*
f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2 + 4
0*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^2*d
^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2)*((c + d
*tan(e + f*x))^(1/2)*(320*B^2*c^4*d^14*f^3 - 16*B^2*d^18*f^3 + 1024*B^2*c^6
*d^12*f^3 + 1440*B^2*c^8*d^10*f^3 + 1024*B^2*c^10*d^8*f^3 + 320*B^2*c^12*d^
6*f^3 - 16*B^2*c^16*d^2*f^3) - (-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 -
1760*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) -
4*B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^
10*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^
4))^(1/2)*(96*B*c*d^20*f^4 - (-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 17
60*B^4*c^4*d^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*
B^2*c^5*f^2 + 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10
*f^4 + 80*c^2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4)
)^(1/2)*(c + d*tan(e + f*x))^(1/2)*(64*c*d^22*f^5 + 640*c^3*d^20*f^5 + 2880
*c^5*d^18*f^5 + 7680*c^7*d^16*f^5 + 13440*c^9*d^14*f^5 + 16128*c^11*d^12*f^
5 + 13440*c^13*d^10*f^5 + 7680*c^15*d^8*f^5 + 2880*c^17*d^6*f^5 + 640*c^19*
d^4*f^5 + 64*c^21*d^2*f^5) + 736*B*c^3*d^18*f^4 + 2432*B*c^5*d^16*f^4 + 448
0*B*c^7*d^14*f^4 + 4928*B*c^9*d^12*f^4 + 3136*B*c^11*d^10*f^4 + 896*B*c^13*
d^8*f^4 - 128*B*c^15*d^6*f^4 - 160*B*c^17*d^4*f^4 - 32*B*c^19*d^2*f^4)) + 8
*B^3*d^16*f^2 + 40*B^3*c^2*d^14*f^2 + 72*B^3*c^4*d^12*f^2 + 40*B^3*c^6*d^10
*f^2 - 40*B^3*c^8*d^8*f^2 - 72*B^3*c^10*d^6*f^2 - 40*B^3*c^12*d^4*f^2 - 8*B
^3*c^14*d^2*f^2)*(-((320*B^4*c^2*d^8*f^4 - 16*B^4*d^10*f^4 - 1760*B^4*c^4*d
^6*f^4 + 1600*B^4*c^6*d^4*f^4 - 400*B^4*c^8*d^2*f^4)^(1/2) - 4*B^2*c^5*f^2
+ 40*B^2*c^3*d^2*f^2 - 20*B^2*c*d^4*f^2)/(16*c^10*f^4 + 16*d^10*f^4 + 80*c^
2*d^8*f^4 + 160*c^4*d^6*f^4 + 160*c^6*d^4*f^4 + 80*c^8*d^2*f^4))^(1/2) + ((
2*B*c)/(3*(c^2 + d^2)) + (2*B*(c^2 - d^2)*(c + d*tan(e + f*x)))/(c^2 + d^2)
^2)/(f*(c + d*tan(e + f*x))^(3/2)) - ((2*A*d)/(3*(c^2 + d^2)) + (4*A*c*d*(c
 + d*tan(e + f*x)))/(c^2 + d^2)^2)/(f*(c + d*tan(e + f*x))^(3/2)) - ((2*C*c
^2)/(3*(c^2 + d^2)) - (4*C*c*d^2*(c + d*tan(e + f*x)))/(c^2 + d^2)^2)/(d*f*
(c + d*tan(e + f*x))^(3/2))
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\left(c + d\tan\left(e + fx\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(5/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(c + d*tan(e + f*x))**(5/
2), x)
```

3.126 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=365

$$-\frac{2b^{3/2}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{f\left(a^2+b^2\right)(bc-ad)^{5/2}}+\frac{2\left(Ad^2-Bcd+c^2C\right)}{3f\left(c^2+d^2\right)(bc-ad)(c+d\tan(e+fx))^{3/2}}+\frac{2\left(b\left(c^2d^2(3A\right.\right.}{}$$

```
[Out] (A-I*B-C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(I*a+b)/(c-I*d)^(5/
2)/f+(I*A-B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(a+I*b)/(c+I
*d)^(5/2)/f-2*b^(3/2)*(A*b^2-a*(B*b-C*a))*arctanh(b^(1/2)*(c+d*tan(f*x+e))^
(1/2)/(-a*d+b*c)^(1/2))/(a^2+b^2)/(-a*d+b*c)^(5/2)/f+2*(b*(c^4*C-2*B*c^3*d+
c^2*(3*A-C)*d^2+A*d^4)-a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))/(-a*d+b*c)^2/(c^2+d
^2)^2/f/(c+d*tan(f*x+e))^(1/2)+2/3*(A*d^2-B*c*d+C*c^2)/(-a*d+b*c)/(c^2+d^2)
/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] time = 2.47, antiderivative size = 365, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3649, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{2b^{3/2}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{f\left(a^2+b^2\right)(bc-ad)^{5/2}}+\frac{2\left(b\left(c^2d^2(3A-C)+Ad^4-2Bc^3d+c^4C\right)-ad^2\left(2cd(A-\right.\right.}{f\left(c^2+d^2\right)^2(bc-ad)^2\sqrt{c+d\tan(e+fx)}}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c + d*Ta
n[e + f*x])^(5/2)),x]
```

```
[Out] ((A - I*B - C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((I*a + b)*
(c - I*d)^(5/2)*f) + ((I*A - B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt
[c + I*d]])/((a + I*b)*(c + I*d)^(5/2)*f) - (2*b^(3/2)*(A*b^2 - a*(b*B - a*
C))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c - a*d]])/((a^2 + b^
2)*(b*c - a*d)^(5/2)*f) + (2*(c^2*C - B*c*d + A*d^2))/(3*(b*c - a*d)*(c^2 +
 d^2)*f*(c + d*Tan[e + f*x])^(3/2)) + (2*(b*(c^4*C - 2*B*c^3*d + c^2*(3*A -
 C)*d^2 + A*d^4) - a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2))))/((b*c - a*d)^2*(
c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
```

```
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))(c+d\tan(e+fx))^{5/2}}\,dx = \frac{2\left(c^2C-Bcd+Ad^2\right)}{3(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\int \frac{-\frac{3}{2}\left(aAcd-ad(cC-\right.}{}}{}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)}{3(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\left(b\left(c^4C-2Bc^3d\right.\right.}{}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)}{3(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\left(b\left(c^4C-2Bc^3d\right.\right.}{}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)}{3(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\left(b\left(c^4C-2Bc^3d\right.\right.}{}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)}{3(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\left(b\left(c^4C-2Bc^3d\right.\right.}{}$$

$$= -\frac{2b^{3/2}\left(Ab^2-a(bB-aC)\right)\tanh^{-1}\left(\frac{\sqrt{b}\sqrt{c+d\tan(e+fx)}}{\sqrt{bc-ad}}\right)}{\left(a^2+b^2\right)(bc-ad)^{5/2}f} + \frac{}{3(bc-a}$$

$$= \frac{(A-iB-C)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(ia+b)(c-id)^{5/2}f} - \frac{(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c}}{}\right.}{(ia-b)(c+id)^5}$$

Mathematica [B] time = 6.28, size = 1948, normalized size = 5.34

result too large to display

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])*(c
+ d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] (-2*(A*d^2 - c*(-(c*C) + B*d)))/(3*(-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[
e + f*x])^(3/2)) - (2*((-2*(((I*Sqrt[c - I*d]*((b*(-(b*c) + a*d)*((-3*c*(b*
c - a*d)*(B*c - (A - C)*d))/2 - (3*b*d*(c^2*C - B*c*d + A*d^2))/2 - (3*d*(a
*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2))/2 + a*((-3*((b*d^2)/2 - (c
*(-(b*c) + a*d))/2)*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2 - (a*d
*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2
))/2 - (b*((-3*d^2*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2 - c*((3
*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2)))/
2) - I*((a*(-(b*c) + a*d)*((-3*c*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*d*
(c^2*C - B*c*d + A*d^2))/2 - (3*d*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d
^2)))/2))/2 - b*((-3*((b*d^2)/2 - (c*(-(b*c) + a*d))/2)*(a*A*c*d - a*d*(c*C
 - B*d) - A*b*(c^2 + d^2)))/2 - (a*d*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2
 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2))/2 - (b*((-3*d^2*(a*A*c*d - a*d*(c*C
- B*d) - A*b*(c^2 + d^2)))/2 - c*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (
3*b*c*(c^2*C - B*c*d + A*d^2))/2)))/2)))*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/S
qrt[c - I*d]])/((-c + I*d)*f) - (I*Sqrt[c + I*d]*((b*(-(b*c) + a*d)*((-3*c*
(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*d*(c^2*C - B*c*d + A*d^2))/2 - (3*d
*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2))/2 + a*((-3*((b*d^2)/2 -
 (c*(-(b*c) + a*d))/2)*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2 - (
a*d*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A*d^2)
)/2))/2 - (b*((-3*d^2*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2 - c*
((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2)
))/2) + I*((a*(-(b*c) + a*d)*((-3*c*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b
*d*(c^2*C - B*c*d + A*d^2))/2 - (3*d*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2
+ d^2)))/2))/2 - b*((-3*((b*d^2)/2 - (c*(-(b*c) + a*d))/2)*(a*A*c*d - a*d*(
c*C - B*d) - A*b*(c^2 + d^2)))/2 - (a*d*((3*d*(b*c - a*d)*(B*c - (A - C)*d)
)/2 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2))/2 - (b*((-3*d^2*(a*A*c*d - a*d*(c
*C - B*d) - A*b*(c^2 + d^2)))/2 - c*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2
- (3*b*c*(c^2*C - B*c*d + A*d^2))/2)))/2)))*ArcTanh[Sqrt[c + d*Tan[e + f*x]
]/Sqrt[c + I*d]])/((-c - I*d)*f))/(a^2 + b^2) + (2*Sqrt[b*c - a*d]*(-1/2*(a
*b*(-(b*c) + a*d)*((-3*c*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*d*(c^2*C -
 B*c*d + A*d^2))/2 - (3*d*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2)
) + (a^2*b*((-3*d^2*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)))/2 - c*((
3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A*d^2))/2)))
/2 + b^2*((-3*((b*d^2)/2 - (c*(-(b*c) + a*d))/2)*(a*A*c*d - a*d*(c*C - B*d)
 - A*b*(c^2 + d^2)))/2 - (a*d*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b
```

```
*c*(c^2*C - B*c*d + A*d^2))/2))/2))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x
]])/Sqrt[b*c - a*d]])/(Sqrt[b]*(a^2 + b^2)*(-(b*c) + a*d)*f)))/((-(b*c) + a
*d)*(c^2 + d^2)) - (2*((-3*d^2*(a*A*c*d - a*d*(c*C - B*d) - A*b*(c^2 + d^2)
))/2 - c*((3*d*(b*c - a*d)*(B*c - (A - C)*d))/2 - (3*b*c*(c^2*C - B*c*d + A
*d^2))/2)))/((-(b*c) + a*d)*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])))/(3*(-
(b*c) + a*d)*(c^2 + d^2))
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.67, size = 45119, normalized size = 123.61

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))^(5/2)
,x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e))
^(5/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))*(c + d*ta
n(e + f*x))^(5/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)\left(c + d\tan\left(e + fx\right)\right)^{\frac{5}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))/(c+d*tan(f*x+e)
)**(5/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))*(c
+ d*tan(e + f*x))**(5/2)), x)
```

3.127 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=679

$$-\frac{d\left(A\left(2a^2d^2+b^2\left(3c^2+5d^2\right)\right)+a^2\left(-2Bcd+5c^2C+3Cd^2\right)-3abB\left(c^2+d^2\right)+2b^2c(cC-Bd)\right)}{3f\left(a^2+b^2\right)\left(c^2+d^2\right)(bc-ad)^2(c+d\tan(e+fx))^{3/2}}-\frac{}{f\left(a^2+b^2\right)(bc}$$

```
[Out] -(I*A+B-I*C)*arctanh((c+d*tan(f*x+e))^(1/2)/(c-I*d)^(1/2))/(a-I*b)^2/(c-I*d
)^(5/2)/f-(B-I*(A-C))*arctanh((c+d*tan(f*x+e))^(1/2)/(c+I*d)^(1/2))/(a+I*b)
^2/(c+I*d)^(5/2)/f-b^(3/2)*(7*a^3*b*B*d-5*a^4*C*d+b^4*(-5*A*d+2*B*c)+a*b^3*
(4*A*c+3*B*d-4*C*c)-a^2*b^2*(2*B*c+(9*A+C)*d))*arctanh(b^(1/2)*(c+d*tan(f*x
+e))^(1/2)/(-a*d+b*c)^(1/2))/(a^2+b^2)^2/(-a*d+b*c)^(7/2)/f-d*(2*a^3*d^2*(B
*c^2-B*d^2+2*C*c*d)+2*b^3*c*(-3*B*c^2*d-B*d^3+2*C*c^3)-a*b^2*(B*c^4+3*B*d^4
-4*C*c*d^3)+a^2*b*(-6*B*c^3*d-2*B*c*d^3+5*C*c^4+2*C*c^2*d^2+C*d^4)-A*(4*a^3
*c*d^3+4*a*b^2*c*d^3-4*a^2*b*d^2*(2*c^2+d^2)-b^3*(c^4+10*c^2*d^2+5*d^4)))/(
a^2+b^2)/(-a*d+b*c)^3/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)-1/3*d*(2*b^2*c*(
-B*d+C*c)-3*a*b*B*(c^2+d^2)+a^2*(-2*B*c*d+5*C*c^2+3*C*d^2)+A*(2*a^2*d^2+b^2
*(3*c^2+5*d^2)))/(a^2+b^2)/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)+
(-A*b^2+a*(B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))/(c+d*tan(f*x+e
))^(3/2)
```

Rubi [A] time = 5.06, antiderivative size = 678, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 7, integrand size = 47, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.149, Rules used = {3649, 3653, 3539, 3537, 63, 208, 3634}

$$-\frac{d\left(-A\left(-4a^2bd^2\left(2c^2+d^2\right)+4a^3cd^3+4ab^2cd^3+b^3\left(-\left(10c^2d^2+c^4+5d^4\right)\right)\right)+a^2b\left(-6Bc^3d-2Bcd^3+2c^2Cd^2\right.}{f\left(a^2+b^2\right)\left(c^2+d^2\right)^2(bc-a}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(c + d*
Tan[e + f*x])^(5/2)),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[Sqrt[c + d*Tan[e + f*x]]/Sqrt[c - I*d]])/((a - I
*b)^2*(c - I*d)^(5/2)*f)) - ((B - I*(A - C))*ArcTanh[Sqrt[c + d*Tan[e + f*x
]]/Sqrt[c + I*d]])/((a + I*b)^2*(c + I*d)^(5/2)*f) - (b^(3/2)*(7*a^3*b*B*d
- 5*a^4*C*d + b^4*(2*B*c - 5*A*d) + a*b^3*(4*A*c - 4*c*C + 3*B*d) - a^2*b^2
*(2*B*c + (9*A + C)*d))*ArcTanh[(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])/Sqrt[b*c
 - a*d]])/((a^2 + b^2)^2*(b*c - a*d)^(7/2)*f) - (d*(2*a^2*A*d^2 + 2*b^2*c*(
c*C - B*d) - 3*a*b*B*(c^2 + d^2) + A*b^2*(3*c^2 + 5*d^2) + a^2*(5*c^2*C - 2
```

```
*B*c*d + 3*C*d^2)))/(3*(a^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f*(c + d*Tan[e
 + f*x])^(3/2)) - (A*b^2 - a*(b*B - a*C))/((a^2 + b^2)*(b*c - a*d)*f*(a + b
*Tan[e + f*x])*(c + d*Tan[e + f*x])^(3/2)) - (d*(2*a^3*d^2*(B*c^2 + 2*c*C*d
 - B*d^2) + 2*b^3*c*(2*c^3*C - 3*B*c^2*d - B*d^3) - a*b^2*(B*c^4 - 4*c*C*d^
3 + 3*B*d^4) + a^2*b*(5*c^4*C - 6*B*c^3*d + 2*c^2*C*d^2 - 2*B*c*d^3 + C*d^4
) - A*(4*a^3*c*d^3 + 4*a*b^2*c*d^3 - 4*a^2*b*d^2*(2*c^2 + d^2) - b^3*(c^4 +
 10*c^2*d^2 + 5*d^4))))/((a^2 + b^2)*(b*c - a*d)^3*(c^2 + d^2)^2*f*Sqrt[c +
 d*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{A + B\tan(e+fx) + C\tan^2(e+fx)}{(a+b\tan(e+fx))^2(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{Ab^2 - a(bB - aC)}{\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{d\left(2a^2Ad^2 + 2b^2c(cC - Bd) - 3abB\left(c^2+d^2\right) + Ab^2\left(3c^2 + 5\right.\right.}{3\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f(c + d\,\mathrm{t}}$$

$$= -\frac{d\left(2a^2Ad^2 + 2b^2c(cC - Bd) - 3abB\left(c^2+d^2\right) + Ab^2\left(3c^2 + 5\right.\right.}{3\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f(c + d\,\mathrm{t}}$$

$$= -\frac{d\left(2a^2Ad^2 + 2b^2c(cC - Bd) - 3abB\left(c^2+d^2\right) + Ab^2\left(3c^2 + 5\right.\right.}{3\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f(c + d\,\mathrm{t}}$$

$$= -\frac{d\left(2a^2Ad^2 + 2b^2c(cC - Bd) - 3abB\left(c^2+d^2\right) + Ab^2\left(3c^2 + 5\right.\right.}{3\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f(c + d\,\mathrm{t}}$$

$$= -\frac{d\left(2a^2Ad^2 + 2b^2c(cC - Bd) - 3abB\left(c^2+d^2\right) + Ab^2\left(3c^2 + 5\right.\right.}{3\left(a^2+b^2\right)(bc-ad)^2\left(c^2+d^2\right)f(c + d\,\mathrm{t}}$$

$$= -\frac{b^{3/2}\left(7a^3bBd - 5a^4Cd + b^4(2Bc - 5Ad) + ab^3(4Ac - 4cC + \right.}{\left(a^2+b^2\right)^2(b}$$

$$= -\frac{(iA + B - iC)\tanh^{-1}\left(\frac{\sqrt{c+d\tan(e+fx)}}{\sqrt{c-id}}\right)}{(a-ib)^2(c-id)^{5/2}f} - \frac{(B - i(A-C))\tanh}{(a+ib)^2(c}$$

Mathematica [B] time = 6.43, size = 6052, normalized size = 8.91

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^2*(
c + d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [B] time = 0.85, size = 67570, normalized size = 99.51

output too large to display

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e))^(5/
2),x)
```

```
[Out] result too large to display
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^2/(c+d*tan(f*x+e
))^(5/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(a*d-b*c>0)', see `assume?` for more
 details)Is a*d-b*c positive or negative?
```

mupad **[F(-1)]** time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^2*(c + d*
tan(e + f*x))^(5/2)),x)
```

```
[Out] \text{Hanged}
```

sympy **[F]** time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)^{2}\left(c + d\tan\left(e + fx\right)\right)^{\frac{5}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**2/(c+d*tan(f*x
+e))**(5/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**2*
(c + d*tan(e + f*x))**(5/2)), x)
```

3.128 $\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}\,\big(A+B\tan(e$

Optimal. Leaf size=679

$$\frac{\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}\,\left(64bd^3\left(a^2B+2ab(A-C)-b^2B\right)-(bc-ad)\left(16bd^2(aB+Ab-bC)+(bc\right.\right.}{64bd^3f}$$

```
[Out] -1/64*(5*a^4*C*d^4-20*a^3*b*d^3*(2*B*d+C*c)+30*a^2*b^2*d^2*(c^2*C-4*B*c*d-8
*(A-C)*d^2)-20*a*b^3*d*(c^3*C-2*B*c^2*d+8*c*(A-C)*d^2-16*B*d^3)+b^4*(5*c^4*
C-8*B*c^3*d+16*c^2*(A-C)*d^2+64*B*c*d^3+128*(A-C)*d^4))*arctanh(d^(1/2)*(a+
b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(3/2)/d^(7/2)/f-(a-I*
b)^(5/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(
1/2)/(c+d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/f-(a+I*b)^(5/2)*(B-I*(A-C))*arct
anh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/
2))*(c+I*d)^(1/2)/f+1/64*(64*b*(a^2*B-b^2*B+2*a*b*(A-C))*d^3-(-a*d+b*c)*(16
*b*(A*b+B*a-C*b)*d^2+(-a*d+b*c)*(-8*B*b*d-5*C*a*d+5*C*b*c)))*(a+b*tan(f*x+e
))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b/d^3/f+1/32*(16*b*(A*b+B*a-C*b)*d^2+(-a*d+
b*c)*(-8*B*b*d-5*C*a*d+5*C*b*c))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3
/2)/d^3/f-1/24*(-8*B*b*d-5*C*a*d+5*C*b*c)*(a+b*tan(f*x+e))^(3/2)*(c+d*tan(f
*x+e))^(3/2)/d^2/f+1/4*C*(a+b*tan(f*x+e))^(5/2)*(c+d*tan(f*x+e))^(3/2)/d/f
```

Rubi [A] time = 9.93, antiderivative size = 679, normalized size of antiderivative = 1.00, number of steps used = 16, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{\left(30a^2b^2d^2\left(-8d^2(A-C)-4Bcd+c^2C\right)-20a^3bd^3(2Bd+cC)+5a^4Cd^4-20ab^3d\left(8cd^2(A-C)-2Bc^2d-16Bd\right.\right.}{64b^{3/2}d^{7/2}f}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^(5/2)*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^(5/2)*(I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqr
t[a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) - ((a
+ I*b)^(5/2)*(B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f - ((5*a^4*C*d
^4 - 20*a^3*b*d^3*(c*C + 2*B*d) + 30*a^2*b^2*d^2*(c^2*C - 4*B*c*d - 8*(A -
C)*d^2) - 20*a*b^3*d*(c^3*C - 2*B*c^2*d + 8*c*(A - C)*d^2 - 16*B*d^3) + b^4
*(5*c^4*C - 8*B*c^3*d + 16*c^2*(A - C)*d^2 + 64*B*c*d^3 + 128*(A - C)*d^4))
*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x
]])])/(64*b^(3/2)*d^(7/2)*f) + ((64*b*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3 -
```

```
 (b*c - a*d)*(16*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 8*b*B*d -
 5*a*C*d)))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(64*b*d^3*f)
 + ((16*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 8*b*B*d - 5*a*C*d)
)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(32*d^3*f) - ((5*b*c
*C - 8*b*B*d - 5*a*C*d)*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(3/
2))/(24*d^2*f) + (C*(a + b*Tan[e + f*x])^(5/2)*(c + d*Tan[e + f*x])^(3/2))/
(4*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
```

```
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$\int (a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{C(a+b\tan(e+fx))^{5/2}(}{4df}$

$= -\frac{(5bcC-8bBd-5aCd}{}$

$= \frac{\left(16b(Ab+aB-bC)d^2\right.}{}$

$= \frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$

$= \frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$

$= \frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$

$= \frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$

$= \frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$

$= \frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$

$= -\frac{\left(5a^4Cd^4-20a^3bd^3(c\right.}{}$

$= -\frac{(a-ib)^{5/2}(iA+B-iC}{}$

Mathematica [A] time = 9.95, size = 1202, normalized size = 1.77

$$\frac{C(c+d\tan(e+fx))^{3/2}(a+b\tan(e+fx))^{5/2}}{4df}+\frac{\frac{(-5bcC+5adC+8bBd)(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{6df}+\frac{3\left(16b(Ab-Cb+aB)d^2+(bc-ad\right.}{}}{}$$

Warning: Unable to verify antiderivative.

```
[In] Integrate[(a + b*Tan[e + f*x])^(5/2)*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e
+ f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(5/2)*(c + d*Tan[e + f*x])^(3/2))/(4*d*f) + (((-5*b
*c*C + 8*b*B*d + 5*a*C*d)*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(
3/2))/(6*d*f) + ((3*(16*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 8*
b*B*d - 5*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(8*d
*f) + (((24*b*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3 - (3*(b*c - a*d)*(16*b*(A
*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 8*b*B*d - 5*a*C*d)))/8)*Sqrt[a
 + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(b*f) + ((24*b*d^3*(b*(3*a^2*b
*(A*c - c*C - B*d) - b^3*(A*c - c*C - B*d) + a^3*(B*c + (A - C)*d) - 3*a*b^
2*(B*c + (A - C)*d)) + Sqrt[-b^2]*(a^3*(A*c - c*C - B*d) - 3*a*b^2*(A*c - c
*C - B*d) - 3*a^2*b*(B*c + (A - C)*d) + b^3*(B*c + (A - C)*d)))*ArcTanh[(Sq
rt[-c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*
Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)
/b]) - (24*b*d^3*(b*(3*a^2*b*(A*c - c*C - B*d) - b^3*(A*c - c*C - B*d) + a^
3*(B*c + (A - C)*d) - 3*a*b^2*(B*c + (A - C)*d)) - Sqrt[-b^2]*(a^3*(A*c - c
*C - B*d) - 3*a*b^2*(A*c - c*C - B*d) - 3*a^2*b*(B*c + (A - C)*d) + b^3*(B*
c + (A - C)*d)))*ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + Sqrt[-b^2]]
*Sqrt[c + (Sqrt[-b^2]*d)/b]) - (3*Sqrt[b]*Sqrt[c - (a*d)/b]*Sqrt[(c/(c - (a
*d)/b) - (a*d)/(b*(c - (a*d)/b)))^(-1)]*Sqrt[c/(c - (a*d)/b) - (a*d)/(b*(c
- (a*d)/b))]*(5*a^4*C*d^4 - 20*a^3*b*d^3*(c*C + 2*B*d) + 30*a^2*b^2*d^2*(c^
2*C - 4*B*c*d - 8*(A - C)*d^2) - 20*a*b^3*d*(c^3*C - 2*B*c^2*d + 8*c*(A - C
)*d^2 - 16*B*d^3) + b^4*(5*c^4*C - 8*B*c^3*d + 16*c^2*(A - C)*d^2 + 64*B*c*
d^3 + 128*(A - C)*d^4))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]
*Sqrt[c - (a*d)/b]*Sqrt[c/(c - (a*d)/b) - (a*d)/(b*(c - (a*d)/b))])]*Sqrt[(
c + d*Tan[e + f*x])/(c - (a*d)/b)])/(8*Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(
b^2*f))/(2*d))/(3*d))/(4*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \sqrt{c+d\tan\left(fx+e\right)}\,\left(a+b\tan\left(fx+e\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int {\left(a+b\,\mathrm{tan}\left(e+f\,x\right)\right)}^{5/2}\,\sqrt{c+d\,\mathrm{tan}\left(e+f\,x\right)}\,\left(C\,{\mathrm{tan}\left(e+f\,x\right)}^2+B\,\mathrm{tan}\left(e+f\,x\right)+A\right) \,d x$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^(5/2)*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^(5/2)*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2), x)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(a+b*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
```

```
[Out] Timed out
```

3.129 $\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}\,\big(A+B\tan(e$

Optimal. Leaf size=505

$$-\frac{\left(a^3Cd^3-3a^2bd^2(2Bd+cC)+3ab^2d\left(-8d^2(A-C)-4Bcd+c^2C\right)-\left(b^3\left(8cd^2(A-C)-2Bc^2d-16Bd^3+c^3C\right.\right.\right.}{8b^{3/2}d^{5/2}f}$$

```
[Out] -1/8*(a^3*C*d^3-3*a^2*b*d^2*(2*B*d+C*c)+3*a*b^2*d*(c^2*C-4*B*c*d-8*(A-C)*d^
2)-b^3*(c^3*C-2*B*c^2*d+8*c*(A-C)*d^2-16*B*d^3))*arctanh(d^(1/2)*(a+b*tan(f
*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(3/2)/d^(5/2)/f-(a-I*b)^(3/2
)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c
+d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/f+(a+I*b)^(3/2)*(I*A-B-I*C)*arctanh((c+
I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(c+
I*d)^(1/2)/f+1/8*(8*b*(A*b+B*a-C*b)*d^2+(-a*d+b*c)*(-2*B*b*d-C*a*d+C*b*c))*
(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b/d^2/f-1/4*(-2*B*b*d-C*a*d+C
*b*c)*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)/d^2/f+1/3*C*(a+b*tan(f*
x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)/d/f
```

Rubi [A] time = 7.34, antiderivative size = 505, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{\left(-3a^2bd^2(2Bd+cC)+a^3Cd^3+3ab^2d\left(-8d^2(A-C)-4Bcd+c^2C\right)+b^3\left(-\left(8cd^2(A-C)-2Bc^2d-16Bd^3+\right.\right.\right.}{8b^{3/2}d^{5/2}f}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^(3/2)*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^(3/2)*(I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqr
t[a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) + ((a
+ I*b)^(3/2)*(I*A - B - I*C)*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f - ((a^3*C*d^3
 - 3*a^2*b*d^2*(c*C + 2*B*d) + 3*a*b^2*d*(c^2*C - 4*B*c*d - 8*(A - C)*d^2)
- b^3*(c^3*C - 2*B*c^2*d + 8*c*(A - C)*d^2 - 16*B*d^3))*ArcTanh[(Sqrt[d]*Sq
rt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(8*b^(3/2)*d^(
5/2)*f) + ((8*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(b*c*C - 2*b*B*d - a*C*
d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(8*b*d^2*f) - ((b*c*
C - 2*b*B*d - a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(
4*d^2*f) + (C*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(3/2))/(3*d*f
)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int (a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}\,\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{C(a+b\tan(e+fx))^{3/2}(c}{3df}$$

$$= -\frac{(bcC-2bBd-aCd)\sqrt{a}}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(\right.}{}$$

$$= -\frac{\left(a^3Cd^3-3a^2bd^2(cC+2\right.}{}$$

$$= -\frac{(a-ib)^{3/2}(iA+B-iC)}{}$$

Mathematica [A] time = 8.88, size = 835, normalized size = 1.65

$$\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}{3df}+\frac{\frac{3\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}\left(8b(Ab-Cb+aB)d^2+(bc-ad)(bcC-adC-2bBd)\right)}{4bf}+\frac{6b\left(\sqrt{-b^2}\left((Ac-Cc-\cdots\right.\right.}{\cdots}}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^(3/2)*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e
+ f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(3/2))/(3*d*f) + ((-3*(b
*c*C - 2*b*B*d - a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2)
)/(4*d*f) + ((3*(8*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(b*c*C - 2*b*B*d -
 a*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*b*f) + ((6*b
*d^2*(Sqrt[-b^2]*(a^2*(A*c - c*C - B*d) - b^2*(A*c - c*C - B*d) - 2*a*b*(B*
c + (A - C)*d)) + b*(2*a*b*(A*c - c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*
(B*c + (A - C)*d)))*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e +
 f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[-a + Sqrt[
-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) + (6*b*d^2*(Sqrt[-b^2]*(a^2*(A*c - c*C
- B*d) - b^2*(A*c - c*C - B*d) - 2*a*b*(B*c + (A - C)*d)) - b*(2*a*b*(A*c -
 c*C - B*d) + a^2*(B*c + (A - C)*d) - b^2*(B*c + (A - C)*d)))*ArcTanh[(Sqrt
[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt
[c + d*Tan[e + f*x]])]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) -
 (3*Sqrt[b]*Sqrt[c - (a*d)/b]*(a^3*C*d^3 - 3*a^2*b*d^2*(c*C + 2*B*d) + 3*a*
b^2*d*(c^2*C - 4*B*c*d - 8*(A - C)*d^2) - b^3*(c^3*C - 2*B*c^2*d + 8*c*(A -
 C)*d^2 - 16*B*d^3))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sq
rt[c - (a*d)/b])]*Sqrt[(b*c + b*d*Tan[e + f*x])/(b*c - a*d)])/(4*Sqrt[d]*Sq
rt[c + d*Tan[e + f*x]]))/(b^2*f))/(2*d))/(3*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \sqrt{c+d\tan\left(fx+e\right)}\,\left(a+b\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{\frac{3}{2}}\sqrt{d\tan\left(fx+e\right)+c}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^(3/2
)*sqrt(d*tan(f*x + e) + c), x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int {\left(a+b\,\mathrm{tan}\left(e+f\,x\right)\right)}^{3/2}\,\sqrt{c+d\,\mathrm{tan}\left(e+f\,x\right)}\,\left(C\,{\mathrm{tan}\left(e+f\,x\right)}^2+B\,\mathrm{tan}\left(e+f\,x\right)+A\right) \,d x$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^(3/2)*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^(3/2)*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a+b\tan\left(e+fx\right)\right)^{\frac{3}{2}}\sqrt{c+d\tan\left(e+fx\right)}\,\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(a+b*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(3/2)*sqrt(c + d*tan(e + f*x))*(A + B*tan(e
+ f*x) + C*tan(e + f*x)**2), x)
```

3.130 $\int \sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\,\left(A+B\tan(e\right.$

Optimal. Leaf size=381

$$-\frac{\left(a^2Cd^2-2abd(2Bd+cC)+b^2\left(-8d^2(A-C)-4Bcd+c^2C\right)\right)\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{3/2}d^{3/2}f}-\frac{\sqrt{a-ib}\sqrt{c-id}\,(iA+}{}$$

```
[Out] -1/4*(a^2*C*d^2-2*a*b*d*(2*B*d+C*c)+b^2*(c^2*C-4*B*c*d-8*(A-C)*d^2))*arctan
h(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(3/2)/d^
(3/2)/f-(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1
/2)/(c+d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)*(c-I*d)^(1/2)/f-(B-I*(A-C))*arcta
nh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2
))*(a+I*b)^(1/2)*(c+I*d)^(1/2)/f-1/4*(-4*B*b*d-C*a*d+C*b*c)*(a+b*tan(f*x+e)
)^(1/2)*(c+d*tan(f*x+e))^(1/2)/b/d/f+1/2*C*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(
f*x+e))^(3/2)/d/f
```

Rubi [A] time = 4.97, antiderivative size = 383, normalized size of antiderivative = 1.01, number of steps used = 14, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{\left(a^2Cd^2-2abd(2Bd+cC)+b^2\left(-8d^2(A-C)-4Bcd+c^2C\right)\right)\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{3/2}d^{3/2}f}-\frac{\sqrt{a-ib}\sqrt{c-id}\,(iA+}{}$$

Antiderivative was successfully verified.

```
[In] Int[Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqrt[
a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) + (Sqrt[
a + I*b]*(I*A - B - I*C)*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Ta
n[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f - ((a^2*C*d^2 - 2
*a*b*d*(c*C + 2*B*d) + b^2*(c^2*C - 4*B*c*d - 8*(A - C)*d^2))*ArcTanh[(Sqrt
[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(4*b^(3/
2)*d^(3/2)*f) - ((b*c*C - 4*b*B*d - a*C*d)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c
+ d*Tan[e + f*x]])/(4*b*d*f) + (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f
*x])^(3/2))/(2*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
```

```
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
```

```
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}\,\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{C\sqrt{a+b\tan(e+fx)}\,(c+\cdots}{2df}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a\cdots}}{\cdots}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a\cdots}}{\cdots}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a\cdots}}{\cdots}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a\cdots}}{\cdots}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a\cdots}}{\cdots}$$

$$= -\frac{(bcC-4bBd-aCd)\sqrt{a\cdots}}{\cdots}$$

$$= -\frac{\left(a^2Cd^2-2abd(cC+2B\cdots\right.}{\cdots}$$

$$= -\frac{\sqrt{a-ib}\,(iA+B-iC)\sqrt{\cdots}}{\cdots}$$

Mathematica [A] time = 7.79, size = 619, normalized size = 1.62

$$-\frac{\sqrt{b}\sqrt{c-\frac{ad}{b}}\left(a^2Cd^2-2abd(2Bd+cC)+b^2\left(-8d^2(A-C)-4Bcd+c^2C\right)\right)\sqrt{\frac{bc+bd\tan(e+fx)}{bc-ad}}\,\sinh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c-\frac{ad}{b}}}\right)}{2\sqrt{d}\sqrt{c+d\tan(e+fx)}}+\frac{2bd\left(b(aAd+aBc-aCd+Abc-bBd-bcC)-\sqrt{-b^2}\,(-a(Ac-\cdots\right.}{\sqrt{\sqrt{-b^2}\cdots}}$$

Antiderivative was successfully verified.

```
[In] Integrate[Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e +
f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(2*d*f) + (((-(b*c*
C) + 4*b*B*d + a*C*d)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(2
*b*f) + ((2*b*d*(b*(A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d) - Sqrt[-
b^2]*(b*B*c + b*(A - C)*d - a*(A*c - c*C - B*d)))*ArcTanh[(Sqrt[-c + (Sqrt[
-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan
[e + f*x]])]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (2*b*d*
(b*(A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d) + Sqrt[-b^2]*(b*B*c + b*
(A - C)*d - a*(A*c - c*C - B*d)))*ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[
a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqr
t[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) - (Sqrt[b]*Sqrt[c - (a*d)/b]*
(a^2*C*d^2 - 2*a*b*d*(c*C + 2*B*d) + b^2*(c^2*C - 4*B*c*d - 8*(A - C)*d^2))
*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sq
rt[(b*c + b*d*Tan[e + f*x])/(b*c - a*d)])/(2*Sqrt[d]*Sqrt[c + d*Tan[e + f*x
]]))/(b^2*f))/(2*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \sqrt{a+b\tan\left(fx+e\right)}\,\sqrt{c+d\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^2 + B\tan\left(fx+e\right) + A\right)\sqrt{b\tan\left(fx+e\right)+a}\,\sqrt{d\tan\left(fx+e\right)+c}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(b*tan(f*x + e) + a)*
sqrt(d*tan(f*x + e) + c), x)
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^(1/2)*(c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \sqrt{a+b\tan\left(e+fx\right)}\,\sqrt{c+d\tan\left(e+fx\right)}\,\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
```

```
[Out] Integral(sqrt(a + b*tan(e + f*x))*sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f
*x) + C*tan(e + f*x)**2), x)
```

3.131 $\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx$

Optimal. Leaf size=287

$$-\frac{\sqrt{c-id}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}}-\frac{\sqrt{c+id}\,(B-i(A-C))\tanh^{-1}\left(\frac{\sqrt{c+id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a+ib}}+\frac{(-aC\cdots}{}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/f/(a-I*b)^(1/2)-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(c+I
*d)^(1/2)/f/(a+I*b)^(1/2)+(2*B*b*d-C*a*d+C*b*c)*arctanh(d^(1/2)*(a+b*tan(f*
x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(3/2)/f/d^(1/2)+C*(a+b*tan(f*
x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b/f
```

Rubi [A] time = 2.63, antiderivative size = 287, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{\sqrt{c-id}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}}-\frac{\sqrt{c+id}\,(B-i(A-C))\tanh^{-1}\left(\frac{\sqrt{c+id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a+ib}}+\frac{(-aC\cdots}{}$$

Antiderivative was successfully verified.

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sqrt
[a + b*Tan[e + f*x]],x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a - I*b]*f)) - ((B
- I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])
/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*f) + ((b*c*C + 2
*b*B*d - a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c
+ d*Tan[e + f*x]])])/(b^(3/2)*Sqrt[d]*f) + (C*Sqrt[a + b*Tan[e + f*x]]*Sqrt
[c + d*Tan[e + f*x]])/(b*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
```

```
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx = \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{\int}{}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{\mathrm{Su}}{}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{\mathrm{Su}}{}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{\mathrm{Su}}{}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{\mathrm{Su}}{}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{bf} + \frac{((}{}$$

$$= \frac{(bcC+2bBd-aCd)\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{3/2}\sqrt{d}\,f}$$

$$= -\frac{(iA+B-iC)\sqrt{c-id}\,\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e}}{\sqrt{a-ib}\sqrt{c+d\tan(e}}\right.}{\sqrt{a-ib}\,f}$$

Mathematica [A] time = 4.24, size = 441, normalized size = 1.54

$$\frac{\frac{b\left(\sqrt{-b^2}\,(Ac-Bd-cC)+bd(A-C)+bBc\right)\tanh^{-1}\left(\frac{\sqrt{\frac{\sqrt{-b^2}\,d}{b}-c}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{\sqrt{-b^2}-a}\,\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{\sqrt{-b^2}-a}\,\sqrt{\frac{\sqrt{-b^2}\,d}{b}-c}}+\frac{b\left(\sqrt{-b^2}\,(Ac-Bd-cC)-b(d(A-C)+Bc)\right)\tanh^{-1}\left(\frac{\sqrt{\frac{\sqrt{-b^2}\,d}{b}+c}\,\sqrt{a+b\tan(e+}}{\sqrt{a+\sqrt{-b^2}}\,\sqrt{c+d\tan(e+f}}\right.}{\sqrt{a+\sqrt{-b^2}}\,\sqrt{\frac{\sqrt{-b^2}\,d}{b}+c}}}{b^2 f}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/Sqrt[a + b*Tan[e + f*x]],x]
```

```
[Out] ((b*(b*B*c + b*(A - C)*d + Sqrt[-b^2]*(A*c - c*C - B*d))*ArcTanh[(Sqrt[-c +
 (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c
+ d*Tan[e + f*x]])]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) +
(b*(Sqrt[-b^2]*(A*c - c*C - B*d) - b*(B*c + (A - C)*d))*ArcTanh[(Sqrt[c + (
Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d
*Tan[e + f*x]])]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) + b*C*S
qrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]] + (Sqrt[b]*Sqrt[c - (a*d)/
b]*(b*c*C + 2*b*B*d - a*C*d)*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sq
rt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)])/(Sqrt
[d]*Sqrt[c + d*Tan[e + f*x]]))/(b^2*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c + d\tan\left(fx + e\right)}\ \left(A + B\tan\left(fx + e\right) + C\left(\tan^2\left(fx + e\right)\right)\right)}{\sqrt{a + b\tan\left(fx + e\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(C\tan\left(fx + e\right)^2 + B\tan\left(fx + e\right) + A\right)\sqrt{d\tan\left(fx + e\right) + c}}{\sqrt{b\tan\left(fx + e\right) + a}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(d*tan(f*x + e) + c)/
sqrt(b*tan(f*x + e) + a), x)
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(1/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c + d\tan\left(e + fx\right)}\ \left(A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)\right)}{\sqrt{a + b\tan\left(e + fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
sqrt(a + b*tan(e + f*x)), x)
```

3.132 $\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=300

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{bf\left(a^2+b^2\right)\sqrt{a+b\tan(e+fx)}}-\frac{\sqrt{c-id}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{3/2}}-\frac{\sqrt{c+id}\,(B-i(A$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/(a-I*b)^(3/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(c+I
*d)^(1/2)/(a+I*b)^(3/2)/f+2*C*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2
)/(c+d*tan(f*x+e))^(1/2))*d^(1/2)/b^(3/2)/f-2*(A*b^2-a*(B*b-C*a))*(c+d*tan(
f*x+e))^(1/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(1/2)
```

Rubi [A] time = 3.80, antiderivative size = 300, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3645, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{bf\left(a^2+b^2\right)\sqrt{a+b\tan(e+fx)}}-\frac{\sqrt{c-id}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{3/2}}-\frac{\sqrt{c+id}\,(B-i(A$$

Antiderivative was successfully verified.

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*f)) - ((
B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*f) + (2*C*Sq
rt[d]*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e
+ f*x]])])/(b^(3/2)*f) - (2*(A*b^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]
])/(b*(a^2 + b^2)*f*Sqrt[a + b*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
```

```
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{b\left(a^2+b^2\right)f\sqrt{a+b\tan(e+fx)}} +$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{b\left(a^2+b^2\right)f\sqrt{a+b\tan(e+fx)}} +$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{b\left(a^2+b^2\right)f\sqrt{a+b\tan(e+fx)}} +$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{b\left(a^2+b^2\right)f\sqrt{a+b\tan(e+fx)}} +$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{b\left(a^2+b^2\right)f\sqrt{a+b\tan(e+fx)}} +$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{b\left(a^2+b^2\right)f\sqrt{a+b\tan(e+fx)}} +$$

$$= \frac{2C\sqrt{d}\,\tanh^{-1}\left(\frac{\sqrt{d}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\,\sqrt{c+d\tan(e+fx)}}\right)}{b^{3/2}f} - \frac{2\left(Ab^2-a\right.}{b\left(a^2\right.}$$

$$= -\frac{(iA+B-iC)\sqrt{c-id}\,\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e}}\right.}{(a-ib)^{3/2}f}$$

Mathematica [C] time = 35.73, size = 621058, normalized size = 2070.19

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x])^(3/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c+d\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(3/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c+d\tan\left(e+fx\right)}\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)}{\left(a+b\tan\left(e+fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x))**(3/2), x)
```

3.133 $\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=370

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3bf\left(a^2+b^2\right)(a+b\tan(e+fx))^{3/2}}-\frac{2\sqrt{c+d\tan(e+fx)}\left(a^4Cd+2a^3bBd-a^2b^2(5Ad+3Bc-7Cd)+\right.}{3bf\left(a^2+b^2\right)^2(bc-ad)\sqrt{a+b\tan}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/(a-I*b)^(5/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(c+I
*d)^(1/2)/(a+I*b)^(5/2)/f-2/3*(2*a^3*b*B*d+a^4*C*d+b^4*(A*d+3*B*c)+2*a*b^3*
(3*A*c-2*B*d-3*C*c)-a^2*b^2*(5*A*d+3*B*c-7*C*d))*(c+d*tan(f*x+e))^(1/2)/b/(
a^2+b^2)^2/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(1/2)-2/3*(A*b^2-a*(B*b-C*a))*(c+d
*tan(f*x+e))^(1/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] time = 2.05, antiderivative size = 370, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.122, Rules used = {3645, 3649, 3616, 3615, 93, 208}

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3bf\left(a^2+b^2\right)(a+b\tan(e+fx))^{3/2}}-\frac{2\sqrt{c+d\tan(e+fx)}\left(-a^2b^2(5Ad+3Bc-7Cd)+2a^3bBd+a^4Cd\right.}{3bf\left(a^2+b^2\right)^2(bc-ad)\sqrt{a+b\tan}}$$

Antiderivative was successfully verified.

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(5/2)*f)) - ((
B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(5/2)*f) - (2*(A*b
^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]])/(3*b*(a^2 + b^2)*f*(a + b*Tan
[e + f*x])^(3/2)) - (2*(2*a^3*b*B*d + a^4*C*d + b^4*(3*B*c + A*d) + 2*a*b^3
*(3*A*c - 3*c*C - 2*B*d) - a^2*b^2*(3*B*c + 5*A*d - 7*C*d))*Sqrt[c + d*Tan[
e + f*x]])/(3*b*(a^2 + b^2)^2*(b*c - a*d)*f*Sqrt[a + b*Tan[e + f*x]])
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
```

```
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
```

```
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}}$$

$$= -\frac{(iA+B-iC)\sqrt{c-id}\,\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan}}{\sqrt{a-ib}\,\sqrt{c+d\tan}}\right.}{(a-ib)^{5/2}f}$$

Mathematica [A] time = 7.01, size = 600, normalized size = 1.62

$$-\frac{C\sqrt{c+d\tan(e+fx)}}{bf(a+b\tan(e+fx))^{3/2}}-\frac{-\frac{2\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}b^2(-aCd-2Abc+3bcC)-a\left(-\frac{1}{2}a(-aCd-2bBd+bcC)-\left(b^2(d(A-C)+Bc)\right)\right)\right)}{3f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))^{3/2}}-\frac{2\left(-\frac{2\sqrt{c+d\tan(}}{}\right.}{}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x])^(5/2),x]
```

```
[Out] -((C*Sqrt[c + d*Tan[e + f*x]])/(b*f*(a + b*Tan[e + f*x])^(3/2))) - ((-2*((b
^2*(-2*A*b*c + 3*b*c*C - a*C*d))/2 - a*(-(b^2*(B*c + (A - C)*d)) - (a*(b*c*
C - 2*b*B*d - a*C*d))/2))*Sqrt[c + d*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c - a
*d)*f*(a + b*Tan[e + f*x])^(3/2)) - (2*((-3*b*(b*c - a*d)*(((a + I*b)^2*(I*
A + B - I*C)*Sqrt[-c + I*d]*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/Sqrt[-a + I*b] + ((a - I*b)^
2*(B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f
*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/Sqrt[a + I*b]))/(2*(a^2 +
b^2)*f) - (2*((b^2*(b*c - a*d)*(a^2*C*d + b^2*(3*B*c + A*d) + a*b*(3*A*c -
3*c*C - B*d)))/2 - a*((a*(2*A*b^2 - 2*a*b*B - a^2*C - 3*b^2*C)*d*(b*c - a*d
))/2 - (3*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C - a*A*d - b*B*d + a*C*d))/
2))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e +
 f*x]])))/(3*(a^2 + b^2)*(b*c - a*d)))/b
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c+d\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(((2*b*d+2*a*c)^2>0)', see `assume?`
 for more details)Is ((2*b*d+2*a*c)^2    -4*((a*c-b*d)^2       -((-a*d)-b*c
)        *(a*d+b*c)))    ^2 positive or zero?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(5/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c + d \tan \left(e + f x\right)} \left(A + B \tan \left(e + f x\right) + C \tan^{2} \left(e + f x\right)\right)}{\left(a + b \tan \left(e + f x\right)\right)^{\frac{5}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x))**(5/2), x)
```

3.134 $\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx$

Optimal. Leaf size=597

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{5bf\left(a^2+b^2\right)(a+b\tan(e+fx))^{5/2}}-\frac{2\sqrt{c+d\tan(e+fx)}\left(a^4Cd+4a^3bBd-a^2b^2(9Ad+5Bc-11Cd)+\right.}{15bf\left(a^2+b^2\right)^2(bc-ad)(a+b\tan(e}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(c-I*d)^(1/2)/(a-I*b)^(7/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(c+I
*d)^(1/2)/(a+I*b)^(7/2)/f+2/15*(8*a^5*b*B*d^2+2*a^6*C*d^2-a^4*b^2*d*(33*A*d
+25*B*c-39*C*d)-a^2*b^4*(45*A*c^2-29*A*d^2-90*B*c*d-45*C*c^2+23*C*d^2)+a^3*
b^3*(80*c*(A-C)*d+B*(15*c^2-49*d^2))-a*b^5*(40*c*(A-C)*d+B*(45*c^2-3*d^2))-
b^6*(5*c*(B*d+3*C*c)-A*(15*c^2+2*d^2)))*(c+d*tan(f*x+e))^(1/2)/b/(a^2+b^2)^
3/(-a*d+b*c)^2/f/(a+b*tan(f*x+e))^(1/2)-2/5*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*
x+e))^(1/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(5/2)-2/15*(4*a^3*b*B*d+a^4*C*d+
b^4*(A*d+5*B*c)+2*a*b^3*(5*A*c-3*B*d-5*C*c)-a^2*b^2*(9*A*d+5*B*c-11*C*d))*(
c+d*tan(f*x+e))^(1/2)/b/(a^2+b^2)^2/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] time = 3.59, antiderivative size = 597, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 6, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}} = 0.122$, Rules used = {3645, 3649, 3616, 3615, 93, 208}

$$\frac{2\sqrt{c+d\tan(e+fx)}\left(a^3b^3\left(80cd(A-C)+B\left(15c^2-49d^2\right)\right)-a^2b^4\left(45Ac^2-29Ad^2-90Bcd-45c^2C+23Cd^2\right)\right.}{15bf\left(a^2+b^2\right)}$$

Antiderivative was successfully verified.

```
[In] Int[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a +
 b*Tan[e + f*x])^(7/2),x]
```

```
[Out] -(((I*A + B - I*C)*Sqrt[c - I*d]*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(7/2)*f)) - ((
B - I*(A - C))*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(7/2)*f) - (2*(A*b
^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]])/(5*b*(a^2 + b^2)*f*(a + b*Tan
[e + f*x])^(5/2)) - (2*(4*a^3*b*B*d + a^4*C*d + b^4*(5*B*c + A*d) + 2*a*b^3
*(5*A*c - 5*c*C - 3*B*d) - a^2*b^2*(5*B*c + 9*A*d - 11*C*d))*Sqrt[c + d*Tan
[e + f*x]])/(15*b*(a^2 + b^2)^2*(b*c - a*d)*f*(a + b*Tan[e + f*x])^(3/2)) +
 (2*(8*a^5*b*B*d^2 + 2*a^6*C*d^2 - a^4*b^2*d*(25*B*c + 33*A*d - 39*C*d) - a
```

```
^2*b^4*(45*A*c^2 - 45*c^2*C - 90*B*c*d - 29*A*d^2 + 23*C*d^2) + a^3*b^3*(80
*c*(A - C)*d + B*(15*c^2 - 49*d^2)) - a*b^5*(40*c*(A - C)*d + B*(45*c^2 - 3
*d^2)) - b^6*(5*c*(3*c*C + B*d) - A*(15*c^2 + 2*d^2)))*Sqrt[c + d*Tan[e + f
*x]])/(15*b*(a^2 + b^2)^3*(b*c - a*d)^2*f*Sqrt[a + b*Tan[e + f*x]])
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
```

```
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{\sqrt{c+d\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{5b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{5/2}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{5b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{5/2}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{5b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{5/2}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{5b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{5/2}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{5b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{5/2}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{5b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{5/2}}$$

$$= -\frac{(iA+B-iC)\sqrt{c-id}\,\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan}}{\sqrt{a-ib}\,\sqrt{c+d\tan}}\right.}{(a-ib)^{7/2}f}$$

Mathematica [A] time = 7.35, size = 1109, normalized size = 1.86

$$-\frac{\sqrt{c+d\tan(e+fx)}\,C}{2bf(a+b\tan(e+fx))^{5/2}}-\frac{-\frac{2\sqrt{c+d\tan(e+fx)}\left(\frac{1}{2}b^2(-4Abc+5bCc-aCd)-a\left(-2(Bc+(A-C)d)b^2-\frac{1}{2}a(bcC-adC-4bBd)\right)\right)}{5\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{5/2}}-\frac{2\left(-\frac{2\sqrt{c+d\tan(e+\cdots}}{\cdots}\cdots\right)}{\cdots}}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Sqrt[c + d*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(a + b*Tan[e + f*x])^(7/2),x]
```

```
[Out] -1/2*(C*Sqrt[c + d*Tan[e + f*x]])/(b*f*(a + b*Tan[e + f*x])^(5/2)) - ((-2*(
(b^2*(-4*A*b*c + 5*b*c*C - a*C*d))/2 - a*(-2*b^2*(B*c + (A - C)*d) - (a*(b*
c*C - 4*b*B*d - a*C*d))/2))*Sqrt[c + d*Tan[e + f*x]])/(5*(a^2 + b^2)*(b*c -
 a*d)*f*(a + b*Tan[e + f*x])^(5/2)) - (2*((-2*(b^2*(b*c - a*d)*(a^2*C*d + b
^2*(5*B*c + A*d) + a*b*(5*A*c - 5*c*C - B*d)) - a*(a*(4*A*b^2 - 4*a*b*B - a
^2*C - 5*b^2*C)*d*(b*c - a*d) - 5*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C -
a*A*d - b*B*d + a*C*d)))*Sqrt[c + d*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c - a*
d)*f*(a + b*Tan[e + f*x])^(3/2)) - (2*((-15*b*(b*c - a*d)^2*(((I*a - b)^3*(
A - I*B - C)*Sqrt[-c + I*d]*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/Sqrt[-a + I*b] - ((I*a + b)^
3*(A + I*B - C)*Sqrt[c + I*d]*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/Sqrt[a + I*b]))/(2*(a^2 + b^
2)*f) - (2*(b^2*((b*c - a*d)*(b^2*d - (3*a*(b*c - a*d))/2)*(a^2*C*d + b^2*(
5*B*c + A*d) + a*b*(5*A*c - 5*c*C - B*d)) + ((-3*b*c)/2 + (a*d)/2)*(a*(4*A*
b^2 - 4*a*b*B - a^2*C - 5*b^2*C)*d*(b*c - a*d) - 5*b^2*(b*c - a*d)*(A*b*c -
 a*B*c - b*c*C - a*A*d - b*B*d + a*C*d))) - a*((3*b*(b*c - a*d)*(b*(4*A*b^2
 - 4*a*b*B - a^2*C - 5*b^2*C)*d*(b*c - a*d) + 5*a*b*(b*c - a*d)*(A*b*c - a*
B*c - b*c*C - a*A*d - b*B*d + a*C*d) + b*(b*c - a*d)*(a^2*C*d + b^2*(5*B*c
+ A*d) + a*b*(5*A*c - 5*c*C - B*d))))/2 - a*d*(b^2*(b*c - a*d)*(a^2*C*d + b
^2*(5*B*c + A*d) + a*b*(5*A*c - 5*c*C - B*d)) - a*(a*(4*A*b^2 - 4*a*b*B - a
^2*C - 5*b^2*C)*d*(b*c - a*d) - 5*b^2*(b*c - a*d)*(A*b*c - a*B*c - b*c*C -
a*A*d - b*B*d + a*C*d)))))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*
```

```
d)*f*Sqrt[a + b*Tan[e + f*x]])))/(3*(a^2 + b^2)*(b*c - a*d))))/(5*(a^2 + b^
2)*(b*c - a*d)))/(2*b)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c+d\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{7}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(7/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{c + d\tan\left(e + fx\right)}\left(A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)\right)}{\left(a + b\tan\left(e + fx\right)\right)^{\frac{7}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(7/2),x)
```

```
[Out] Integral(sqrt(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(a + b*tan(e + f*x))**(7/2), x)
```

3.135 $\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e\right.$

Optimal. Leaf size=682

$$\frac{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}\left(64bd^3\left(a^2B+2ab(A-C)-b^2B\right)+(bc-ad)\left(48bd^2(aB+Ab-bC)+(\right.\right.}{64b^2d^2f}$$

```
[Out] -(a-I*b)^(3/2)*(B+I*(A-C))*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x
+e))^(1/2)/(a-I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/f-(a+I*b)^(3/2)*(B-I*(A-C)
)*(c+I*d)^(3/2)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/
(c+d*tan(f*x+e))^(1/2))/f+1/64*(3*a^4*C*d^4-4*a^3*b*d^3*(2*B*d+3*C*c)+6*a^2
*b^2*d^2*(3*c^2*C+12*B*c*d+8*(A-C)*d^2)-12*a*b^3*d*(c^3*C-6*B*c^2*d-24*c*(A
-C)*d^2+16*B*d^3)+b^4*(3*c^4*C-8*B*c^3*d+48*c^2*(A-C)*d^2-192*B*c*d^3-128*(
A-C)*d^4))*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^
(1/2))/b^(5/2)/d^(5/2)/f+1/64*(64*b*(a^2*B-b^2*B+2*a*b*(A-C))*d^3+(-a*d+b*c
)*(48*b*(A*b+B*a-C*b)*d^2+(-a*d+b*c)*(-8*B*b*d-3*C*a*d+3*C*b*c)))*(a+b*tan(
f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^2/d^2/f+1/96*(48*b*(A*b+B*a-C*b)*d^2
+(-a*d+b*c)*(-8*B*b*d-3*C*a*d+3*C*b*c))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x
+e))^(3/2)/b/d^2/f-1/24*(-8*B*b*d-3*C*a*d+3*C*b*c)*(a+b*tan(f*x+e))^(1/2)*(
c+d*tan(f*x+e))^(5/2)/d^2/f+1/4*C*(a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(
5/2)/d/f
```

Rubi [A] time = 11.90, antiderivative size = 682, normalized size of antiderivative = 1.00, number of steps used = 16, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$\frac{\left(6a^2b^2d^2\left(8d^2(A-C)+12Bcd+3c^2C\right)-4a^3bd^3(2Bd+3cC)+3a^4Cd^4-12ab^3d\left(-24cd^2(A-C)-6Bc^2d+16\right.\right.}{64b^{5/2}d^{5/}}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*
x] + C*Tan[e + f*x]^2),x]
```

```
[Out] -(((a - I*b)^(3/2)*(B + I*(A - C))*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*S
qrt[a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) - ((
a + I*b)^(3/2)*(B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[
a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f + ((3*a^4
*C*d^4 - 4*a^3*b*d^3*(3*c*C + 2*B*d) + 6*a^2*b^2*d^2*(3*c^2*C + 12*B*c*d +
8*(A - C)*d^2) - 12*a*b^3*d*(c^3*C - 6*B*c^2*d - 24*c*(A - C)*d^2 + 16*B*d^
3) + b^4*(3*c^4*C - 8*B*c^3*d + 48*c^2*(A - C)*d^2 - 192*B*c*d^3 - 128*(A -
 C)*d^4))*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Ta
```

```
n[e + f*x]])])/(64*b^(5/2)*d^(5/2)*f) + ((64*b*(a^2*B - b^2*B + 2*a*b*(A -
C))*d^3 + (b*c - a*d)*(48*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(3*b*c*C -
8*b*B*d - 3*a*C*d)))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(64
*b^2*d^2*f) + ((48*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(3*b*c*C - 8*b*B*d
 - 3*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(96*b*d^2
*f) - ((3*b*c*C - 8*b*B*d - 3*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e
+ f*x])^(5/2))/(24*d^2*f) + (C*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*
x])^(5/2))/(4*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
```

```
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int (a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx=\frac{C(a+b\tan(e+fx))^{3/2}(}{4dj}$$

$$=-\frac{(3bcC-8bBd-3aCd}{}$$

$$=\frac{\left(48b(Ab+aB-bC)d^2\right.}{}$$

$$=\frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$$

$$=\frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$$

$$=\frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$$

$$=\frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$$

$$=\frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$$

$$=\frac{\left(64b\left(a^2B-b^2B+2ab(\right.\right.}{}$$

$$=\frac{\left(3a^4Cd^4-4a^3bd^3(3cC\right.}{}$$

$$=-\frac{(a-ib)^{3/2}(B+i(A-C}{}$$

Mathematica [A] time = 9.14, size = 1304, normalized size = 1.91

$$\frac{C(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}{4df}+\frac{\frac{(-3bcC+3adC+8bBd)\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{5/2}}{6df}+\frac{\left(48b(Ab-Cb+aB)d^2+(bc-ad)\right.}{}}{}$$

Warning: Unable to verify antiderivative.

```
[In] Integrate[(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[
e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(3/2)*(c + d*Tan[e + f*x])^(5/2))/(4*d*f) + (((-3*b
*c*C + 8*b*B*d + 3*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/
2))/(6*d*f) + (((48*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(3*b*c*C - 8*b*B*
d - 3*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(8*b*f)
+ (((24*b*(a^2*B - b^2*B + 2*a*b*(A - C))*d^3 - (3*(-(b*c) + a*d)*(48*b*(A*
b + a*B - b*C)*d^2 + (b*c - a*d)*(3*b*c*C - 8*b*B*d - 3*a*C*d)))/8)*Sqrt[a
+ b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(b*f) + ((24*(-(b^4*Sqrt[-b^2]*
d^2*(a^2*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^2*(c^2*C + 2*B*c*d -
 C*d^2 - A*(c^2 - d^2)) + 2*a*b*(2*c*(A - C)*d + B*(c^2 - d^2)))) - b^5*d^2
*(2*a*b*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - a^2*(2*c*(A - C)*d + B*
(c^2 - d^2)) + b^2*(2*c*(A - C)*d + B*(c^2 - d^2))))*ArcTanh[(Sqrt[-c + (Sq
rt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*
Tan[e + f*x]])])/(b^2*Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) -
(24*b^2*d^2*(Sqrt[-b^2]*(a^2*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b^
2*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + 2*a*b*(2*c*(A - C)*d + B*(c^2
 - d^2))) - b*(2*a*b*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - a^2*(2*c*(
A - C)*d + B*(c^2 - d^2)) + b^2*(2*c*(A - C)*d + B*(c^2 - d^2))))*ArcTanh[(
Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*
Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b
]) + (3*Sqrt[b]*Sqrt[c - (a*d)/b]*Sqrt[(c/(c - (a*d)/b) - (a*d)/(b*(c - (a*
d)/b)))^(-1)]*Sqrt[c/(c - (a*d)/b) - (a*d)/(b*(c - (a*d)/b))]*(3*a^4*C*d^4
- 4*a^3*b*d^3*(3*c*C + 2*B*d) + 6*a^2*b^2*d^2*(3*c^2*C + 12*B*c*d + 8*(A -
C)*d^2) - 12*a*b^3*d*(c^3*C - 6*B*c^2*d - 24*c*(A - C)*d^2 + 16*B*d^3) + b^
4*(3*c^4*C - 8*B*c^3*d + 48*c^2*(A - C)*d^2 - 192*B*c*d^3 - 128*(A - C)*d^4
))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b]*Sq
rt[c/(c - (a*d)/b) - (a*d)/(b*(c - (a*d)/b))])]*Sqrt[(c + d*Tan[e + f*x])/(
c - (a*d)/b)])/(8*Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b^2*f))/(2*b))/(3*d))
/(4*d)
```

fricas [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [**F(-1)**] time = 180.00, size = 0, normalized size = 0.00

$$\int \left(a+b\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

maxima [**F**] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{\frac{3}{2}}\left(d\tan\left(fx+e\right)+c\right)^{\frac{3}{2}}dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^(3/2
)*(d*tan(f*x + e) + c)^(3/2), x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \left(a+b\tan\left(e+f\,x\right)\right)^{3/2}\,\left(c+d\tan\left(e+f\,x\right)\right)^{3/2}\,\left(C\tan\left(e+f\,x\right)^2+B\tan\left(e+f\,x\right)+A\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^(3/2)*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^(3/2)*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a+b\tan\left(e+fx\right)\right)^{\frac{3}{2}}\left(c+d\tan\left(e+fx\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(3/2)*(c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(3/2)*(c + d*tan(e + f*x))**(3/2)*(A + B*tan
(e + f*x) + C*tan(e + f*x)**2), x)
```

3.136 $\int \sqrt{a + b\tan(e + fx)}\,(c+d\tan(e+fx))^{3/2}\left(A + B\tan(e\right.$

Optimal. Leaf size=508

$$\frac{\left(a^3Cd^3 - a^2bd^2(2Bd + 3cC) + ab^2d\left(8d^2(A - C) + 12Bcd + 3c^2C\right) - \left(b^3\left(-24cd^2(A - C) - 6Bc^2d + 16Bd^3 + c^3C\right)\right.\right.}{8b^{5/2}d^{3/2}f}$$

```
[Out] 1/8*(a^3*C*d^3-a^2*b*d^2*(2*B*d+3*C*c)+a*b^2*d*(3*c^2*C+12*B*c*d+8*(A-C)*d^
2)-b^3*(c^3*C-6*B*c^2*d-24*c*(A-C)*d^2+16*B*d^3))*arctanh(d^(1/2)*(a+b*tan(
f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(5/2)/d^(3/2)/f-(I*A+B-I*C)
*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(
c+d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)/f-(B-I*(A-C))*(c+I*d)^(3/2)*arctanh((c
+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a
+I*b)^(1/2)/f+1/8*(8*b*(A*b+B*a-C*b)*d^2-(-a*d+b*c)*(-6*B*b*d-C*a*d+C*b*c))
*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^2/d/f-1/12*(-6*B*b*d-C*a*d
+C*b*c)*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)/b/d/f+1/3*C*(a+b*tan(
f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)/d/f
```

Rubi [A] time = 7.49, antiderivative size = 508, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$\frac{\left(-a^2bd^2(2Bd + 3cC) + a^3Cd^3 + ab^2d\left(8d^2(A - C) + 12Bcd + 3c^2C\right) + b^3\left(-\left(-24cd^2(A - C) - 6Bc^2d + 16Bd^3 + \right.\right.\right.}{8b^{5/2}d^{3/2}f}$$

Antiderivative was successfully verified.

```
[In] Int[Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*Sqr
t[a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) - (Sqr
t[a + I*b]*(B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f + ((a^3*C*d^3
 - a^2*b*d^2*(3*c*C + 2*B*d) + a*b^2*d*(3*c^2*C + 12*B*c*d + 8*(A - C)*d^2)
 - b^3*(c^3*C - 6*B*c^2*d - 24*c*(A - C)*d^2 + 16*B*d^3))*ArcTanh[(Sqrt[d]*
Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(8*b^(5/2)*d
^(3/2)*f) + ((8*b*(A*b + a*B - b*C)*d^2 - (b*c - a*d)*(b*c*C - 6*b*B*d - a*
C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(8*b^2*d*f) - ((b*
c*C - 6*b*B*d - a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))
/(12*b*d*f) + (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2))/(3*d*
f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{C\sqrt{a+b\tan(e+fx)}\,(c}{3df}$$

$$= -\frac{(bcC-6bBd-aCd)\sqrt{}}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2-\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2-\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2-\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2-\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2-\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2-\right.}{}$$

$$= \frac{\left(a^3Cd^3-a^2bd^2(3cC+\right.}{}$$

$$= -\frac{\sqrt{a-ib}\,(iA+B-iC)}{}$$

Mathematica [A] time = 9.00, size = 867, normalized size = 1.71

$$\frac{C\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{5/2}}{3df}+\frac{\frac{(-bcC+adC+6bBd)\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{3/2}}{4bf}+\frac{3\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}\,(8}{}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e
+ f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2))/(3*d*f) + (((-(b*c*
C) + 6*b*B*d + a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/
(4*b*f) + ((3*(8*b*(A*b + a*B - b*C)*d^2 - (b*c - a*d)*(b*c*C - 6*b*B*d - a
*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*b*f) + ((6*b^2
*d*(b*(2*a*A*c*d - 2*a*c*C*d + A*b*(c^2 - d^2) + a*B*(c^2 - d^2) - b*(c^2*C
 + 2*B*c*d - C*d^2)) - Sqrt[-b^2]*(a*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^
2)) + b*(2*c*(A - C)*d + B*(c^2 - d^2))))*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)
/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x
]])])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (6*b^2*d*(b*(2*
a*A*c*d - 2*a*c*C*d + A*b*(c^2 - d^2) + a*B*(c^2 - d^2) - b*(c^2*C + 2*B*c*
d - C*d^2)) + Sqrt[-b^2]*(a*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + b*(
2*c*(A - C)*d + B*(c^2 - d^2))))*ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a
 + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt
[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) + (3*Sqrt[b]*Sqrt[c - (a*d)/b]
*(a^3*C*d^3 - a^2*b*d^2*(3*c*C + 2*B*d) + a*b^2*d*(3*c^2*C + 12*B*c*d + 8*(
A - C)*d^2) - b^3*(c^3*C - 6*B*c^2*d - 24*c*(A - C)*d^2 + 16*B*d^3))*ArcSin
h[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*c
 + b*d*Tan[e + f*x])/(b*c - a*d)])/(4*Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b
^2*f))/(2*b))/(3*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \sqrt{a+b\tan\left(fx+e\right)}\,\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\sqrt{b\tan\left(fx+e\right)+a}\left(d\tan\left(fx+e\right)+c\right)^{\frac{3}{2}}dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(b*tan(f*x + e) + a)*
(d*tan(f*x + e) + c)^(3/2), x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \sqrt{a+b\tan\left(e+f\,x\right)}\,{\left(c+d\tan\left(e+f\,x\right)\right)}^{3/2}\,\left(C\tan\left(e+f\,x\right)^{2}+B\tan\left(e+f\,x\right)+A\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^(1/2)*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^(1/2)*(c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \sqrt{a+b\tan\left(e+fx\right)}\left(c+d\tan\left(e+fx\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
```

```
[Out] Integral(sqrt(a + b*tan(e + f*x))*(c + d*tan(e + f*x))**(3/2)*(A + B*tan(e
+ f*x) + C*tan(e + f*x)**2), x)
```

3.137 $\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx$

Optimal. Leaf size=384

$$\frac{\left(3a^2Cd^2-2abd(2Bd+3cC)+b^2\left(8d^2(A-C)+12Bcd+3c^2C\right)\right)\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{5/2}\sqrt{d}f}-\frac{(c-id)^{3/2}(iA+B\cdots}{\cdots}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/f/(a-I*b)^(1/2)+(I*A-B-I*C)*(c+I*d)^(3/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/f/(a+I*b)^(1/2)+1/4*(3*a^2*C*d^2-2*a*b*d*(2*B*d+3*C*c)+b^2*(3*c^2
*C+12*B*c*d+8*(A-C)*d^2))*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c
+d*tan(f*x+e))^(1/2))/b^(5/2)/f/d^(1/2)+1/4*(4*B*b*d-3*C*a*d+3*C*b*c)*(a+b*
tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^2/f+1/2*C*(a+b*tan(f*x+e))^(1/2)
*(c+d*tan(f*x+e))^(3/2)/b/f
```

Rubi [A] time = 4.31, antiderivative size = 384, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$\frac{\left(3a^2Cd^2-2abd(2Bd+3cC)+b^2\left(8d^2(A-C)+12Bcd+3c^2C\right)\right)\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{5/2}\sqrt{d}f}-\frac{(c-id)^{3/2}(iA+B\cdots}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sq
rt[a + b*Tan[e + f*x]],x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a - I*b]*f)) + ((
I*A - B - I*C)*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*
x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a + I*b]*f) + ((3*a^2
*C*d^2 - 2*a*b*d*(3*c*C + 2*B*d) + b^2*(3*c^2*C + 12*B*c*d + 8*(A - C)*d^2)
)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*
x]])]])/(4*b^(5/2)*Sqrt[d]*f) + ((3*b*c*C + 4*b*B*d - 3*a*C*d)*Sqrt[a + b*Ta
n[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*b^2*f) + (C*Sqrt[a + b*Tan[e + f*x
]]*(c + d*Tan[e + f*x])^(3/2))/(2*b*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
```

```
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx = \frac{C\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{3/2}}{2bf}$$

$$= \frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\,\sqrt{}}{4b^2f}$$

$$= \frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\,\sqrt{}}{4b^2f}$$

$$= \frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\,\sqrt{}}{4b^2f}$$

$$= \frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\,\sqrt{}}{4b^2f}$$

$$= \frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\,\sqrt{}}{4b^2f}$$

$$= \frac{(3bcC+4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\,\sqrt{}}{4b^2f}$$

$$= \frac{\left(3a^2Cd^2-2abd(3cC+2Bd)+b^2\left(3c^2C+1\right.\right.}{4b^{5}}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b}}{\sqrt{a-ib}\,\sqrt{c+d}}\right.}{\sqrt{a-ib}\,f}$$

Mathematica [A] time = 7.69, size = 613, normalized size = 1.60

$$\frac{\frac{\sqrt{b}\sqrt{c-\frac{ad}{b}}\left(3a^2Cd^2-2abd(2Bd+3cC)+b^2\left(8d^2(A-C)+12Bcd+3c^2C\right)\right)\sqrt{\frac{bc+bd\tan(e+fx)}{bc-ad}}\,\sinh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c-\frac{ad}{b}}}\right)}{2\sqrt{d}\sqrt{c+d\tan(e+fx)}}-\frac{2b^2\left(\sqrt{-b^2}\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-Cd^2\right)-b\left(2cd(A-C)\right.\right.}{\sqrt{\sqrt{-b^2}-a}\,\sqrt{}}}{b^2f}$$

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/Sqrt[a + b*Tan[e + f*x]],x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(2*b*f) + (((3*b*c*
C + 4*b*B*d - 3*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(
2*b*f) + ((-2*b^2*(Sqrt[-b^2]*(c^2*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) - b
*(2*c*(A - C)*d + B*(c^2 - d^2)))*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)/b]*Sqrt
[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])]])/(S
qrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (2*b^2*(Sqrt[-b^2]*(c^2
*C + 2*B*c*d - C*d^2 - A*(c^2 - d^2)) + b*(2*c*(A - C)*d + B*(c^2 - d^2)))*
ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqr
t[-b^2]]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-
b^2]*d)/b]) + (Sqrt[b]*Sqrt[c - (a*d)/b]*(3*a^2*C*d^2 - 2*a*b*d*(3*c*C + 2*
B*d) + b^2*(3*c^2*C + 12*B*c*d + 8*(A - C)*d^2))*ArcSinh[(Sqrt[d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*c + b*d*Tan[e + f*x])
/(b*c - a*d)])/(2*Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b^2*f))/(2*b)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\sqrt{a+b\tan\left(fx+e\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(d\tan\left(fx+e\right)+c\right)^{\frac{3}{2}}}{\sqrt{b\tan\left(fx+e\right)+a}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)^(3/2
)/sqrt(b*tan(f*x + e) + a), x)
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(1/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(e+fx\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\sqrt{a+b\tan\left(e+fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/sqrt(a + b*tan(e + f*x)), x)
```

3.138 $\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=382

$$-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{bf\left(a^2+b^2\right)\sqrt{a+b\tan(e+fx)}}+\frac{d\left(3a^2C-2abB+2Ab^2+b^2C\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e}{b^2f\left(a^2+b^2\right)}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(3/2)/f-(B-I*(A-C))*(c+I*d)^(3/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(3/2)/f+(2*B*b*d-3*C*a*d+3*C*b*c)*arctanh(d^(1/2)*(a+b*ta
n(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))*d^(1/2)/b^(5/2)/f+(2*A*b^2-
2*B*a*b+3*C*a^2+C*b^2)*d*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^2/
(a^2+b^2)/f-2*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(3/2)/b/(a^2+b^2)/f/(a+b
*tan(f*x+e))^(1/2)
```

Rubi [A] time = 5.74, antiderivative size = 382, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 9, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3645, 3647, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{bf\left(a^2+b^2\right)\sqrt{a+b\tan(e+fx)}}+\frac{d\left(3a^2C-2abB+2Ab^2+b^2C\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e}{b^2f\left(a^2+b^2\right)}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*f)) -
((B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*f) + (Sq
rt[d]*(3*b*c*C + 2*b*B*d - 3*a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(b^(5/2)*f) + ((2*A*b^2 - 2*a*b*B
+ 3*a^2*C + b^2*C)*d*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(b^
2*(a^2 + b^2)*f) - (2*(A*b^2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^(3/2))/(
b*(a^2 + b^2)*f*Sqrt[a + b*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
```

```
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
```

```
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{b\left(a^2+b^2\right)f\sqrt{a+b\tan(e+fx)}}$$

$$= \frac{\left(2Ab^2-2abB+3a^2C+b^2C\right)d\sqrt{a+b\tan(e}}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{\left(2Ab^2-2abB+3a^2C+b^2C\right)d\sqrt{a+b\tan(e}}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{\left(2Ab^2-2abB+3a^2C+b^2C\right)d\sqrt{a+b\tan(e}}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{\left(2Ab^2-2abB+3a^2C+b^2C\right)d\sqrt{a+b\tan(e}}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{\left(2Ab^2-2abB+3a^2C+b^2C\right)d\sqrt{a+b\tan(e}}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{\left(2Ab^2-2abB+3a^2C+b^2C\right)d\sqrt{a+b\tan(e}}{b^2\left(a^2+b^2\right)f}$$

$$= \frac{\sqrt{d}\,(3bcC+2bBd-3aCd)\tanh^{-1}\left(\frac{\sqrt{d}\,\sqrt{a+b\,\mathrm{ta}}}{\sqrt{b}\,\sqrt{c+d\,\mathrm{ta}}}\right.}{b^{5/2}f}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b}}{\sqrt{a-ib}\,\sqrt{c+d}}\right.}{(a-ib)^{3/2}f}$$

Mathematica [C] time = 39.71, size = 1073629, normalized size = 2810.55

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(3/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c + d\tan\left(fx+e\right)\right)^{\frac{3}{2}} \left(A + B\tan\left(fx+e\right) + C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(a + b\tan\left(fx+e\right)\right)^{\frac{3}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(3/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}\left(A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)\right)}{\left(a + b\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(3/2), x)
```

3.139 $\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=402

$$-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{3bf\left(a^2+b^2\right)(a+b\tan(e+fx))^{3/2}}-\frac{2\sqrt{c+d\tan(e+fx)}\,\left(a^4Cd-a^2b^2(d(A-3C)+Bc)+2ab^3(Ac-\right.}{b^2f\left(a^2+b^2\right)^2\sqrt{a+b\tan(e+fx)}}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(5/2)/f-(B-I*(A-C))*(c+I*d)^(3/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(5/2)/f+2*C*d^(3/2)*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2
)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(5/2)/f-2*(a^4*C*d+b^4*(A*d+B*c)+2*a*b^
3*(A*c-B*d-C*c)-a^2*b^2*(B*c+(A-3*C)*d))*(c+d*tan(f*x+e))^(1/2)/b^2/(a^2+b^
2)^2/f/(a+b*tan(f*x+e))^(1/2)-2/3*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(3/2
)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] time = 7.13, antiderivative size = 402, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3645, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{3bf\left(a^2+b^2\right)(a+b\tan(e+fx))^{3/2}}-\frac{2\sqrt{c+d\tan(e+fx)}\,\left(-a^2b^2(d(A-3C)+Bc)+a^4Cd+2ab^3(Ac\right.}{b^2f\left(a^2+b^2\right)^2\sqrt{a+b\tan(e+fx)}}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(5/2)*f)) -
((B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(5/2)*f) + (2*
C*d^(3/2)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Ta
n[e + f*x]])])/(b^(5/2)*f) - (2*(a^4*C*d + b^4*(B*c + A*d) + 2*a*b^3*(A*c -
 c*C - B*d) - a^2*b^2*(B*c + (A - 3*C)*d))*Sqrt[c + d*Tan[e + f*x]])/(b^2*(
a^2 + b^2)^2*f*Sqrt[a + b*Tan[e + f*x]]) - (2*(A*b^2 - a*(b*B - a*C))*(c +
d*Tan[e + f*x])^(3/2))/(3*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^(3/2))
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
```

```
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-\right.}{b^2\left(a^2+b^2\right)^2 f}$$

$$= -\frac{2\left(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-\right.}{b^2\left(a^2+b^2\right)^2 f}$$

$$= -\frac{2\left(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-\right.}{b^2\left(a^2+b^2\right)^2 f}$$

$$= -\frac{2\left(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-\right.}{b^2\left(a^2+b^2\right)^2 f}$$

$$= -\frac{2\left(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-\right.}{b^2\left(a^2+b^2\right)^2 f}$$

$$= -\frac{2\left(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-\right.}{b^2\left(a^2+b^2\right)^2 f}$$

$$= \frac{2Cd^{3/2}\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{5/2}f} - \frac{2\left(a^4Cd\right.}{}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b}}{\sqrt{a-ib}\sqrt{c+d}}\right.}{(a-ib)^{5/2}f}$$

Mathematica [C] time = 41.05, size = 1347065, normalized size = 3350.91

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(5/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(5/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}\left(A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)\right)}{\left(a + b\tan\left(e + fx\right)\right)^{\frac{5}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(5/2), x)
```

3.140 $\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx$

Optimal. Leaf size=586

$$-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{5bf\left(a^2+b^2\right)(a+b\tan(e+fx))^{5/2}}-\frac{2\sqrt{c+d\tan(e+fx)}\,\left(3a^4Cd+2a^3bBd-a^2b^2(7Ad+5Bc-13C\ldots}{15b^2f\left(a^2+b^2\right)^2(a+b\tan(\ldots}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(3/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(7/2)/f-(B-I*(A-C))*(c+I*d)^(3/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(7/2)/f-2/15*(2*a^5*b*B*d^2+3*a^6*C*d^2+a^4*b^2*d*(10*B*c
+(8*A+C)*d)+a^2*b^4*(45*A*c^2-49*A*d^2-90*B*c*d-45*C*c^2+58*C*d^2)-a^3*b^3*
(50*c*(A-C)*d+B*(15*c^2-39*d^2))+a*b^5*(70*c*(A-C)*d+B*(45*c^2-23*d^2))+b^6
*(5*c*(4*B*d+3*C*c)-3*A*(5*c^2-d^2)))*(c+d*tan(f*x+e))^(1/2)/b^2/(a^2+b^2)^
3/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(1/2)-2/15*(2*a^3*b*B*d+3*a^4*C*d+b^4*(3*A*
d+5*B*c)+2*a*b^3*(5*A*c-4*B*d-5*C*c)-a^2*b^2*(7*A*d+5*B*c-13*C*d))*(c+d*tan
(f*x+e))^(1/2)/b^2/(a^2+b^2)^2/f/(a+b*tan(f*x+e))^(3/2)-2/5*(A*b^2-a*(B*b-C
*a))*(c+d*tan(f*x+e))^(3/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(5/2)
```

Rubi [A] time = 3.67, antiderivative size = 586, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 6, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.122, Rules used = {3645, 3649, 3616, 3615, 93, 208}

$$-\frac{2\sqrt{c+d\tan(e+fx)}\,\left(-a^3b^3\left(50cd(A-C)+B\left(15c^2-39d^2\right)\right)+a^2b^4\left(45Ac^2-49Ad^2-90Bcd-45c^2C+58C\ldots}{15b^2f\left(a^2+\ldots}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(7/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(3/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(7/2)*f)) -
((B - I*(A - C))*(c + I*d)^(3/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(7/2)*f) - (2*
(2*a^3*b*B*d + 3*a^4*C*d + b^4*(5*B*c + 3*A*d) + 2*a*b^3*(5*A*c - 5*c*C - 4
*B*d) - a^2*b^2*(5*B*c + 7*A*d - 13*C*d))*Sqrt[c + d*Tan[e + f*x]])/(15*b^2
*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x])^(3/2)) - (2*(2*a^5*b*B*d^2 + 3*a^6*C*
d^2 + a^4*b^2*d*(10*B*c + (8*A + C)*d) + a^2*b^4*(45*A*A*c^2 - 45*c^2*C - 90*
B*c*d - 49*A*d^2 + 58*C*d^2) - a^3*b^3*(50*c*(A - C)*d + B*(15*c^2 - 39*d^2
```

```
)) + a*b^5*(70*c*(A - C)*d + B*(45*c^2 - 23*d^2)) + b^6*(5*c*(3*c*C + 4*B*d
) - 3*A*(5*c^2 - d^2)))*Sqrt[c + d*Tan[e + f*x]])/(15*b^2*(a^2 + b^2)^3*(b*
c - a*d)*f*Sqrt[a + b*Tan[e + f*x]]) - (2*(A*b^2 - a*(b*B - a*C))*(c + d*Ta
n[e + f*x])^(3/2))/(5*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^(5/2))
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
```

```
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{3/2}}{5b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{5/2}}$$

$$= -\frac{2\left(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab\right.}{15b^2\left(\right.}$$

$$= -\frac{2\left(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab\right.}{15b^2\left(\right.}$$

$$= -\frac{2\left(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab\right.}{15b^2\left(\right.}$$

$$= -\frac{2\left(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab\right.}{15b^2\left(\right.}$$

$$= -\frac{2\left(2a^3bBd+3a^4Cd+b^4(5Bc+3Ad)+2ab\right.}{15b^2\left(\right.}$$

$$= -\frac{(iA+B-iC)(c-id)^{3/2}\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b}}{\sqrt{a-ib}\sqrt{c+d}}\right.}{(a-ib)^{7/2}f}$$

Mathematica [B] time = 9.06, size = 3134, normalized size = 5.35

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(7/2),x]
```

```
[Out] -((C*(c + d*Tan[e + f*x])^(3/2))/(b*f*(a + b*Tan[e + f*x])^(5/2))) - (-1/4*
((3*b*c*C - 2*b*B*d - 3*a*C*d)*Sqrt[c + d*Tan[e + f*x]])/(b*f*(a + b*Tan[e
+ f*x])^(5/2)) - ((-2*((b^2*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B
*d) - 5*b^2*c*(c*C + 2*B*d)))/4 - a*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (
b*c - a*d)*(3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2
```

```
- d^2))))*Sqrt[c + d*Tan[e + f*x]])/(5*(a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan
[e + f*x])^(5/2)) - (2*((-2*(b^2*(((2*b^2*d - (5*a*(b*c - a*d))/2)*(8*A*b^2
*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B*d)))/4 + ((
-5*b*c)/2 + (a*d)/2)*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(3*b
*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)))) - a*(
(5*b*(b*c - a*d)*((b*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5
*b^2*c*(c*C + 2*B*d)))/4 - (b*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(3*b
*c*C - 2*b*B*d - 3*a*C*d)))/4 - 2*a*b^2*(2*c*(A - C)*d + B*(c^2 - d^2))))/2
 - 2*a*d*((b^2*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c
*(c*C + 2*B*d)))/4 - a*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(3
*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 - d^2))))))*S
qrt[c + d*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])^
(3/2)) - (2*((-15*b^2*(b*c - a*d)^2*(((3*a^2*A*b*c^2 - A*b^3*c^2 - a^3*B*c^
2 + 3*a*b^2*B*c^2 - 3*a^2*b*c^2*C + b^3*c^2*C - 2*a^3*A*c*d + 6*a*A*b^2*c*d
 - 6*a^2*b*B*c*d + 2*b^3*B*c*d + 2*a^3*c*C*d - 6*a*b^2*c*C*d - 3*a^2*A*b*d^
2 + A*b^3*d^2 + a^3*B*d^2 - 3*a*b^2*B*d^2 + 3*a^2*b*C*d^2 - b^3*C*d^2 + I*(
-(a^3*A*c^2) + 3*a*A*b^2*c^2 - 3*a^2*b*B*c^2 + b^3*B*c^2 + a^3*c^2*C - 3*a*
b^2*c^2*C - 6*a^2*A*b*c*d + 2*A*b^3*c*d + 2*a^3*B*c*d - 6*a*b^2*B*c*d + 6*a
^2*b*c*C*d - 2*b^3*c*C*d + a^3*A*d^2 - 3*a*A*b^2*d^2 + 3*a^2*b*B*d^2 - b^3*
B*d^2 - a^3*C*d^2 + 3*a*b^2*C*d^2))*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[
e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*Sqrt
[-c + I*d]) + ((-3*a^2*A*b*c^2 + A*b^3*c^2 + a^3*B*c^2 - 3*a*b^2*B*c^2 + 3*
a^2*b*c^2*C - b^3*c^2*C + 2*a^3*A*c*d - 6*a*A*b^2*c*d + 6*a^2*b*B*c*d - 2*b
^3*B*c*d - 2*a^3*c*C*d + 6*a*b^2*c*C*d + 3*a^2*A*b*d^2 - A*b^3*d^2 - a^3*B*
d^2 + 3*a*b^2*B*d^2 - 3*a^2*b*C*d^2 + b^3*C*d^2 + I*(-(a^3*A*c^2) + 3*a*A*b
^2*c^2 - 3*a^2*b*B*c^2 + b^3*B*c^2 + a^3*c^2*C - 3*a*b^2*c^2*C - 6*a^2*A*b*
c*d + 2*A*b^3*c*d + 2*a^3*B*c*d - 6*a*b^2*B*c*d + 6*a^2*b*c*C*d - 2*b^3*c*C
*d + a^3*A*d^2 - 3*a*A*b^2*d^2 + 3*a^2*b*B*d^2 - b^3*B*d^2 - a^3*C*d^2 + 3*
a*b^2*C*d^2))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*
b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*Sqrt[c + I*d])))/(2*(a^2 + b^
2)*f) - (2*(b^2*((b^2*d - (3*a*(b*c - a*d))/2)*(((2*b^2*d - (5*a*(b*c - a*d
))/2)*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2
*B*d)))/4 + ((-5*b*c)/2 + (a*d)/2)*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b
*c - a*d)*(3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 -
 d^2)))) + ((-3*b*c)/2 + (a*d)/2)*((5*b*(b*c - a*d)*((b*(8*A*b^2*c^2 + 3*a^
2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B*d)))/4 - (b*(8*b^2*d*(
B*c + (A - C)*d) + (b*c - a*d)*(3*b*c*C - 2*b*B*d - 3*a*C*d)))/4 - 2*a*b^2*
(2*c*(A - C)*d + B*(c^2 - d^2))))/2 - 2*a*d*((b^2*(8*A*b^2*c^2 + 3*a^2*C*d^
2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B*d)))/4 - a*(-1/4*(a*(8*b^2*d
*(B*c + (A - C)*d) + (b*c - a*d)*(3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2
*c*(A - C)*d + B*(c^2 - d^2)))))) - a*((3*b*(b*c - a*d)*((-5*a*(b*c - a*d)*
((b*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B
*d)))/4 - (b*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(3*b*c*C - 2*b*B*d -
3*a*C*d)))/4 - 2*a*b^2*(2*c*(A - C)*d + B*(c^2 - d^2))))/2 - 2*b*d*((b^2*(8
*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B*d)))/
```

```
4 - a*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(3*b*c*C - 2*b*B*d
- 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)))) + b*(((2*b^2*d - (5*
a*(b*c - a*d))/2)*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^
2*c*(c*C + 2*B*d)))/4 + ((-5*b*c)/2 + (a*d)/2)*(-1/4*(a*(8*b^2*d*(B*c + (A
- C)*d) + (b*c - a*d)*(3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*
d + B*(c^2 - d^2))))))/2 - a*d*(b^2*(((2*b^2*d - (5*a*(b*c - a*d))/2)*(8*A*
b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^2*c*(c*C + 2*B*d)))/4 +
 ((-5*b*c)/2 + (a*d)/2)*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(
3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)))) -
a*((5*b*(b*c - a*d)*((b*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d)
- 5*b^2*c*(c*C + 2*B*d)))/4 - (b*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(
3*b*c*C - 2*b*B*d - 3*a*C*d)))/4 - 2*a*b^2*(2*c*(A - C)*d + B*(c^2 - d^2)))
)/2 - 2*a*d*((b^2*(8*A*b^2*c^2 + 3*a^2*C*d^2 - 2*a*b*d*(3*c*C - B*d) - 5*b^
2*c*(c*C + 2*B*d)))/4 - a*(-1/4*(a*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)
*(3*b*c*C - 2*b*B*d - 3*a*C*d))) + 2*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)))))
)))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e +
 f*x]])))/(3*(a^2 + b^2)*(b*c - a*d))))/(5*(a^2 + b^2)*(b*c - a*d)))/(2*b))
/b
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{7}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(((2*b*d+2*a*c)^2>0)', see `assume?`
 for more details)Is ((2*b*d+2*a*c)^2    -4*((a*c-b*d)^2       -((-a*d)-b*c
)        *(a*d+b*c)))    ^2 positive or zero?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(7/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}\left(A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)\right)}{\left(a + b\tan\left(e + fx\right)\right)^{\frac{7}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(7/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(7/2), x)
```

3.141 $\int \sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e\right.$

Optimal. Leaf size=697

$$-\frac{\left(5a^4Cd^4-4a^3bd^3(2Bd+5cC)+2a^2b^2d^2\left(8d^2(A-C)+20Bcd+15c^2C\right)-4ab^3d\left(40cd^2(A-C)+30Bc^2d-16\right.\right.}{64b^{7/2}}$$

```
[Out] -1/64*(5*a^4*C*d^4-4*a^3*b*d^3*(2*B*d+5*C*c)+2*a^2*b^2*d^2*(15*c^2*C+20*B*c
*d+8*(A-C)*d^2)-4*a*b^3*d*(5*c^3*C+30*B*c^2*d+40*c*(A-C)*d^2-16*B*d^3)+b^4*
(5*c^4*C-40*B*c^3*d-240*c^2*(A-C)*d^2+320*B*c*d^3+128*(A-C)*d^4))*arctanh(d
^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(7/2)/d^(3/
2)/f-(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)
/(a-I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)/f+(I*A-B-I*C)*(c+I*d)^
(5/2)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f
*x+e))^(1/2))*(a+I*b)^(1/2)/f+1/64*(64*b^2*d^2*(A*a*d+A*b*c+B*a*c-B*b*d-C*a
*d-C*b*c)+(-a*d+b*c)*(48*b*(A*b+B*a-C*b)*d^2-5*(-a*d+b*c)*(-8*B*b*d-C*a*d+C
*b*c)))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^3/d/f+1/96*(48*b*(A
*b+B*a-C*b)*d^2-5*(-a*d+b*c)*(-8*B*b*d-C*a*d+C*b*c))*(a+b*tan(f*x+e))^(1/2)
*(c+d*tan(f*x+e))^(3/2)/b^2/d/f-1/24*(-8*B*b*d-C*a*d+C*b*c)*(a+b*tan(f*x+e)
)^(1/2)*(c+d*tan(f*x+e))^(5/2)/b/d/f+1/4*C*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(
f*x+e))^(7/2)/d/f
```

Rubi [A] time = 10.42, antiderivative size = 697, normalized size of antiderivative = 1.00, number of steps used = 16, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{\left(2a^2b^2d^2\left(8d^2(A-C)+20Bcd+15c^2C\right)-4a^3bd^3(2Bd+5cC)+5a^4Cd^4-4ab^3d\left(40cd^2(A-C)+30Bc^2d-16\right.\right.}{64b^{7/2}}$$

Antiderivative was successfully verified.

```
[In] Int[Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqr
t[a + b*Tan[e + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/f) + (Sqr
t[a + I*b]*(I*A - B - I*C)*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/f - ((5*a^4*C*d
^4 - 4*a^3*b*d^3*(5*c*C + 2*B*d) + 2*a^2*b^2*d^2*(15*c^2*C + 20*B*c*d + 8*(
A - C)*d^2) - 4*a*b^3*d*(5*c^3*C + 30*B*c^2*d + 40*c*(A - C)*d^2 - 16*B*d^3
) + b^4*(5*c^4*C - 40*B*c^3*d - 240*c^2*(A - C)*d^2 + 320*B*c*d^3 + 128*(A
- C)*d^4))*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*T
```

```
an[e + f*x]])])/(64*b^(7/2)*d^(3/2)*f) + ((64*b^2*d^2*(A*b*c + a*B*c - b*c*
C + a*A*d - b*B*d - a*C*d) + (b*c - a*d)*(48*b*(A*b + a*B - b*C)*d^2 - 5*(b
*c - a*d)*(b*c*C - 8*b*B*d - a*C*d)))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*T
an[e + f*x]])/(64*b^3*d*f) + ((48*b*(A*b + a*B - b*C)*d^2 - 5*(b*c - a*d)*(
b*c*C - 8*b*B*d - a*C*d))*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/
2))/(96*b^2*d*f) - ((b*c*C - 8*b*B*d - a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c +
 d*Tan[e + f*x])^(5/2))/(24*b*d*f) + (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan
[e + f*x])^(7/2))/(4*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
```

```
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{C\sqrt{a+b\tan(e+fx)}\,(c}{4df}$$

$$= -\frac{(bcC-8bBd-aCd)\sqrt{}}{}$$

$$= \frac{\left(48b(Ab+aB-bC)d^2\right.}{}$$

$$= \frac{\left(64b^2d^2(Abc+aBc-b\right.}{}$$

$$= \frac{\left(64b^2d^2(Abc+aBc-b\right.}{}$$

$$= \frac{\left(64b^2d^2(Abc+aBc-b\right.}{}$$

$$= \frac{\left(64b^2d^2(Abc+aBc-b\right.}{}$$

$$= \frac{\left(64b^2d^2(Abc+aBc-b\right.}{}$$

$$= \frac{\left(64b^2d^2(Abc+aBc-b\right.}{}$$

$$= -\frac{\left(5a^4Cd^4-4a^3bd^3(5c\right.}{}$$

$$= -\frac{\sqrt{a-ib}\,(iA+B-iC)}{}$$

Mathematica [A] time = 9.73, size = 1261, normalized size = 1.81

$$\frac{C\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{7/2}}{4df}+\frac{\frac{(-bcC+adC+8bBd)\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{5/2}}{6bf}+\frac{\left(48b(Ab-Cb+aB)d^2-5(bc-ad)(bcC-a\right.}{}}{}$$

Warning: Unable to verify antiderivative.

```
[In] Integrate[Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e
+ f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(7/2))/(4*d*f) + (((-(b*c*
C) + 8*b*B*d + a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2))/
(6*b*f) + (((48*b*(A*b + a*B - b*C)*d^2 - 5*(b*c - a*d)*(b*c*C - 8*b*B*d -
a*C*d))*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))/(8*b*f) + (((2
4*b^2*d^2*(A*b*c + a*B*c - b*c*C + a*A*d - b*B*d - a*C*d) - (3*(-(b*c) + a*
d)*(48*b*(A*b + a*B - b*C)*d^2 - 5*(b*c - a*d)*(b*c*C - 8*b*B*d - a*C*d)))/
8)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(b*f) + ((-24*b^3*d*(
Sqrt[-b^2]*(b*(A - C)*d*(3*c^2 - d^2) + b*B*(c^3 - 3*c*d^2) - a*(A*c^3 - c^
3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3)) - b*(A*(b*c^3 + 3*a*c^2*d
 - 3*b*c*d^2 - a*d^3) - b*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3) + a*(B*c^
3 - 3*c^2*C*d - 3*B*c*d^2 + C*d^3)))*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)/b]*S
qrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])
/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (24*b^3*d*(Sqrt[-b^2
]*(b*(A - C)*d*(3*c^2 - d^2) + b*B*(c^3 - 3*c*d^2) - a*(A*c^3 - c^3*C - 3*B
*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3)) + b*(A*(b*c^3 + 3*a*c^2*d - 3*b*c*
d^2 - a*d^3) - b*(c^3*C + 3*B*c^2*d - 3*c*C*d^2 - B*d^3) + a*(B*c^3 - 3*c^2
*C*d - 3*B*c*d^2 + C*d^3)))*ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*
Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a +
Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) - (3*Sqrt[b]*Sqrt[c - (a*d)/b]*Sqrt
[(c/(c - (a*d)/b) - (a*d)/(b*(c - (a*d)/b)))^(-1)]*Sqrt[c/(c - (a*d)/b) - (
a*d)/(b*(c - (a*d)/b))]*(5*a^4*C*d^4 - 4*a^3*b*d^3*(5*c*C + 2*B*d) + 2*a^2*
b^2*d^2*(15*c^2*C + 20*B*c*d + 8*(A - C)*d^2) - 4*a*b^3*d*(5*c^3*C + 30*B*c
^2*d + 40*c*(A - C)*d^2 - 16*B*d^3) + b^4*(5*c^4*C - 40*B*c^3*d - 240*c^2*(
A - C)*d^2 + 320*B*c*d^3 + 128*(A - C)*d^4))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Ta
n[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b]*Sqrt[c/(c - (a*d)/b) - (a*d)/(b*(c
- (a*d)/b))])]*Sqrt[(c + d*Tan[e + f*x])/(c - (a*d)/b)])/(8*Sqrt[d]*Sqrt[c
+ d*Tan[e + f*x]]))/(b^2*f))/(2*b))/(3*b))/(4*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \sqrt{a+b\tan\left(fx+e\right)}\,\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x
+e)^2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*t
an(f*x+e)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \sqrt{a+b\tan\left(e+f\,x\right)}\,{\left(c+d\tan\left(e+f\,x\right)\right)}^{5/2}\,\left(C\,{\tan\left(e+f\,x\right)}^2+B\tan\left(e+f\,x\right)+A\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^(1/2)*(c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^(1/2)*(c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*
x) + C*tan(e + f*x)^2), x)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C
*tan(f*x+e)**2),x)
```

```
[Out] Timed out
```

3.142 $\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx$

Optimal. Leaf size=505

$$-\frac{\left(5a^3Cd^3-3a^2bd^2(2Bd+5cC)+ab^2d\left(8d^2(A-C)+20Bcd+15c^2C\right)-\left(b^3\left(40cd^2(A-C)+30Bc^2d-16Bd^3\right.\right.\right.}{8b^{7/2}\sqrt{d}\,f}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/f/(a-I*b)^(1/2)-(B-I*(A-C))*(c+I*d)^(5/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/f/(a+I*b)^(1/2)-1/8*(5*a^3*C*d^3-3*a^2*b*d^2*(2*B*d+5*C*c)+a*b^2*
d*(15*c^2*C+20*B*c*d+8*(A-C)*d^2)-b^3*(5*c^3*C+30*B*c^2*d+40*c*(A-C)*d^2-16
*B*d^3))*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1
/2))/b^(7/2)/f/d^(1/2)+1/8*(8*b^2*d*(B*c+(A-C)*d)+(-a*d+b*c)*(6*B*b*d-5*C*a
*d+5*C*b*c))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^3/f+1/12*(6*B*
b*d-5*C*a*d+5*C*b*c)*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)/b^2/f+1/
3*C*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(5/2)/b/f
```

Rubi [A] time = 6.23, antiderivative size = 505, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{\left(-3a^2bd^2(2Bd+5cC)+5a^3Cd^3+ab^2d\left(8d^2(A-C)+20Bcd+15c^2C\right)+b^3\left(-\left(40cd^2(A-C)+30Bc^2d-16\right.\right.\right.}{8b^{7/2}\sqrt{d}\,f}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sq
rt[a + b*Tan[e + f*x]],x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
 + f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a - I*b]*f)) - ((
B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*
x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*f) - ((5*a^3
*C*d^3 - 3*a^2*b*d^2*(5*c*C + 2*B*d) + a*b^2*d*(15*c^2*C + 20*B*c*d + 8*(A
- C)*d^2) - b^3*(5*c^3*C + 30*B*c^2*d + 40*c*(A - C)*d^2 - 16*B*d^3))*ArcTa
nh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/
(8*b^(7/2)*Sqrt[d]*f) + ((8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(5*b*c*C
+ 6*b*B*d - 5*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(8
*b^3*f) + ((5*b*c*C + 6*b*B*d - 5*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Ta
```

n[e + f*x])^(3/2))/(12*b^2*f) + (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2))/(3*b*f)

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
```

```
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{a+b\tan(e+fx)}}\,dx = \frac{C\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{5/2}}{3bf}$$

$$= \frac{(5bcC+6bBd-5aCd)\sqrt{a+b\tan(e+fx)}\,(c}{12b^2f}$$

$$= \frac{\left(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6b\right.}{}$$

$$= \frac{\left(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6b\right.}{}$$

$$= \frac{\left(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6b\right.}{}$$

$$= \frac{\left(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6b\right.}{}$$

$$= \frac{\left(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6b\right.}{}$$

$$= \frac{\left(8b^2d(Bc+(A-C)d)+(bc-ad)(5bcC+6b\right.}{}$$

$$= -\frac{\left(5a^3Cd^3-3a^2bd^2(5cC+2Bd)+ab^2d\left(15c\right.\right.}{}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b}}{\sqrt{a-ib}\,\sqrt{c+d}}\right.}{\sqrt{a-ib}\,f}$$

Mathematica [A] time = 8.97, size = 780, normalized size = 1.54

$$-\frac{3\sqrt{b}\sqrt{c-\frac{ad}{b}}\left(5a^3Cd^3-3a^2bd^2(2Bd+5cC)+ab^2d\left(8d^2(A-C)+20Bcd+15c^2C\right)-\left(b^3\left(40cd^2(A-C)+30Bc^2d-16Bd^3+5c^3C\right)\right)\right)\sqrt{\frac{bc+bd\tan(e+fx)}{bc-ad}}\sinh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c-\frac{ad}{b}}}\right)}{4\sqrt{d}\sqrt{c+d\tan(e+fx)}}+\cdots$$

Antiderivative was successfully verified.

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/Sqrt[a + b*Tan[e + f*x]],x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(5/2))/(3*b*f) + (((5*b*c*
C + 6*b*B*d - 5*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e + f*x])^(3/2))
/(4*b*f) + ((3*(8*b^2*d*(B*c + (A - C)*d) + (b*c - a*d)*(5*b*c*C + 6*b*B*d
- 5*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*b*f) + ((
6*b^3*(b*(A - C)*d*(3*c^2 - d^2) + b*B*(c^3 - 3*c*d^2) + Sqrt[-b^2]*(A*c^3
- c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3))*ArcTanh[(Sqrt[-c + (S
qrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d
*Tan[e + f*x]])]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (6*
b^3*(b*(A - C)*d*(3*c^2 - d^2) + b*B*(c^3 - 3*c*d^2) - Sqrt[-b^2]*(A*c^3 -
c^3*C - 3*B*c^2*d - 3*A*c*d^2 + 3*c*C*d^2 + B*d^3))*ArcTanh[(Sqrt[c + (Sqrt
[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan
[e + f*x]])]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) - (3*Sqrt[b
]*Sqrt[c - (a*d)/b]*(5*a^3*C*d^3 - 3*a^2*b*d^2*(5*c*C + 2*B*d) + a*b^2*d*(1
5*c^2*C + 20*B*c*d + 8*(A - C)*d^2) - b^3*(5*c^3*C + 30*B*c^2*d + 40*c*(A -
 C)*d^2 - 16*B*d^3))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sq
rt[c - (a*d)/b])]*Sqrt[(b*c + b*d*Tan[e + f*x])/(b*c - a*d)])/(4*Sqrt[d]*Sq
rt[c + d*Tan[e + f*x]]))/(b^2*f))/(2*b))/(3*b)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\sqrt{a+b\tan\left(fx+e\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(1/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(1/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(e+fx\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\sqrt{a+b\tan\left(e+fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/sqrt(a + b*tan(e + f*x)), x)
```

3.143 $\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=535

$$-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{bf\left(a^2+b^2\right)\sqrt{a+b\tan(e+fx)}}+\frac{d\left(5a^2C-4abB+4Ab^2+b^2C\right)\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(3/2)/f-(B-I*(A-C))*(c+I*d)^(5/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(3/2)/f+1/4*(15*a^2*C*d^2-6*a*b*d*(2*B*d+5*C*c)+b^2*(15*c
^2*C+20*B*c*d+8*(A-C)*d^2))*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/
(c+d*tan(f*x+e))^(1/2))*d^(1/2)/b^(7/2)/f-1/4*d*(15*a^3*C*d-8*A*b^2*(-a*d+b
*c)-3*a^2*b*(4*B*d+5*C*c)-b^3*(4*B*d+7*C*c)+a*b^2*(8*B*c+7*C*d))*(a+b*tan(f
*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^3/(a^2+b^2)/f+1/2*(4*A*b^2-4*B*a*b+5*
C*a^2+C*b^2)*d*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(3/2)/b^2/(a^2+b^2)/
f-2*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(5/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e
))^(1/2)
```

Rubi [A] time = 8.31, antiderivative size = 535, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 9, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3645, 3647, 3655, 6725, 63, 217, 206, 93, 208}

$$\frac{\sqrt{d}\left(15a^2Cd^2-6abd(2Bd+5cC)+b^2\left(8d^2(A-C)+20Bcd+15c^2C\right)\right)\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4b^{7/2}f}-\frac{2\left(Ab^2-a(bB\right.}{bf\left(a^2+\right.}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*f)) -
((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*f) + (Sq
rt[d]*(15*a^2*C*d^2 - 6*a*b*d*(5*c*C + 2*B*d) + b^2*(15*c^2*C + 20*B*c*d +
8*(A - C)*d^2))*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c
+ d*Tan[e + f*x]])])/(4*b^(7/2)*f) - (d*(15*a^3*C*d - 8*A*b^2*(b*c - a*d) -
 3*a^2*b*(5*c*C + 4*B*d) - b^3*(7*c*C + 4*B*d) + a*b^2*(8*B*c + 7*C*d))*Sqr
t[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*b^3*(a^2 + b^2)*f) + ((4
*A*b^2 - 4*a*b*B + 5*a^2*C + b^2*C)*d*Sqrt[a + b*Tan[e + f*x]]*(c + d*Tan[e
```

```
 + f*x])^(3/2))/(2*b^2*(a^2 + b^2)*f) - (2*(A*b^2 - a*(b*B - a*C))*(c + d*T
an[e + f*x])^(5/2))/(b*(a^2 + b^2)*f*Sqrt[a + b*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
```

```
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{3/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5}}{b\left(a^2+b^2\right)f\sqrt{a+b\tan(e+fx)}}$$

$$= \frac{\left(4Ab^2-4abB+5a^2C+b^2C\right)d\sqrt{a+b\tan}}{2b^2\left(a^2+b^2\right)}$$

$$= -\frac{d\left(15a^3Cd-8Ab^2(bc-ad)-3a^2b(5cC+\right.}{}$$

$$= -\frac{d\left(15a^3Cd-8Ab^2(bc-ad)-3a^2b(5cC+\right.}{}$$

$$= -\frac{d\left(15a^3Cd-8Ab^2(bc-ad)-3a^2b(5cC+\right.}{}$$

$$= -\frac{d\left(15a^3Cd-8Ab^2(bc-ad)-3a^2b(5cC+\right.}{}$$

$$= -\frac{d\left(15a^3Cd-8Ab^2(bc-ad)-3a^2b(5cC+\right.}{}$$

$$= -\frac{d\left(15a^3Cd-8Ab^2(bc-ad)-3a^2b(5cC+\right.}{}$$

$$= \frac{\sqrt{d}\left(15a^2Cd^2-6abd(5cC+2Bd)+b^2\left(15\right.\right.}{}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+}}{\sqrt{a-ib}\,\sqrt{c+}}\right.}{(a-ib)^{3/2}f}$$

Mathematica [C] time = 44.38, size = 1654245, normalized size = 3092.05

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(3/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c + d\tan\left(fx+e\right)\right)^{\frac{5}{2}} \left(A + B\tan\left(fx+e\right) + C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(a + b\tan\left(fx+e\right)\right)^{\frac{3}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(3/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(3/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c + d\tan\left(e + fx\right)\right)^{\frac{5}{2}}\left(A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)\right)}{\left(a + b\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(3/2), x)
```

3.144 $\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=545

$$-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{3bf\left(a^2+b^2\right)(a+b\tan(e+fx))^{3/2}}+\frac{2(c+d\tan(e+fx))^{3/2}\left(-5a^4Cd+2a^3bBd+a^2b^2(d(A-11C)+3B\right.}{3b^2f\left(a^2+b^2\right)^2\sqrt{a+b\tan}}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(5/2)/f-(B-I*(A-C))*(c+I*d)^(5/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(5/2)/f+d^(3/2)*(2*B*b*d-5*C*a*d+5*C*b*c)*arctanh(d^(1/2)
*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(7/2)/f-d*(2*a^3*
b*B*d-5*a^4*C*d-2*a*b^3*(2*A*c-3*B*d-2*C*c)+2*a^2*b^2*(B*c-5*C*d)-b^4*(2*B*
c+(4*A+C)*d))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/b^3/(a^2+b^2)^2
/f+2/3*(2*a^3*b*B*d-5*a^4*C*d-b^4*(5*A*d+3*B*c)-2*a*b^3*(3*A*c-4*B*d-3*C*c)
+a^2*b^2*(3*B*c+(A-11*C)*d))*(c+d*tan(f*x+e))^(3/2)/b^2/(a^2+b^2)^2/f/(a+b*
tan(f*x+e))^(1/2)-2/3*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(5/2)/b/(a^2+b^2
)/f/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] time = 11.07, antiderivative size = 545, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 9, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3645, 3647, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{3bf\left(a^2+b^2\right)(a+b\tan(e+fx))^{3/2}}+\frac{2(c+d\tan(e+fx))^{3/2}\left(a^2b^2(d(A-11C)+3Bc)+2a^3bBd-5a^4C\right.}{3b^2f\left(a^2+b^2\right)^2\sqrt{a+b\tan(}}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(5/2)*f)) -
((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(5/2)*f) + (d^
(3/2)*(5*b*c*C + 2*b*B*d - 5*a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(b^(7/2)*f) - (d*(2*a^3*b*B*d - 5*
a^4*C*d - 2*a*b^3*(2*A*c - 2*c*C - 3*B*d) + 2*a^2*b^2*(B*c - 5*C*d) - b^4*(
2*B*c + (4*A + C)*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(b
^3*(a^2 + b^2)^2*f) + (2*(2*a^3*b*B*d - 5*a^4*C*d - b^4*(3*B*c + 5*A*d) - 2
```

```
*a*b^3*(3*A*c - 3*c*C - 4*B*d) + a^2*b^2*(3*B*c + (A - 11*C)*d))*(c + d*Tan
[e + f*x])^(3/2))/(3*b^2*(a^2 + b^2)^2*f*Sqrt[a + b*Tan[e + f*x]]) - (2*(A*
b^2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^(5/2))/(3*b*(a^2 + b^2)*f*(a + b*
Tan[e + f*x])^(3/2))
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
```

```
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{5/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5}}{3b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{3/2}}$$

$$= \frac{2\left(2a^3bBd-5a^4Cd-b^4(3Bc+5Ad)-2ab\right.}{3b^2}$$

$$= -\frac{d\left(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3\right.}{}$$

$$= -\frac{d\left(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3\right.}{}$$

$$= -\frac{d\left(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3\right.}{}$$

$$= -\frac{d\left(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3\right.}{}$$

$$= -\frac{d\left(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3\right.}{}$$

$$= -\frac{d\left(2a^3bBd-5a^4Cd-2ab^3(2Ac-2cC-3\right.}{}$$

$$= \frac{d^{3/2}(5bcC+2bBd-5aCd)\tanh^{-1}\left(\frac{\sqrt{d}\,\sqrt{a+b}}{\sqrt{b}\,\sqrt{c+d}}\right.}{b^{7/2}f}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+}}{\sqrt{a-ib}\,\sqrt{c+}}\right.}{(a-ib)^{5/2}f}$$

Mathematica [C] time = 47.10, size = 2018669, normalized size = 3703.98

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(5/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c + d \tan \left(f x + e\right)\right)^{\frac{5}{2}} \left(A + B \tan \left(f x + e\right) + C \left(\tan^{2}\left(f x + e\right)\right)\right)}{\left(a + b \tan \left(f x + e\right)\right)^{\frac{5}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(5/2),x)
```

maxima [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [**F(-1)**] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(5/2),x)
```

```
[Out] \text{Hanged}
```

sympy [**F**] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(e+fx\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\left(a+b\tan\left(e+fx\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral((c + d*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(a + b*tan(e + f*x))**(5/2), x)
```

3.145 $\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx$

Optimal. Leaf size=590

$$-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{5bf\left(a^2+b^2\right)(a+b\tan(e+fx))^{5/2}}-\frac{2(c+d\tan(e+fx))^{3/2}\left(a^4Cd-a^2b^2(d(A-3C)+Bc)+2ab^3(Ac-\right.}{3b^2f\left(a^2+b^2\right)^2(a+b\tan(e+fx))^{3/2}}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(7/2)/f-(B-I*(A-C))*(c+I*d)^(5/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(7/2)/f+2*C*d^(5/2)*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2
)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/b^(7/2)/f-2*(a^6*C*d^2+3*a^4*b^2*C*d^2-3*
a^2*b^4*(c^2*C+2*B*c*d-2*C*d^2-A*(c^2-d^2))+b^6*(c*(2*B*d+C*c)-A*(c^2-d^2))
-a^3*b^3*(2*c*(A-C)*d+B*(c^2-d^2))+3*a*b^5*(2*c*(A-C)*d+B*(c^2-d^2)))*(c+d*
tan(f*x+e))^(1/2)/b^3/(a^2+b^2)^3/f/(a+b*tan(f*x+e))^(1/2)-2/3*(a^4*C*d+b^4
*(A*d+B*c)+2*a*b^3*(A*c-B*d-C*c)-a^2*b^2*(B*c+(A-3*C)*d))*(c+d*tan(f*x+e))^
(3/2)/b^2/(a^2+b^2)^2/f/(a+b*tan(f*x+e))^(3/2)-2/5*(A*b^2-a*(B*b-C*a))*(c+d
*tan(f*x+e))^(5/2)/b/(a^2+b^2)/f/(a+b*tan(f*x+e))^(5/2)
```

Rubi [A] time = 14.02, antiderivative size = 590, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3645, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{2\sqrt{c+d\tan(e+fx)}\left(-a^3b^3\left(2cd(A-C)+B\left(c^2-d^2\right)\right)-3a^2b^4\left(-A\left(c^2-d^2\right)+2Bcd+c^2C-2Cd^2\right)+3a^4b^2C\right.}{b^3f\left(a^2+b^2\right)^3\sqrt{a+b\tan(e+fx}}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(7/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(7/2)*f)) -
((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(7/2)*f) + (2*
C*d^(5/2)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Ta
n[e + f*x]])])/(b^(7/2)*f) - (2*(a^6*C*d^2 + 3*a^4*b^2*C*d^2 - 3*a^2*b^4*(c
^2*C + 2*B*c*d - 2*C*d^2 - A*(c^2 - d^2)) + b^6*(c*(c*C + 2*B*d) - A*(c^2 -
 d^2)) - a^3*b^3*(2*c*(A - C)*d + B*(c^2 - d^2)) + 3*a*b^5*(2*c*(A - C)*d +
 B*(c^2 - d^2)))*Sqrt[c + d*Tan[e + f*x]])/(b^3*(a^2 + b^2)^3*f*Sqrt[a + b*
```

```
Tan[e + f*x]]) - (2*(a^4*C*d + b^4*(B*c + A*d) + 2*a*b^3*(A*c - c*C - B*d)
- a^2*b^2*(B*c + (A - 3*C)*d))*(c + d*Tan[e + f*x])^(3/2))/(3*b^2*(a^2 + b^
2)^2*f*(a + b*Tan[e + f*x])^(3/2)) - (2*(A*b^2 - a*(b*B - a*C))*(c + d*Tan[
e + f*x])^(5/2))/(5*b*(a^2 + b^2)*f*(a + b*Tan[e + f*x])^(5/2))
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
```

```
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{7/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5}}{5b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{5/2}}$$

$$= -\frac{2\left(a^4Cd+b^4(Bc+Ad)+2ab^3(Ac-cC-\right.}{3b^2\left(a^2+b^2\right)}$$

$$= -\frac{2\left(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4\left(c^2C+2Bc\right.\right.}{}$$

$$= -\frac{2\left(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4\left(c^2C+2Bc\right.\right.}{}$$

$$= -\frac{2\left(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4\left(c^2C+2Bc\right.\right.}{}$$

$$= -\frac{2\left(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4\left(c^2C+2Bc\right.\right.}{}$$

$$= -\frac{2\left(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4\left(c^2C+2Bc\right.\right.}{}$$

$$= -\frac{2\left(a^6Cd^2+3a^4b^2Cd^2-3a^2b^4\left(c^2C+2Bc\right.\right.}{}$$

$$= \frac{2Cd^{5/2}\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{b^{7/2}f} - \frac{2\left(a^6C\right.}{}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+}}{\sqrt{a-ib}\sqrt{c+}}\right.}{(a-ib)^{7/2}f}$$

Mathematica [C] time = 49.13, size = 2345519, normalized size = 3975.46

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(7/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{7}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(7/2),x)
```

maxima [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(7/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [**F(-1)**] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(7/2),x)
```

```
[Out] \text{Hanged}
```

sympy [**F(-1)**] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(7/2),x)
```

```
[Out] Timed out
```

3.146 $\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{9/2}}\,dx$

Optimal. Leaf size=946

$$-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)(c-id)^{5/2}}{(a-ib)^{9/2}f}-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{7b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{7/2}}-\frac{2\left(5Cda^4+\right.}{}$$

```
[Out] -(I*A+B-I*C)*(c-I*d)^(5/2)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a-I*b)^(9/2)/f-(B-I*(A-C))*(c+I*d)^(5/2
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(a+I*b)^(9/2)/f-2/105*(6*a^7*b*B*d^3+15*a^8*C*d^3+2*a^6*b^2*d^2*(
4*A*d+7*B*c+26*C*d)-2*a*b^7*(210*A*c^3-406*A*c*d^2-525*B*c^2*d+88*B*d^3-210
*C*c^3+406*C*c*d^2)-a^4*b^4*(525*A*c^2*d-311*A*d^3+105*B*c^3-749*B*c*d^2-52
5*C*c^2*d+221*C*d^3)+2*a^2*b^6*(875*A*c^2*d-261*A*d^3+315*B*c^3-812*B*c*d^2
-875*C*c^2*d+291*C*d^3)+2*a^5*b^3*d*(56*c*(A-C)*d+B*(35*c^2-12*d^2))-b^8*(5
*d*(49*A*c^2-3*A*d^2-49*C*c^2)+7*B*(15*c^3-23*c*d^2))-2*a^3*b^5*(210*c^3*C+
700*B*c^2*d-798*c*C*d^2-317*B*d^3-42*A*(5*c^3-19*c*d^2)))*(c+d*tan(f*x+e))^
(1/2)/b^3/(a^2+b^2)^4/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(1/2)-2/105*(6*a^5*b*B*
d^2+15*a^6*C*d^2+a^4*b^2*d*(8*A*d+14*B*c+37*C*d)+3*a^2*b^4*(35*A*c^2-39*A*d
^2-70*B*c*d-35*C*c^2+54*C*d^2)-a^3*b^3*(98*c*(A-C)*d+B*(35*c^2-75*d^2))+a*b
^5*(182*c*(A-C)*d+B*(105*c^2-71*d^2))+b^6*(7*c*(8*B*d+5*C*c)-5*A*(7*c^2-3*d
^2)))*(c+d*tan(f*x+e))^(1/2)/b^3/(a^2+b^2)^3/f/(a+b*tan(f*x+e))^(3/2)-2/35*
(2*a^3*b*B*d+5*a^4*C*d+b^4*(5*A*d+7*B*c)+2*a*b^3*(7*A*c-6*B*d-7*C*c)-a^2*b^
2*(9*A*d+7*B*c-19*C*d))*(c+d*tan(f*x+e))^(3/2)/b^2/(a^2+b^2)^2/f/(a+b*tan(f
*x+e))^(5/2)-2/7*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(5/2)/b/(a^2+b^2)/f/(
a+b*tan(f*x+e))^(7/2)
```

Rubi [A] time = 6.46, antiderivative size = 946, normalized size of antiderivative = 1.00, number of steps used = 11, number of rules used = 6, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}} = 0.122$, Rules used = {3645, 3649, 3616, 3615, 93, 208}

$$-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)(c-id)^{5/2}}{(a-ib)^{9/2}f}-\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5/2}}{7b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{7/2}}-\frac{2\left(5Cda^4+\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(a
 + b*Tan[e + f*x])^(9/2),x]
```

```
[Out] -(((I*A + B - I*C)*(c - I*d)^(5/2)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(9/2)*f)) -
```

```
((B - I*(A - C))*(c + I*d)^(5/2)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(9/2)*f) - (2*
(6*a^5*b*B*d^2 + 15*a^6*C*d^2 + a^4*b^2*d*(14*B*c + 8*A*d + 37*C*d) + 3*a^2
*b^4*(35*A*c^2 - 35*c^2*C - 70*B*c*d - 39*A*d^2 + 54*C*d^2) - a^3*b^3*(98*c
*(A - C)*d + B*(35*c^2 - 75*d^2)) + a*b^5*(182*c*(A - C)*d + B*(105*c^2 - 7
1*d^2)) + b^6*(7*c*(5*c*C + 8*B*d) - 5*A*(7*c^2 - 3*d^2)))*Sqrt[c + d*Tan[e
 + f*x]])/(105*b^3*(a^2 + b^2)^3*f*(a + b*Tan[e + f*x])^(3/2)) - (2*(6*a^7*
b*B*d^3 + 15*a^8*C*d^3 + 2*a^6*b^2*d^2*(7*B*c + 4*A*d + 26*C*d) - 2*a*b^7*(
210*A*c^3 - 210*c^3*C - 525*B*c^2*d - 406*A*c*d^2 + 406*c*C*d^2 + 88*B*d^3)
 - a^4*b^4*(105*B*c^3 + 525*A*c^2*d - 525*c^2*C*d - 749*B*c*d^2 - 311*A*d^3
 + 221*C*d^3) + 2*a^2*b^6*(315*B*c^3 + 875*A*c^2*d - 875*c^2*C*d - 812*B*c*
d^2 - 261*A*d^3 + 291*C*d^3) + 2*a^5*b^3*d*(56*c*(A - C)*d + B*(35*c^2 - 12
*d^2)) - b^8*(5*d*(49*A*c^2 - 49*c^2*C - 3*A*d^2) + 7*B*(15*c^3 - 23*c*d^2)
) - 2*a^3*b^5*(210*c^3*C + 700*B*c^2*d - 798*c*C*d^2 - 317*B*d^3 - 42*A*(5*
c^3 - 19*c*d^2)))*Sqrt[c + d*Tan[e + f*x]])/(105*b^3*(a^2 + b^2)^4*(b*c - a
*d)*f*Sqrt[a + b*Tan[e + f*x]]) - (2*(2*a^3*b*B*d + 5*a^4*C*d + b^4*(7*B*c
+ 5*A*d) + 2*a*b^3*(7*A*c - 7*c*C - 6*B*d) - a^2*b^2*(7*B*c + 9*A*d - 19*C*
d))*(c + d*Tan[e + f*x])^(3/2))/(35*b^2*(a^2 + b^2)^2*f*(a + b*Tan[e + f*x]
)^(5/2)) - (2*(A*b^2 - a*(b*B - a*C))*(c + d*Tan[e + f*x])^(5/2))/(7*b*(a^2
 + b^2)*f*(a + b*Tan[e + f*x])^(7/2))
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
```

```
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{(c+d\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(a+b\tan(e+fx))^{9/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)(c+d\tan(e+fx))^{5}}{7b\left(a^2+b^2\right)f(a+b\tan(e+fx))^{7/2}}$$

$$= -\frac{2\left(2a^3bBd+5a^4Cd+b^4(7Bc+5Ad)+2\right.}{35b}$$

$$= -\frac{2\left(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8\right.}{}$$

$$= -\frac{2\left(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8\right.}{}$$

$$= -\frac{2\left(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8\right.}{}$$

$$= -\frac{2\left(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8\right.}{}$$

$$= -\frac{2\left(6a^5bBd^2+15a^6Cd^2+a^4b^2d(14Bc+8\right.}{}$$

$$= -\frac{(iA+B-iC)(c-id)^{5/2}\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+}}{\sqrt{a-ib}\,\sqrt{c+}}\right.}{(a-ib)^{9/2}f}$$

Mathematica [C] time = 53.64, size = 2719441, normalized size = 2874.67

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[((c + d*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(a + b*Tan[e + f*x])^(9/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(9/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(9/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{9}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(9/2),x)
```

```
[Out] int((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))
^(9/2),x)
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f
*x+e))^(9/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
```

```
elp (example of legal syntax is 'assume(((2*b*d+2*a*c)^2>0)', see `assume?`
 for more details)Is ((2*b*d+2*a*c)^2    -4*((a*c-b*d)^2       -((-a*d)-b*c
)        *(a*d+b*c)))    ^2 positive or zero?
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((c + d*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(a
 + b*tan(e + f*x))^(9/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((c+d*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan
(f*x+e))**(9/2),x)
```

```
[Out] Timed out
```

3.147 $\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=505

$$\frac{\left(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab^2d\left(8d^2(A-C)-4Bcd+3c^2C\right)-\left(b^3\left(8cd^2(A-C)-6Bc^2d+16Bd^3+5c^3C\right.\right.\right.}{8\sqrt{b}\,d^{7/2}f}$$

```
[Out] 1/8*(5*a^3*C*d^3-15*a^2*b*d^2*(-2*B*d+C*c)+5*a*b^2*d*(3*c^2*C-4*B*c*d+8*(A-
C)*d^2)-b^3*(5*c^3*C-6*B*c^2*d+8*c*(A-C)*d^2+16*B*d^3))*arctanh(d^(1/2)*(a+
b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/d^(7/2)/f/b^(1/2)-(a-I*
b)^(5/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(
1/2)/(c+d*tan(f*x+e))^(1/2))/f/(c-I*d)^(1/2)-(a+I*b)^(5/2)*(B-I*(A-C))*arct
anh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/
2))/f/(c+I*d)^(1/2)+1/8*(8*b*(A*b+B*a-C*b)*d^2+(-a*d+b*c)*(-6*B*b*d-5*C*a*d
+5*C*b*c))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/d^3/f-1/12*(-6*B*b
*d-5*C*a*d+5*C*b*c)*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)/d^2/f+1/3
*C*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(5/2)/d/f
```

Rubi [A] time = 5.95, antiderivative size = 505, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$\frac{\left(-15a^2bd^2(cC-2Bd)+5a^3Cd^3+5ab^2d\left(8d^2(A-C)-4Bcd+3c^2C\right)+b^3\left(-\left(8cd^2(A-C)-6Bc^2d+16Bd^3+5\right.\right.\right.}{8\sqrt{b}\,d^{7/2}f}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sq
rt[c + d*Tan[e + f*x]],x]
```

```
[Out] -(((a - I*b)^(5/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c - I*d]*f)) - ((
a + I*b)^(5/2)*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*
x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c + I*d]*f) + ((5*a^3
*C*d^3 - 15*a^2*b*d^2*(c*C - 2*B*d) + 5*a*b^2*d*(3*c^2*C - 4*B*c*d + 8*(A -
 C)*d^2) - b^3*(5*c^3*C - 6*B*c^2*d + 8*c*(A - C)*d^2 + 16*B*d^3))*ArcTanh[
(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(8*
Sqrt[b]*d^(7/2)*f) + ((8*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 6
*b*B*d - 5*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(8*d^
3*f) - ((5*b*c*C - 6*b*B*d - 5*a*C*d)*(a + b*Tan[e + f*x])^(3/2)*Sqrt[c + d
```

```
*Tan[e + f*x]])/(12*d^2*f) + (C*(a + b*Tan[e + f*x])^(5/2)*Sqrt[c + d*Tan[e
 + f*x]])/(3*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
```

```
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{C(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}{3df}$$

$$= -\frac{(5bcC-6bBd-5aCd)(a+b\tan(e+fx))}{12d^2f}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-\right.}{}$$

$$= \frac{\left(8b(Ab+aB-bC)d^2+(bc-ad)(5bcC-\right.}{}$$

$$= \frac{\left(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab^2d\left(3c\right.\right.}{}$$

$$= -\frac{(a-ib)^{5/2}(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+}}{\sqrt{a-ib}\sqrt{c+}}\right.}{\sqrt{c-id}\,f}$$

Mathematica [A] time = 8.50, size = 785, normalized size = 1.55

$$\frac{3\sqrt{b}\sqrt{c-\frac{ad}{b}}\left(5a^3Cd^3-15a^2bd^2(cC-2Bd)+5ab^2d\left(8d^2(A-C)-4Bcd+3c^2C\right)-\left(b^3\left(8cd^2(A-C)-6Bc^2d+16Bd^3+5c^3C\right)\right)\right)\sqrt{\frac{bc+bd\tan(e+fx)}{bc-ad}}\sinh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c-\frac{ad}{b}}}\right)}{4\sqrt{d}\sqrt{c+d\tan(e+fx)}}-\frac{6d^3\left(\sqrt{-b\ldots}\right.}{\ldots}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(5/2)*Sqrt[c + d*Tan[e + f*x]])/(3*d*f) + (((-5*b*c
*C + 6*b*B*d + 5*a*C*d)*(a + b*Tan[e + f*x])^(3/2)*Sqrt[c + d*Tan[e + f*x]]
)/(4*d*f) + ((3*(8*b*(A*b + a*B - b*C)*d^2 + (b*c - a*d)*(5*b*c*C - 6*b*B*d
 - 5*a*C*d))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*d*f) + (
(-6*(Sqrt[-b^2]*(3*a^2*b*B - b^3*B - a^3*(A - C) + 3*a*b^2*(A - C)) - b*(a^
3*B - 3*a*b^2*B + 3*a^2*b*(A - C) - b^3*(A - C)))*d^3*ArcTanh[(Sqrt[-c + (S
qrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d
*Tan[e + f*x]])])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (6*
(Sqrt[-b^2]*(3*a^2*b*B - b^3*B - a^3*(A - C) + 3*a*b^2*(A - C)) + b*(a^3*B
- 3*a*b^2*B + 3*a^2*b*(A - C) - b^3*(A - C)))*d^3*ArcTanh[(Sqrt[c + (Sqrt[-
b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e
 + f*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) + (3*Sqrt[b]*
Sqrt[c - (a*d)/b]*(5*a^3*C*d^3 - 15*a^2*b*d^2*(c*C - 2*B*d) + 5*a*b^2*d*(3*
c^2*C - 4*B*c*d + 8*(A - C)*d^2) - b^3*(5*c^3*C - 6*B*c^2*d + 8*c*(A - C)*d
^2 + 16*B*d^3))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c
- (a*d)/b])]*Sqrt[(b*c + b*d*Tan[e + f*x])/(b*c - a*d)])/(4*Sqrt[d]*Sqrt[c
+ d*Tan[e + f*x]]))/(b*d*f))/(2*d))/(3*d)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(fx+e\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\sqrt{c+d\tan\left(fx+e\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{{\left(a+b\,\tan\left(e+f\,x\right)\right)}^{5/2}\,\left(C\,{\tan\left(e+f\,x\right)}^2+B\,\tan\left(e+f\,x\right)+A\right)}{\sqrt{c+d\,\tan\left(e+f\,x\right)}} \,d x$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(1/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(1/2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a + b\tan\left(e + fx\right)\right)^{\frac{5}{2}} \left(A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)\right)}{\sqrt{c + d\tan\left(e + fx\right)}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/sqrt(c + d*tan(e + f*x)), x)
```

3.148 $\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=383

$$\frac{\left(3a^2Cd^2-6abd(cC-2Bd)+b^2\left(8d^2(A-C)-4Bcd+3c^2C\right)\right)\tanh^{-1}\left(\frac{\sqrt{d}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\,\sqrt{c+d\tan(e+fx)}}\right)}{4\sqrt{b}\,d^{5/2}f}-\frac{(a-ib)^{3/2}(iA+B-i}{}$$

```
[Out] 1/4*(3*a^2*C*d^2-6*a*b*d*(-2*B*d+C*c)+b^2*(3*c^2*C-4*B*c*d+8*(A-C)*d^2))*ar
ctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/d^(5/2
)/f/b^(1/2)-(a-I*b)^(3/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e)
)^(1/2)/(a-I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/f/(c-I*d)^(1/2)+(a+I*b)^(3/2)
*(I*A-B-I*C)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/f/(c+I*d)^(1/2)-1/4*(-4*B*b*d-3*C*a*d+3*C*b*c)*(a+b*ta
n(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/d^2/f+1/2*C*(c+d*tan(f*x+e))^(1/2)*(
a+b*tan(f*x+e))^(3/2)/d/f
```

Rubi [A] time = 4.08, antiderivative size = 383, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$\frac{\left(3a^2Cd^2-6abd(cC-2Bd)+b^2\left(8d^2(A-C)-4Bcd+3c^2C\right)\right)\tanh^{-1}\left(\frac{\sqrt{d}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\,\sqrt{c+d\tan(e+fx)}}\right)}{4\sqrt{b}\,d^{5/2}f}-\frac{(a-ib)^{3/2}(iA+B-i}{}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sq
rt[c + d*Tan[e + f*x]],x]
```

```
[Out] -(((a - I*b)^(3/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c - I*d]*f)) + ((
a + I*b)^(3/2)*(I*A - B - I*C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*
x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c + I*d]*f) + ((3*a^2
*C*d^2 - 6*a*b*d*(c*C - 2*B*d) + b^2*(3*c^2*C - 4*B*c*d + 8*(A - C)*d^2))*A
rcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]]
)])/(4*Sqrt[b]*d^(5/2)*f) - ((3*b*c*C - 4*b*B*d - 3*a*C*d)*Sqrt[a + b*Tan[e
 + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*d^2*f) + (C*(a + b*Tan[e + f*x])^(3/2
)*Sqrt[c + d*Tan[e + f*x]])/(2*d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
```

```
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{C(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}{2df} \cdots$$

$$= -\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\cdots}{4d^2f}$$

$$= -\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\cdots}{4d^2f}$$

$$= -\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\cdots}{4d^2f}$$

$$= -\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\cdots}{4d^2f}$$

$$= -\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\cdots}{4d^2f}$$

$$= -\frac{(3bcC-4bBd-3aCd)\sqrt{a+b\tan(e+fx)}\cdots}{4d^2f}$$

$$= \frac{\left(3a^2Cd^2-6abd(cC-2Bd)+b^2\left(3c^2C-4B\cdots\right.\right.}{4\sqrt{b}\cdots}$$

$$= -\frac{(a-ib)^{3/2}(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\cdots}}{\sqrt{a-ib}\sqrt{c+d\cdots}}\right.}{\sqrt{c-id}\,f}$$

Mathematica [A] time = 7.08, size = 582, normalized size = 1.52

$$\frac{\frac{\sqrt{c-\frac{ad}{b}}\left(3a^2Cd^2+6abd(2Bd-cC)+b^2\left(8d^2(A-C)-4Bcd+3c^2C\right)\right)\sqrt{\frac{b(c+d\tan(e+fx))}{bc-ad}}\sinh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c-\frac{ad}{b}}}\right)}{\sqrt{b}\sqrt{d}\sqrt{c+d\tan(e+fx)}}+\frac{4d^2\left(b\left(a^2B+2ab(A-C)-b^2B\right)-\sqrt{-b^2}\left(a^2\cdots\right.\right.}{b\cdots}}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] ((4*(b*(a^2*B - b^2*B + 2*a*b*(A - C)) - Sqrt[-b^2]*(2*a*b*B + b^2*(A - C)
+ a^2*(-A + C)))*d^2*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])]])/(b*Sqrt[-a + Sq
rt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - (4*(b*(a^2*B - b^2*B + 2*a*b*(A -
C)) + Sqrt[-b^2]*(2*a*b*B + b^2*(A - C) + a^2*(-A + C)))*d^2*ArcTanh[(Sqrt[
c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[
c + d*Tan[e + f*x]])]])/(b*Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b])
+ (-3*b*c*C + 4*b*B*d + 3*a*C*d)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e
+ f*x]] + 2*C*d*(a + b*Tan[e + f*x])^(3/2)*Sqrt[c + d*Tan[e + f*x]] + (Sqrt
[c - (a*d)/b]*(3*a^2*C*d^2 + 6*a*b*d*(-(c*C) + 2*B*d) + b^2*(3*c^2*C - 4*B*
c*d + 8*(A - C)*d^2))*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*S
qrt[c - (a*d)/b])]*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a*d)])/(Sqrt[b]*Sqr
t[d]*Sqrt[c + d*Tan[e + f*x]]))/(4*d^2*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\sqrt{c+d\tan\left(fx+e\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{\frac{3}{2}}}{\sqrt{d\tan\left(fx+e\right)+c}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^(3/2
)/sqrt(d*tan(f*x + e) + c), x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{{\left(a+b\tan\left(e+f\,x\right)\right)}^{3/2}\,\left(C\,{\tan\left(e+f\,x\right)}^2+B\,\tan\left(e+f\,x\right)+A\right)}{\sqrt{c+d\,\tan\left(e+f\,x\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(1/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(1/2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(e+fx\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\sqrt{c+d\tan\left(e+fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/sqrt(c + d*tan(e + f*x)), x)
```

3.149 $\int \frac{\sqrt{a+b\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=290

$$-\frac{\sqrt{a-ib}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{c-id}}+\frac{\sqrt{a+ib}\,(iA-B-iC)\tanh^{-1}\left(\frac{\sqrt{c+id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{c+id}}-\frac{(-aCd}{}$$

```
[Out] -(-2*B*b*d-C*a*d+C*b*c)*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d
*tan(f*x+e))^(1/2))/d^(3/2)/f/b^(1/2)-(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+
b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)/f/(
c-I*d)^(1/2)+(I*A-B-I*C)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*
b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a+I*b)^(1/2)/f/(c+I*d)^(1/2)+C*(a+b*tan(f
*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/d/f
```

Rubi [A] time = 2.55, antiderivative size = 290, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3647, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{\sqrt{a-ib}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{c-id}}+\frac{\sqrt{a+ib}\,(iA-B-iC)\tanh^{-1}\left(\frac{\sqrt{c+id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{c+id}}-\frac{(-aCd}{}$$

Antiderivative was successfully verified.

```
[In] Int[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/Sqrt
[c + d*Tan[e + f*x]],x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c - I*d]*f)) + (Sqr
t[a + I*b]*(I*A - B - I*C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])
/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[c + I*d]*f) - ((b*c*C - 2
*b*B*d - a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c
+ d*Tan[e + f*x]])])/(Sqrt[b]*d^(3/2)*f) + (C*Sqrt[a + b*Tan[e + f*x]]*Sqrt
[c + d*Tan[e + f*x]])/(d*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
```

```
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{\sqrt{a+b\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{\sqrt{c+d\tan(e+fx)}}\,dx = \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{\int}{}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{\text{Su}}{}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{\text{Su}}{}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{\text{Su}}{}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{\text{Su}}{}$$

$$= \frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df} + \frac{((}{}$$

$$= -\frac{(bcC-2bBd-aCd)\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right.}{\sqrt{b}\,d^{3/2}f}$$

$$= -\frac{\sqrt{a-ib}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(}}{\sqrt{a-ib}\sqrt{c+d\tan(}}\right.}{\sqrt{c-id}\,f}$$

Mathematica [A] time = 6.63, size = 450, normalized size = 1.55

$$\frac{C\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{df}-\frac{\frac{d\left(\sqrt{-b^2}\,(a(C-A)+bB)-b(aB+Ab-bC)\right)\tanh^{-1}\left(\frac{\sqrt{\frac{\sqrt{-b^2}\,d}{b}-c}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{\sqrt{-b^2}-a}\,\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{\sqrt{-b^2}-a}\,\sqrt{\frac{\sqrt{-b^2}\,d}{b}-c}}+\frac{d\left(\sqrt{-b^2}\,(a(C\cdots\right.}{\cdots}}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/Sqrt[c + d*Tan[e + f*x]],x]
```

```
[Out] (C*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(d*f) - (((-(b*(A*b +
 a*B - b*C)) + Sqrt[-b^2]*(b*B + a*(-A + C)))*d*ArcTanh[(Sqrt[-c + (Sqrt[-b
^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e
 + f*x]])])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) + ((b*(A*b
+ a*B - b*C) + Sqrt[-b^2]*(b*B + a*(-A + C)))*d*ArcTanh[(Sqrt[c + (Sqrt[-b^
2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e +
 f*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[-b^2]*d)/b]) + (Sqrt[b]*Sqrt
[c - (a*d)/b]*(b*c*C - 2*b*B*d - a*C*d)*ArcSinh[(Sqrt[d]*Sqrt[a + b*Tan[e +
 f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*(c + d*Tan[e + f*x]))/(b*c - a
*d)])/(Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b*d*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{a+b\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\sqrt{c+d\tan\left(fx+e\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(1/2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\sqrt{b\tan\left(fx+e\right)+a}}{\sqrt{d\tan\left(fx+e\right)+c}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*sqrt(b*tan(f*x + e) + a)/
sqrt(d*tan(f*x + e) + c), x)
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(1/2),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{a+b\tan\left(e+fx\right)}\,\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\sqrt{c+d\tan\left(e+fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral(sqrt(a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
sqrt(c + d*tan(e + f*x)), x)
```

3.150 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=239

$$-\frac{(B+i(A-C))\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}\sqrt{c-id}}+\frac{(iA-B-iC)\tanh^{-1}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a+ib}\sqrt{c+id}}+\frac{2C\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}\sqrt{d}f}$$

```
[Out] -(B+I*(A-C))*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/f/(a-I*b)^(1/2)/(c-I*d)^(1/2)+(I*A-B-I*C)*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/f/(a
+I*b)^(1/2)/(c+I*d)^(1/2)+2*C*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2
)/(c+d*tan(f*x+e))^(1/2))/f/b^(1/2)/d^(1/2)
```

Rubi [A] time = 1.46, antiderivative size = 239, normalized size of antiderivative = 1.00, number of steps used = 12, number of rules used = 7, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.143, Rules used = {3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{(B+i(A-C))\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}\sqrt{c-id}}+\frac{(iA-B-iC)\tanh^{-1}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a+ib}\sqrt{c+id}}+\frac{2C\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}\sqrt{d}f}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]*Sqrt[
c + d*Tan[e + f*x]]),x]
```

```
[Out] -(((B + I*(A - C))*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a - I*b]*Sqrt[c - I*d]*f)) + ((I*
A - B - I*C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b
]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*Sqrt[c + I*d]*f) + (2*C*ArcTan
h[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(
Sqrt[b]*Sqrt[d]*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}\,dx = \frac{\text{Subst}\left(\int \frac{A+Bx+Cx^2}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e+fx)\right)}{f}$$

$$= \frac{\text{Subst}\left(\int \left(\frac{C}{\sqrt{a+bx}\sqrt{c+dx}} + \frac{A-C+Bx}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\right)dx, x, \tan(e+fx)\right)}{f}$$

$$= \frac{\text{Subst}\left(\int \frac{A-C+Bx}{\sqrt{a+bx}\sqrt{c+dx}(1+x^2)}\,dx, x, \tan(e+fx)\right)}{f} + \frac{C\,\text{Subst}\left(\int \frac{}{\sqrt{a+}}\right.}{}$$

$$= \frac{\text{Subst}\left(\int \left(\frac{-B+i(A-C)}{2(i-x)\sqrt{a+bx}\sqrt{c+dx}} + \frac{B+i(A-C)}{2(i+x)\sqrt{a+bx}\sqrt{c+dx}}\right)dx, x, \tan(e+f$$

$$= \frac{(-B+i(A-C))\,\text{Subst}\left(\int \frac{1}{(i-x)\sqrt{a+bx}\sqrt{c+dx}}\,dx, x, \tan(e+fx)\right)}{2f} + $$

$$= \frac{2C\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{b}\sqrt{d}\,f} + \frac{(-B+i(A-C))\,\text{Subst}\left(\int \frac{1}{a+ib-(}\right.}{f}$$

$$= -\frac{(B+i(A-C))\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}\sqrt{c-id}\,f} - \frac{(B-i(A-C))\,\text{ta}}{\sqrt{a\,}}$$

Mathematica [A] time = 2.30, size = 362, normalized size = 1.51

$$\frac{\frac{\left(\sqrt{-b^2}(A-C)+bB\right)\tanh^{-1}\left(\frac{\sqrt{\frac{\sqrt{-b^2}\,d}{b}-c}\sqrt{a+b\tan(e+fx)}}{\sqrt{\sqrt{-b^2}-a}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{\sqrt{-b^2}-a}\sqrt{\frac{\sqrt{-b^2}\,d}{b}-c}} - \frac{\left(\sqrt{-b^2}(C-A)+bB\right)\tanh^{-1}\left(\frac{\sqrt{\frac{\sqrt{-b^2}\,d}{b}+c}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+\sqrt{-b^2}}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+\sqrt{-b^2}}\sqrt{\frac{\sqrt{-b^2}\,d}{b}+c}} + \frac{2\sqrt{b}\,C\sqrt{c-\frac{ad}{b}}\sqrt{\frac{b(c+d\tan(e+fx))}{bc-ad}}}{\sqrt{d}\sqrt{c+d\,\text{ta}}}}{bf}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]
*Sqrt[c + d*Tan[e + f*x]]),x]
```

```
[Out] (((b*B + Sqrt[-b^2]*(A - C))*ArcTanh[(Sqrt[-c + (Sqrt[-b^2]*d)/b]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[-a + Sqrt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-
```

```
a + Sqrt[-b^2]]*Sqrt[-c + (Sqrt[-b^2]*d)/b]) - ((b*B + Sqrt[-b^2]*(-A + C))
*ArcTanh[(Sqrt[c + (Sqrt[-b^2]*d)/b]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + Sq
rt[-b^2]]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + Sqrt[-b^2]]*Sqrt[c + (Sqrt[
-b^2]*d)/b]) + (2*Sqrt[b]*C*Sqrt[c - (a*d)/b]*ArcSinh[(Sqrt[d]*Sqrt[a + b*T
an[e + f*x]])/(Sqrt[b]*Sqrt[c - (a*d)/b])]*Sqrt[(b*(c + d*Tan[e + f*x]))/(b
*c - a*d)])/(Sqrt[d]*Sqrt[c + d*Tan[e + f*x]]))/(b*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)}{\sqrt{a+b\tan\left(fx+e\right)}\,\sqrt{c+d\tan\left(fx+e\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(1/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(1/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(1/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(1/2)*(c
+ d*tan(e + f*x))^(1/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\sqrt{a + b\tan\left(e + fx\right)}\,\sqrt{c + d\tan\left(e + fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(1/2)/(c+d*tan
(f*x+e))**(1/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(sqrt(a + b*tan(e + f*x))
*sqrt(c + d*tan(e + f*x))), x)
```

3.151 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=251

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{f\left(a^2+b^2\right)(bc-ad)\sqrt{a+b\tan(e+fx)}}-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{3/2}\sqrt{c-id}}-\frac{(B-i(A-C))\tanh^{-1}}{f(a+ib)^{3/2}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(a-I*b)^(3/2)/f/(c-I*d)^(1/2)-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a+I
*b)^(3/2)/f/(c+I*d)^(1/2)-2*(A*b^2-a*(B*b-C*a))*(c+d*tan(f*x+e))^(1/2)/(a^2
+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(1/2)
```

Rubi [A] time = 0.97, antiderivative size = 251, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 5, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used = {3649, 3616, 3615, 93, 208}

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{f\left(a^2+b^2\right)(bc-ad)\sqrt{a+b\tan(e+fx)}}-\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{f(a-ib)^{3/2}\sqrt{c-id}}-\frac{(B-i(A-C))\tanh^{-1}}{f(a+ib)^{3/2}}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/2)*Sqr
t[c + d*Tan[e + f*x]]),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*Sqrt[c - I*d]*f)) - ((
B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I
*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*Sqrt[c + I*d]*f) - (2*(A*b
^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)*f*Sq
rt[a + b*Tan[e + f*x]])
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}} - \frac{2\int \frac{-\frac{1}{2}(bB+a(A-C}{\sqrt{a}}}{}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}} + \frac{(A-iB-C)\int}{}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}} + \frac{(A-iB-C)\,\mathrm{Su}}{}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}} + \frac{(A-iB-C)\,\mathrm{Su}}{}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}\sqrt{c-id}\,f} - \frac{(B-i(A-C))}{(a}$$

Mathematica [A] time = 2.58, size = 264, normalized size = 1.05

$$\frac{\frac{2\left(a(aC-bB)+Ab^2\right)\sqrt{c+d\tan(e+fx)}}{(ad-bc)\sqrt{a+b\tan(e+fx)}}+\frac{(a+ib)(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{-c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{-a+ib}\sqrt{-c+id}}+\frac{(b+ia)(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a+ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a+ib}\sqrt{c+id}}}{f\left(a^2+b^2\right)}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/
2)*Sqrt[c + d*Tan[e + f*x]]),x]
```

```
[Out] (((a + I*b)*(I*A + B - I*C)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*Sqrt[-c + I*
d]) + ((I*a + b)*(A + I*B - C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*
x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*Sqrt[c + I*d
]) + (2*(A*b^2 + a*(-(b*B) + a*C))*Sqrt[c + d*Tan[e + f*x]])/((-(b*c) + a*d
)*Sqrt[a + b*Tan[e + f*x]]))/((a^2 + b^2)*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)}{\sqrt{c+d\tan\left(fx+e\right)}\,\left(a+b\tan\left(fx+e\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
^(3/2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A}{\left(b\tan\left(fx+e\right)+a\right)^{\frac{3}{2}}\sqrt{d\tan\left(fx+e\right)+c}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)/((b*tan(f*x + e) + a)^(3/
2)*sqrt(d*tan(f*x + e) + c)), x)
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(3/2)*(c
+ d*tan(e + f*x))^(1/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)^{\frac{3}{2}}\sqrt{c + d\tan\left(e + fx\right)}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(1/2)/(a+b*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**(3
/2)*sqrt(c + d*tan(e + f*x))), x)
```

3.152 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}\,dx$

Optimal. Leaf size=375

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))^{3/2}}-\frac{2\sqrt{c+d\tan(e+fx)}\,\left(-2a^4Cd+5a^3bBd-a^2b^2(8Ad+3Bc-4Cd)\right.}{3f\left(a^2+b^2\right)^2(bc-ad)^2\sqrt{a+b\tan}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(a-I*b)^(5/2)/f/(c-I*d)^(1/2)-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a+I
*b)^(5/2)/f/(c+I*d)^(1/2)-2/3*(5*a^3*b*B*d-2*a^4*C*d+b^4*(-2*A*d+3*B*c)+a*b
^3*(6*A*c-B*d-6*C*c)-a^2*b^2*(8*A*d+3*B*c-4*C*d))*(c+d*tan(f*x+e))^(1/2)/(a
^2+b^2)^2/(-a*d+b*c)^2/f/(a+b*tan(f*x+e))^(1/2)-2/3*(A*b^2-a*(B*b-C*a))*(c+
d*tan(f*x+e))^(1/2)/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] time = 1.77, antiderivative size = 375, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 5, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used = {3649, 3616, 3615, 93, 208}

$$-\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))^{3/2}}-\frac{2\sqrt{c+d\tan(e+fx)}\,\left(-a^2b^2(8Ad+3Bc-4Cd)+5a^3bBd-2a^4Cd\right.}{3f\left(a^2+b^2\right)^2(bc-ad)^2\sqrt{a+b\tan}}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(5/2)*Sqr
t[c + d*Tan[e + f*x]]),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])]])/((a - I*b)^(5/2)*Sqrt[c - I*d]*f)) - ((
B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I
*b]*Sqrt[c + d*Tan[e + f*x]])]])/((a + I*b)^(5/2)*Sqrt[c + I*d]*f) - (2*(A*b
^2 - a*(b*B - a*C))*Sqrt[c + d*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c - a*d)*f*
(a + b*Tan[e + f*x])^(3/2)) - (2*(5*a^3*b*B*d - 2*a^4*C*d + b^4*(3*B*c - 2*
A*d) + a*b^3*(6*A*c - 6*c*C - B*d) - a^2*b^2*(3*B*c + 8*A*d - 4*C*d))*Sqrt[
c + d*Tan[e + f*x]])/(3*(a^2 + b^2)^2*(b*c - a*d)^2*f*Sqrt[a + b*Tan[e + f*
x]])
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
```

```
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}\sqrt{c+d\tan(e+fx)}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}} - \frac{2\int \frac{\frac{1}{2}\left(2Ab^2d-3aA\right.}{}}{}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}} - \frac{2\left(5a^3bBd-2a\right.}{}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}} - \frac{2\left(5a^3bBd-2a\right.}{}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}} - \frac{2\left(5a^3bBd-2a\right.}{}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)\sqrt{c+d\tan(e+fx)}}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}} - \frac{2\left(5a^3bBd-2a\right.}{}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{5/2}\sqrt{c-id}\,f} - \frac{(B-i(A-C))\,\mathrm{ta}}{(a+}$$

Mathematica [A] time = 6.28, size = 388, normalized size = 1.03

$$\frac{\frac{2\left(a^2+b^2\right)\left(a(aC-bB)+Ab^2\right)\sqrt{c+d\tan(e+fx)}}{(ad-bc)(a+b\tan(e+fx))^{3/2}} + \frac{2\sqrt{c+d\tan(e+fx)}\left(2a^4Cd-5a^3bBd+a^2b^2(8Ad+3Bc-4Cd)+ab^3(-6Ac+Bd+6cC)+b^4(2Ad-3Bc)\right)}{(bc-ad)^2\sqrt{a+b\tan(e+fx)}} + \frac{3i(}{}}{3f\left(a^2+b^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(5/
2)*Sqrt[c + d*Tan[e + f*x]]),x]
```

```
[Out] ((3*(a + I*b)^2*(I*A + B - I*C)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*Sqrt[-c
+ I*d]) + ((3*I)*(a - I*b)^2*(A + I*B - C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a +
b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*
Sqrt[c + I*d]) + (2*(a^2 + b^2)*(A*b^2 + a*(-(b*B) + a*C))*Sqrt[c + d*Tan[e
 + f*x]])/((-(b*c) + a*d)*(a + b*Tan[e + f*x])^(3/2)) + (2*(-5*a^3*b*B*d +
```

```
2*a^4*C*d + b^4*(-3*B*c + 2*A*d) + a*b^3*(-6*A*c + 6*c*C + B*d) + a^2*b^2*(
3*B*c + 8*A*d - 4*C*d))*Sqrt[c + d*Tan[e + f*x]])/((b*c - a*d)^2*Sqrt[a + b
*Tan[e + f*x]]))/(3*(a^2 + b^2)^2*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(fx+e\right) + C\left(\tan^{2}\left(fx+e\right)\right)}{\sqrt{c+d\tan\left(fx+e\right)}\,\left(a+b\tan\left(fx+e\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))
^(5/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(5/2)*(c
+ d*tan(e + f*x))^(1/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)^{\frac{5}{2}}\sqrt{c + d\tan\left(e + fx\right)}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x+e))**(1/2)/(a+b*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**(5
/2)*sqrt(c + d*tan(e + f*x))), x)
```

3.153 $\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=528

$$\frac{\sqrt{b}\left(15a^2Cd^2-10abd(3cC-2Bd)+b^2\left(8d^2(A-C)-12Bcd+15c^2C\right)\right)\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4d^{7/2}f}+\frac{b\left(d^2(4A+\right.}{}$$

```
[Out] -(a-I*b)^(5/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c-I*d)^(3/2)/f-(a+I*b)^(5/2)*(B-I*(A-C)
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(c+I*d)^(3/2)/f+1/4*(15*a^2*C*d^2-10*a*b*d*(-2*B*d+3*C*c)+b^2*(15
*c^2*C-12*B*c*d+8*(A-C)*d^2))*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2
)/(c+d*tan(f*x+e))^(1/2))*b^(1/2)/d^(7/2)/f-1/4*b*(3*(-a*d+b*c)*(5*c^2*C-4*
B*c*d+(4*A+C)*d^2)-4*d^2*((A-C)*(-a*d+b*c)+B*(a*c+b*d)))*(a+b*tan(f*x+e))^(
1/2)*(c+d*tan(f*x+e))^(1/2)/d^3/(c^2+d^2)/f+1/2*b*(5*c^2*C-4*B*c*d+(4*A+C)*
d^2)*(c+d*tan(f*x+e))^(1/2)*(a+b*tan(f*x+e))^(3/2)/d^2/(c^2+d^2)/f-2*(A*d^2
-B*c*d+C*c^2)*(a+b*tan(f*x+e))^(5/2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] time = 8.19, antiderivative size = 528, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 9, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3645, 3647, 3655, 6725, 63, 217, 206, 93, 208}

$$\frac{\sqrt{b}\left(15a^2Cd^2-10abd(3cC-2Bd)+b^2\left(8d^2(A-C)-12Bcd+15c^2C\right)\right)\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{4d^{7/2}f}+\frac{b\left(d^2(4A+\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((a - I*b)^(5/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(3/2)*f)) -
((a + I*b)^(5/2)*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(3/2)*f) + (Sq
rt[b]*(15*a^2*C*d^2 - 10*a*b*d*(3*c*C - 2*B*d) + b^2*(15*c^2*C - 12*B*c*d +
 8*(A - C)*d^2))*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c
 + d*Tan[e + f*x]])])/(4*d^(7/2)*f) - (2*(c^2*C - B*c*d + A*d^2)*(a + b*Tan
[e + f*x])^(5/2))/(d*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]]) - (b*(3*(b*c -
 a*d)*(5*c^2*C - 4*B*c*d + (4*A + C)*d^2) - 4*d^2*((A - C)*(b*c - a*d) + B*
(a*c + b*d)))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(4*d^3*(c^
```

```
2 + d^2)*f) + (b*(5*c^2*C - 4*B*c*d + (4*A + C)*d^2)*(a + b*Tan[e + f*x])^(
3/2)*Sqrt[c + d*Tan[e + f*x]])/(2*d^2*(c^2 + d^2)*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
```

```
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= \frac{\sqrt{b}\left(15a^2Cd^2-10abd(3cC-2Bd)+b^2\left(15c\right.\right.}{}$$

$$= -\frac{(a-ib)^{5/2}(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b}}{\sqrt{a-ib}\,\sqrt{c+d}}\right.}{(c-id)^{3/2}f}$$

Mathematica [C] time = 44.53, size = 1653959, normalized size = 3132.50

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(fx+e\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{\left(a+b\tan\left(e+f\,x\right)\right)^{5/2}\,\left(C\tan\left(e+f\,x\right)^2+B\tan\left(e+f\,x\right)+A\right)}{\left(c+d\tan\left(e+f\,x\right)\right)^{3/2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(e+fx\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)}{\left(c+d\tan\left(e+fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(c + d*tan(e + f*x))**(3/2), x)
```

3.154 $\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=380

$$\frac{b\left(d^2(2A+C)-2Bcd+3c^2C\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{d^2f\left(c^2+d^2\right)}-\frac{2\left(Ad^2-Bcd+c^2C\right)(a+b\tan(e+fx)}{df\left(c^2+d^2\right)\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(a-I*b)^(3/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c-I*d)^(3/2)/f-(a+I*b)^(3/2)*(B-I*(A-C)
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(c+I*d)^(3/2)/f-(-2*B*b*d-3*C*a*d+3*C*b*c)*arctanh(d^(1/2)*(a+b*t
an(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))*b^(1/2)/d^(5/2)/f+b*(3*c^2
*C-2*B*c*d+(2*A+C)*d^2)*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/d^2/(
c^2+d^2)/f-2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^(3/2)/d/(c^2+d^2)/f/(c+d*
tan(f*x+e))^(1/2)
```

Rubi [A] time = 5.63, antiderivative size = 380, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 9, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3645, 3647, 3655, 6725, 63, 217, 206, 93, 208}

$$\frac{b\left(d^2(2A+C)-2Bcd+3c^2C\right)\sqrt{a+b\tan(e+fx)}\sqrt{c+d\tan(e+fx)}}{d^2f\left(c^2+d^2\right)}-\frac{2\left(Ad^2-Bcd+c^2C\right)(a+b\tan(e+fx)}{df\left(c^2+d^2\right)\sqrt{c+d\tan(e+fx)}}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] -(((a - I*b)^(3/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(3/2)*f)) -
((a + I*b)^(3/2)*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(3/2)*f) - (Sq
rt[b]*(3*b*c*C - 2*b*B*d - 3*a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(d^(5/2)*f) - (2*(c^2*C - B*c*d +
A*d^2)*(a + b*Tan[e + f*x])^(3/2))/(d*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]
]) + (b*(3*c^2*C - 2*B*c*d + (2*A + C)*d^2)*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c
 + d*Tan[e + f*x]])/(d^2*(c^2 + d^2)*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
```

```
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
```

```
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3/2}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{\sqrt{b}\,(3bcC-2bBd-3aCd)\tanh^{-1}\left(\frac{\sqrt{d}\,\sqrt{a+b}}{\sqrt{b}\,\sqrt{c+d}}\right.}{d^{5/2}f}$$

$$= -\frac{(a-ib)^{3/2}(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b}}{\sqrt{a-ib}\,\sqrt{c+d}}\right.}{(c-id)^{3/2}f}$$

Mathematica [C] time = 39.86, size = 1073499, normalized size = 2825.00

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a + b\tan\left(fx + e\right)\right)^{\frac{3}{2}}\left(A + B\tan\left(fx + e\right) + C\left(\tan^{2}\left(fx + e\right)\right)\right)}{\left(c + d\tan\left(fx + e\right)\right)^{\frac{3}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{\left(a+b\tan\left(e+f\,x\right)\right)^{3/2}\,\left(C\tan\left(e+f\,x\right)^2+B\tan\left(e+f\,x\right)+A\right)}{\left(c+d\tan\left(e+f\,x\right)\right)^{3/2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(e+fx\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)}{\left(c+d\tan\left(e+fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(c + d*tan(e + f*x))**(3/2), x)
```

3.155 $\int \frac{\sqrt{a+b\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=299

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)\sqrt{a+b\tan(e+fx)}}{df\left(c^2+d^2\right)\sqrt{c+d\tan(e+fx)}}-\frac{\sqrt{a-ib}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f(c-id)^{3/2}}-\frac{\sqrt{a+ib}\,(B-i(A$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)/(c-I*d)^(3/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a+I
*b)^(1/2)/(c+I*d)^(3/2)/f+2*C*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2)/b^(1/2
)/(c+d*tan(f*x+e))^(1/2))*b^(1/2)/d^(3/2)/f-2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(
f*x+e))^(1/2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] time = 3.33, antiderivative size = 299, normalized size of antiderivative = 1.00, number of steps used = 13, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.163, Rules used = {3645, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)\sqrt{a+b\tan(e+fx)}}{df\left(c^2+d^2\right)\sqrt{c+d\tan(e+fx)}}-\frac{\sqrt{a-ib}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f(c-id)^{3/2}}-\frac{\sqrt{a+ib}\,(B-i(A$$

Antiderivative was successfully verified.

```
[In] Int[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c +
 d*Tan[e + f*x])^(3/2),x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(3/2)*f)) - (S
qrt[a + I*b]*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(3/2)*f) + (2*Sqrt
[b]*C*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Tan[e
+ f*x]])])/(d^(3/2)*f) - (2*(c^2*C - B*c*d + A*d^2)*Sqrt[a + b*Tan[e + f*x]
])/(d*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
```

```
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{\sqrt{a+b\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{d\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}}+$$

$$= \frac{2\sqrt{b}\,C\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{3/2}f}-\frac{2\left(c^2C-\right.}{d\left(c^2\right.}$$

$$= -\frac{\sqrt{a-ib}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan}}{\sqrt{a-ib}\sqrt{c+d\tan}}\right.}{(c-id)^{3/2}f}$$

Mathematica [C] time = 35.59, size = 621084, normalized size = 2077.20

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(c + d*Tan[e + f*x])^(3/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{a+b\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(3/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{\sqrt{a+b\tan\left(e+f\,x\right)}\,\left(C\tan\left(e+f\,x\right)^2+B\tan\left(e+f\,x\right)+A\right)}{\left(c+d\tan\left(e+f\,x\right)\right)^{3/2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(3/2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{a+b\tan\left(e+fx\right)}\,\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\left(c+d\tan\left(e+fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral(sqrt(a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(c + d*tan(e + f*x))**(3/2), x)
```

3.156 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=251

$$\frac{2\left(Ad^2-Bcd+c^2C\right)\sqrt{a+b\tan(e+fx)}}{f\left(c^2+d^2\right)(bc-ad)\sqrt{c+d\tan(e+fx)}}-\frac{(B+i(A-C))\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}\,(c-id)^{3/2}}+\frac{(iA-B-iC)\tanh^{-1}\left(\frac{\sqrt{c}}{\sqrt{a}}\right.}{f\sqrt{a+ib}\,(c+}$$

```
[Out] -(B+I*(A-C))*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(c-I*d)^(3/2)/f/(a-I*b)^(1/2)+(I*A-B-I*C)*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c+I
*d)^(3/2)/f/(a+I*b)^(1/2)+2*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^(1/2)/(-a*
d+b*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] time = 1.00, antiderivative size = 251, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 5, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used = {3649, 3616, 3615, 93, 208}

$$\frac{2\left(Ad^2-Bcd+c^2C\right)\sqrt{a+b\tan(e+fx)}}{f\left(c^2+d^2\right)(bc-ad)\sqrt{c+d\tan(e+fx)}}-\frac{(B+i(A-C))\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{f\sqrt{a-ib}\,(c-id)^{3/2}}+\frac{(iA-B-iC)\tanh^{-1}\left(\frac{\sqrt{c}}{\sqrt{a}}\right.}{f\sqrt{a+ib}\,(c+}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]*(c +
d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] -(((B + I*(A - C))*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a - I*b]*(c - I*d)^(3/2)*f)) + ((
I*A - B - I*C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I
*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*b]*(c + I*d)^(3/2)*f) + (2*(c^2
*C - B*c*d + A*d^2)*Sqrt[a + b*Tan[e + f*x]])/((b*c - a*d)*(c^2 + d^2)*f*Sq
rt[c + d*Tan[e + f*x]])
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{3/2}}\,dx = \frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{(bc-ad)\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{2\int \frac{\frac{1}{2}(bc-ad)(Ac-cC+B}{\sqrt{a+b\tan}}}{(bc}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{(bc-ad)\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\int \frac{}{\sqrt{a+}}}{2}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{(bc-ad)\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\,\mathrm{Subst}}{}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{(bc-ad)\left(c^2+d^2\right)f\sqrt{c+d\tan(e+fx)}} + \frac{(A-iB-C)\,\mathrm{Subst}}{}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}\,(c-id)^{3/2}f} - \frac{(B-i(A-C))\,\mathrm{ta}}{\sqrt{a-}}$$

Mathematica [A] time = 3.21, size = 275, normalized size = 1.10

$$-\frac{\frac{2\left(Ad^2-Bcd+c^2C\right)\sqrt{a+b\tan(e+fx)}}{\sqrt{c+d\tan(e+fx)}} + (bc-ad)\left(\frac{(c+id)(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{-c+id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{-a+ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{-a+ib}\,\sqrt{-c+id}} + \frac{(d+ic)(A+iB-C)\tanh^{-1}\left(\frac{\sqrt{c+id}\,\sqrt{a+b\tan(e+}}{\sqrt{a+ib}\,\sqrt{c+d\tan(e+}}\right)}{\sqrt{a+ib}\,\sqrt{c+id}}\right.}{f\left(c^2+d^2\right)(ad-bc)}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]
*(c + d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] -(((b*c - a*d)*(((I*A + B - I*C)*(c + I*d)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a +
 b*Tan[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*
b]*Sqrt[-c + I*d]) + ((A + I*B - C)*(I*c + d)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a
 + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[a + I*
b]*Sqrt[c + I*d])) + (2*(c^2*C - B*c*d + A*d^2)*Sqrt[a + b*Tan[e + f*x]])/S
qrt[c + d*Tan[e + f*x]])/((-(b*c) + a*d)*(c^2 + d^2)*f))
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)}{\sqrt{a+b\tan\left(fx+e\right)}\,\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A}{\sqrt{b\tan\left(fx+e\right)+a}\left(d\tan\left(fx+e\right)+c\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)/(sqrt(b*tan(f*x + e) + a)
*(d*tan(f*x + e) + c)^(3/2)), x)
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(1/2)*(c
+ d*tan(e + f*x))^(3/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)}{\sqrt{a + b\tan\left(e + fx\right)}\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(1/2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(sqrt(a + b*tan(e + f*x))
*(c + d*tan(e + f*x))**(3/2)), x)
```

3.157 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=383

$$-\frac{2d\sqrt{a+b\tan(e+fx)}\left(A\left(a^2d^2+b^2\left(c^2+2d^2\right)\right)+a^2\left(-Bcd+2c^2C+Cd^2\right)-abB\left(c^2+d^2\right)+b^2c(cC-Bd)\right)}{f\left(a^2+b^2\right)\left(c^2+d^2\right)(bc-ad)^2\sqrt{c+d\tan(e+fx)}}-$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(a-I*b)^(3/2)/(c-I*d)^(3/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a+I
*b)^(3/2)/(c+I*d)^(3/2)/f-2*(A*b^2-a*(B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(a+b
*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))^(1/2)-2*d*(b^2*c*(-B*d+C*c)-a*b*B*(c^2+
d^2)+a^2*(-B*c*d+2*C*c^2+C*d^2)+A*(a^2*d^2+b^2*(c^2+2*d^2)))*(a+b*tan(f*x+e
))^(1/2)/(a^2+b^2)/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)
```

Rubi [A] time = 1.88, antiderivative size = 382, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 5, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used = {3649, 3616, 3615, 93, 208}

$$-\frac{2d\sqrt{a+b\tan(e+fx)}\left(a^2Ad^2+a^2\left(-Bcd+2c^2C+Cd^2\right)-abB\left(c^2+d^2\right)+Ab^2\left(c^2+2d^2\right)+b^2c(cC-Bd)\right)}{f\left(a^2+b^2\right)\left(c^2+d^2\right)(bc-ad)^2\sqrt{c+d\tan(e+fx)}}-$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/2)*(c
+ d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*(c - I*d)^(3/2)*f)) -
((B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a +
 I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*(c + I*d)^(3/2)*f) - (2*
(A*b^2 - a*(b*B - a*C)))/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e + f*x]
]*Sqrt[c + d*Tan[e + f*x]]) - (2*d*(a^2*A*d^2 + b^2*c*(c*C - B*d) - a*b*B*(
c^2 + d^2) + A*b^2*(c^2 + 2*d^2) + a^2*(2*c^2*C - B*c*d + C*d^2))*Sqrt[a +
b*Tan[e + f*x]])/((a^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f*Sqrt[c + d*Tan[e
+ f*x]])
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
```

```
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}(c-id)^{3/2}f} - \frac{(B-i(A-C}{(a}$$

Mathematica [A] time = 6.72, size = 484, normalized size = 1.26

$$-\frac{2\left(Ab^2-a(bB-aC)\right)}{f\left(a^2+b^2\right)(bc-ad)\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}} - \frac{2\left(-\frac{2\sqrt{a+b\tan(e+fx)}\left(\frac{1}{2}d^2\left(-aA(bc-ad)-(bB-aC)(ad+bc)+2A\right.\right.}{f\left(c^2+d^2\right)(ad-bc)\sqrt{}}\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/
2)*(c + d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] (-2*(A*b^2 - a*(b*B - a*C)))/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e +
f*x]]*Sqrt[c + d*Tan[e + f*x]]) - (2*(((b*c - a*d)^2*(((a + I*b)*(I*A + B -
 I*C)*(c + I*d)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[-a
+ I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*Sqrt[-c + I*d]) + ((I*a
```

```
+ b)*(A + I*B - C)*(c - I*d)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a + I*b]*Sqrt[c + I*d])
))/(2*(-(b*c) + a*d)*(c^2 + d^2)*f) - (2*(-(c*(-(c*(A*b^2 - a*(b*B - a*C))*
d) + ((A*b - a*B - b*C)*d*(b*c - a*d))/2)) + (d^2*(2*A*b^2*d - a*A*(b*c - a
*d) - (b*B - a*C)*(b*c + a*d)))/2)*Sqrt[a + b*Tan[e + f*x]])/((-(b*c) + a*d
)*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])))/((a^2 + b^2)*(b*c - a*d))
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(fx+e\right) + C\left(\tan^{2}\left(fx+e\right)\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(3/2)*(c
+ d*tan(e + f*x))^(3/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)^{\frac{3}{2}}\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(3/2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**(3
/2)*(c + d*tan(e + f*x))**(3/2)), x)
```

3.158 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx$

Optimal. Leaf size=598

$$-\frac{2\left(Ab^2-a(bB-aC)\right)}{3f\left(a^2+b^2\right)(bc-ad)(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}-\frac{2d\sqrt{a+b\tan(e+fx)}\,\left(a^4(-d)\left(d^2(3A+5C)-\right.\right.}{}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(a-I*b)^(5/2)/(c-I*d)^(3/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a+I
*b)^(5/2)/(c+I*d)^(3/2)/f-2/3*(7*a^3*b*B*d-4*a^4*C*d+b^4*(-4*A*d+3*B*c)+a*b
^3*(6*A*c+B*d-6*C*c)-a^2*b^2*(3*B*c+2*(5*A-C)*d))/(a^2+b^2)^2/(-a*d+b*c)^2/
f/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))^(1/2)-2/3*d*(8*a^3*b*B*d*(c^2+d^2
)+2*a*b^3*(3*A*c+B*d-3*C*c)*(c^2+d^2)-a^4*d*(8*c^2*C-3*B*c*d+(3*A+5*C)*d^2)
-a^2*b^2*(11*A*c^2*d+17*A*d^3+3*B*c^3-3*B*c*d^2+5*C*c^2*d-C*d^3)-b^4*(d*(5*
A*c^2+8*A*d^2+3*C*c^2)-3*B*(c^3+2*c*d^2)))*(a+b*tan(f*x+e))^(1/2)/(a^2+b^2)
^2/(-a*d+b*c)^3/(c^2+d^2)/f/(c+d*tan(f*x+e))^(1/2)-2/3*(A*b^2-a*(B*b-C*a))/
(a^2+b^2)/(-a*d+b*c)/f/(c+d*tan(f*x+e))^(1/2)/(a+b*tan(f*x+e))^(3/2)
```

Rubi [A] time = 3.43, antiderivative size = 598, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 5, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}} = 0.102$, Rules used = {3649, 3616, 3615, 93, 208}

$$-\frac{2d\sqrt{a+b\tan(e+fx)}\left(-a^2b^2\left(11Ac^2d+17Ad^3+3Bc^3-3Bcd^2+5c^2Cd-Cd^3\right)+a^4(-d)\left(d^2(3A+5C)-3Bcd\right.\right.}{3f\left(a^2+b^2\right)^2\left(c^2+d^2\right)(b}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(5/2)*(c
+ d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(5/2)*(c - I*d)^(3/2)*f)) -
((B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a +
 I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(5/2)*(c + I*d)^(3/2)*f) - (2*
(A*b^2 - a*(b*B - a*C)))/(3*(a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*x])^
(3/2)*Sqrt[c + d*Tan[e + f*x]]) - (2*(7*a^3*b*B*d - 4*a^4*C*d + b^4*(3*B*c
- 4*A*d) + a*b^3*(6*A*c - 6*c*C + B*d) - a^2*b^2*(3*B*c + 2*(5*A - C)*d)))/
(3*(a^2 + b^2)^2*(b*c - a*d)^2*f*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e
+ f*x]]) - (2*d*(8*a^3*b*B*d*(c^2 + d^2) + 2*a*b^3*(3*A*c - 3*c*C + B*d)*(c
```

```
^2 + d^2) - a^4*d*(8*c^2*C - 3*B*c*d + (3*A + 5*C)*d^2) - a^2*b^2*(3*B*c^3
+ 11*A*c^2*d + 5*c^2*C*d - 3*B*c*d^2 + 17*A*d^3 - C*d^3) - b^4*(d*(5*A*c^2
+ 3*c^2*C + 8*A*d^2) - 3*B*(c^3 + 2*c*d^2)))*Sqrt[a + b*Tan[e + f*x]])/(3*(
a^2 + b^2)^2*(b*c - a*d)^3*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
```

```
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{5/2}(c+d\tan(e+fx))^{3/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+f}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+f}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+f}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+f}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+f}}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+f}}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{5/2}(c-id)^{3/2}f} - \frac{(B-i(A-C))}{(a-}$$

Mathematica [A] time = 6.90, size = 902, normalized size = 1.51

$$-\frac{2\left(Ab^2-a(bB-aC)\right)}{3\left(a^2+b^2\right)(bc-ad)f(a+b\tan(e+fx))^{3/2}\sqrt{c+d\tan(e+fx)}}-\frac{2\left(-\frac{2\left(\frac{1}{2}b^2\left(4Adb^2-3aA(bc-ad)-(bB-aC)(3bc+ad)\right)-a\left(\frac{3}{2}b(\right.\right.}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+f}}\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(5/
2)*(c + d*Tan[e + f*x])^(3/2)),x]
```

```
[Out] (-2*(A*b^2 - a*(b*B - a*C)))/(3*(a^2 + b^2)*(b*c - a*d)*f*(a + b*Tan[e + f*
x])^(3/2)*Sqrt[c + d*Tan[e + f*x]]) - (2*((-2*(-(a*(-2*a*(A*b^2 - a*(b*B -
a*C))*d + (3*b*(A*b - a*B - b*C)*(b*c - a*d))/2)) + (b^2*(4*A*b^2*d - 3*a*A
*(b*c - a*d) - (b*B - a*C)*(3*b*c + a*d)))/2))/((a^2 + b^2)*(b*c - a*d)*f*S
qrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]]) - (2*((-3*(b*c - a*d)^3*(
((a + I*b)^2*(I*A + B - I*C)*(c + I*d)*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*T
an[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*S
qrt[-c + I*d]) + ((a - I*b)^2*(A + I*B - C)*(I*c + d)*ArcTanh[(Sqrt[c + I*d
]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqr
t[a + I*b]*Sqrt[c + I*d])))/(4*(-(b*c) + a*d)*(c^2 + d^2)*f) - (2*(d^2*((-1
/2*(b*c) - (a*d)/2)*(-2*a*(A*b^2 - a*(b*B - a*C))*d + (3*b*(A*b - a*B - b*C
)*(b*c - a*d))/2) + ((b^2*d - (a*(b*c - a*d))/2)*(4*A*b^2*d - 3*a*A*(b*c -
a*d) - (b*B - a*C)*(3*b*c + a*d)))/2) - c*((d*(b*c - a*d)*(-2*b*(A*b^2 - a*
(b*B - a*C))*d - (3*a*(A*b - a*B - b*C)*(b*c - a*d))/2 + (b*(4*A*b^2*d - 3*
a*A*(b*c - a*d) - (b*B - a*C)*(3*b*c + a*d)))/2))/2 - c*d*(-(a*(-2*a*(A*b^2
 - a*(b*B - a*C))*d + (3*b*(A*b - a*B - b*C)*(b*c - a*d))/2)) + (b^2*(4*A*b
^2*d - 3*a*A*(b*c - a*d) - (b*B - a*C)*(3*b*c + a*d)))/2)))*Sqrt[a + b*Tan[
e + f*x]])/((-(b*c) + a*d)*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])))/((a^2
+ b^2)*(b*c - a*d))))/(3*(a^2 + b^2)*(b*c - a*d))
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(5/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(5/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{5}{2}}\left(c+d\tan\left(fx+e\right)\right)^{\frac{3}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(5/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(5/2)/(c+d*tan(f*x+e))
^(3/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(5/2)/(c+d*tan(f
*x+e))^(3/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(5/2)*(c
+ d*tan(e + f*x))^(3/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)^{\frac{5}{2}}\left(c + d\tan\left(e + fx\right)\right)^{\frac{3}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(5/2)/(c+d*tan
(f*x+e))**(3/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**(5
/2)*(c + d*tan(e + f*x))**(3/2)), x)
```

3.159 $\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=549

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)(a+b\tan(e+fx))^{5/2}}{3df\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}}+\frac{b\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}\,\left(2ad^2\left(2cd(A-C)-B\left(c^2\right.\right.\right.}{d^3f\left(c^2+\right.}$$

```
[Out] -(a-I*b)^(5/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c-I*d)^(5/2)/f-(a+I*b)^(5/2)*(B-I*(A-C)
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(c+I*d)^(5/2)/f-b^(3/2)*(-2*B*b*d-5*C*a*d+5*C*b*c)*arctanh(d^(1/2
)*(a+b*tan(f*x+e))^(1/2)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/d^(7/2)/f+b*(b*(5*
c^4*C-2*B*c^3*d+10*c^2*C*d^2-6*B*c*d^3+(4*A+C)*d^4)+2*a*d^2*(2*c*(A-C)*d-B*
(c^2-d^2)))*(a+b*tan(f*x+e))^(1/2)*(c+d*tan(f*x+e))^(1/2)/d^3/(c^2+d^2)^2/f
-2/3*(b*(5*c^4*C-2*B*c^3*d-c^2*(A-11*C)*d^2-8*B*c*d^3+5*A*d^4)+3*a*d^2*(2*c
*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*x+e))^(3/2)/d^2/(c^2+d^2)^2/f/(c+d*tan(f*
x+e))^(1/2)-2/3*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^(5/2)/d/(c^2+d^2)/f/(c
+d*tan(f*x+e))^(3/2)
```

Rubi [A] time = 10.50, antiderivative size = 549, normalized size of antiderivative = 1.00, number of steps used = 15, number of rules used = 9, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.184, Rules used = {3645, 3647, 3655, 6725, 63, 217, 206, 93, 208}

$$\frac{b\sqrt{a+b\tan(e+fx)}\,\sqrt{c+d\tan(e+fx)}\,\left(2ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(d^4(4A+C)-2Bc^3d-6Bcd^3+10\right.\right.}{d^3f\left(c^2+d^2\right)^2}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((a - I*b)^(5/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(5/2)*f)) -
((a + I*b)^(5/2)*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(5/2)*f) - (b^
(3/2)*(5*b*c*C - 2*b*B*d - 5*a*C*d)*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x
]])/(Sqrt[b]*Sqrt[c + d*Tan[e + f*x]])])/(d^(7/2)*f) - (2*(c^2*C - B*c*d +
A*d^2)*(a + b*Tan[e + f*x])^(5/2))/(3*d*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^
(3/2)) - (2*(b*(5*c^4*C - 2*B*c^3*d - c^2*(A - 11*C)*d^2 - 8*B*c*d^3 + 5*A*
d^4) + 3*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*(a + b*Tan[e + f*x])^(3/2))
```

```
/(3*d^2*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]]) + (b*(b*(5*c^4*C - 2*B*c^
3*d + 10*c^2*C*d^2 - 6*B*c*d^3 + (4*A + C)*d^4) + 2*a*d^2*(2*c*(A - C)*d -
B*(c^2 - d^2)))*Sqrt[a + b*Tan[e + f*x]]*Sqrt[c + d*Tan[e + f*x]])/(d^3*(c^
2 + d^2)^2*f)
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
```

```
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^{5/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{5}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{b^{3/2}(5bcC-2bBd-5aCd)\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a}}{\sqrt{b}\sqrt{c}}\right.}{d^{7/2}f}$$

$$= -\frac{(a-ib)^{5/2}(B+i(A-C))\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a}}{\sqrt{a-ib}\sqrt{c}}\right.}{(c-id)^{5/2}f}$$

Mathematica [C] time = 47.17, size = 2018643, normalized size = 3676.95

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[((a + b*Tan[e + f*x])^(5/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(c + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(fx+e\right)\right)^{\frac{5}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{\left(a + b\tan\left(e + f\,x\right)\right)^{5/2}\,\left(C\tan\left(e + f\,x\right)^2 + B\tan\left(e + f\,x\right) + A\right)}{\left(c + d\tan\left(e + f\,x\right)\right)^{5/2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(5/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2), x)
```

sympy [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(5/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(5/2),x)
```

```
[Out] Timed out
```

3.160 $\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=407

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)(a+b\tan(e+fx))^{3/2}}{3df\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}}-\frac{2\sqrt{a+b\tan(e+fx)}\left(ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-c^2d^2(A\right.\right.}{d^2f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}}$$

```
[Out] -(a-I*b)^(3/2)*(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-
I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c-I*d)^(5/2)/f-(a+I*b)^(3/2)*(B-I*(A-C)
)*arctanh((c+I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e
))^(1/2))/(c+I*d)^(5/2)/f+2*b^(3/2)*C*arctanh(d^(1/2)*(a+b*tan(f*x+e))^(1/2
)/b^(1/2)/(c+d*tan(f*x+e))^(1/2))/d^(5/2)/f-2*(b*(c^4*C-c^2*(A-3*C)*d^2-2*B
*c*d^3+A*d^4)+a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*x+e))^(1/2)/d^2/(
c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)-2/3*(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e)
)^(3/2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] time = 7.16, antiderivative size = 407, normalized size of antiderivative = 1.00, number of steps used = 14, number of rules used = 8, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}} = 0.163$, Rules used = {3645, 3655, 6725, 63, 217, 206, 93, 208}

$$-\frac{2\sqrt{a+b\tan(e+fx)}\left(ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-c^2d^2(A-3C)+Ad^4-2Bcd^3+c^4C\right)\right)}{d^2f\left(c^2+d^2\right)^2\sqrt{c+d\tan(e+fx)}}-\frac{2\left(Ad^2-Bc\right.}{3df\left(c^2\right.}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c
 + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -(((a - I*b)^(3/2)*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e
+ f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(5/2)*f)) -
((a + I*b)^(3/2)*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(5/2)*f) + (2*
b^(3/2)*C*ArcTanh[(Sqrt[d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[b]*Sqrt[c + d*Ta
n[e + f*x]])])/(d^(5/2)*f) - (2*(c^2*C - B*c*d + A*d^2)*(a + b*Tan[e + f*x]
)^(3/2))/(3*d*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2)) - (2*(b*(c^4*C - c^
2*(A - 3*C)*d^2 - 2*B*c*d^3 + A*d^4) + a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)
))*Sqrt[a + b*Tan[e + f*x]])/(d^2*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 63

```
Int[((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_), x_Symbol] :> With[
{p = Denominator[m]}, Dist[p/b, Subst[Int[x^(p*(m + 1) - 1)*(c - (a*d)/b +
```

```
(d*x^p)/b)^n, x], x, (a + b*x)^(1/p)], x]] /; FreeQ[{a, b, c, d}, x] && NeQ
[b*c - a*d, 0] && LtQ[-1, m, 0] && LeQ[-1, n, 0] && LeQ[Denominator[n], Den
ominator[m]] && IntLinearQ[a, b, c, d, m, n, x]
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 206

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(1*ArcTanh[(Rt[-b, 2]*x)/
Rt[a, 2]])/(Rt[a, 2]*Rt[-b, 2]), x] /; FreeQ[{a, b}, x] && NegQ[a/b] && (Gt
Q[a, 0] || LtQ[b, 0])
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 217

```
Int[1/Sqrt[(a_) + (b_.)*(x_)^2], x_Symbol] :> Subst[Int[1/(1 - b*x^2), x],
x, x/Sqrt[a + b*x^2]] /; FreeQ[{a, b}, x] &&  !GtQ[a, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
```

```
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^{3/2}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{3}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}$$

$$= \frac{2b^{3/2}C\tanh^{-1}\left(\frac{\sqrt{d}\sqrt{a+b\tan(e+fx)}}{\sqrt{b}\sqrt{c+d\tan(e+fx)}}\right)}{d^{5/2}f} - \frac{2\left(c^2C\right.}{3d}$$

$$= -\frac{(a-ib)^{3/2}(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+}}{\sqrt{a-ib}\sqrt{c+}}\right.}{(c-id)^{5/2}f}$$

Mathematica [C] time = 41.12, size = 1347117, normalized size = 3309.87

Result too large to show

Warning: Unable to verify antiderivative.

```
[In] Integrate[((a + b*Tan[e + f*x])^(3/2)*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^
2))/(c + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] Result too large to show
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{\left(a+b\tan\left(e+f\,x\right)\right)^{3/2}\,\left(C\tan\left(e+f\,x\right)^2+B\tan\left(e+f\,x\right)+A\right)}{\left(c+d\tan\left(e+f\,x\right)\right)^{5/2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(e+fx\right)\right)^{\frac{3}{2}}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\left(c+d\tan\left(e+fx\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(3/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**(3/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)**
2)/(c + d*tan(e + f*x))**(5/2), x)
```

3.161 $\int \frac{\sqrt{a+b\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=373

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)\sqrt{a+b\tan(e+fx)}}{3df\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}}+\frac{2\sqrt{a+b\tan(e+fx)}\left(3ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-c^2d^2(5A\right.\right.}{3df\left(c^2+d^2\right)^2(bc-ad)\sqrt{c+d\tan(}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))*(a-I*b)^(1/2)/(c-I*d)^(5/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))*(a+I
*b)^(1/2)/(c+I*d)^(5/2)/f+2/3*(b*(c^4*C+2*B*c^3*d-c^2*(5*A-7*C)*d^2-4*B*c*d
^3+A*d^4)+3*a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*x+e))^(1/2)/d/(-a*d
+b*c)/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)-2/3*(A*d^2-B*c*d+C*c^2)*(a+b*tan
(f*x+e))^(1/2)/d/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] time = 1.92, antiderivative size = 373, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.122, Rules used = {3645, 3649, 3616, 3615, 93, 208}

$$-\frac{2\left(Ad^2-Bcd+c^2C\right)\sqrt{a+b\tan(e+fx)}}{3df\left(c^2+d^2\right)(c+d\tan(e+fx))^{3/2}}+\frac{2\sqrt{a+b\tan(e+fx)}\left(3ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)+b\left(-c^2d^2(5A\right.\right.}{3df\left(c^2+d^2\right)^2(bc-ad)\sqrt{c+d\tan(}$$

Antiderivative was successfully verified.

```
[In] Int[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c +
 d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -((Sqrt[a - I*b]*(I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e +
f*x]])/(Sqrt[a - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c - I*d)^(5/2)*f)) - (S
qrt[a + I*b]*(B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]
])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/((c + I*d)^(5/2)*f) - (2*(c^2
*C - B*c*d + A*d^2)*Sqrt[a + b*Tan[e + f*x]])/(3*d*(c^2 + d^2)*f*(c + d*Tan
[e + f*x])^(3/2)) + (2*(b*(c^4*C + 2*B*c^3*d - c^2*(5*A - 7*C)*d^2 - 4*B*c*
d^3 + A*d^4) + 3*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Sqrt[a + b*Tan[e +
f*x]])/(3*d*(b*c - a*d)*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
```

```
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3645

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*d^2 + c*(c*C - B*d))*(a + b*Tan[e
+ f*x])^m*(c + d*Tan[e + f*x])^(n + 1))/(d*f*(n + 1)*(c^2 + d^2)), x] - Dis
t[1/(d*(n + 1)*(c^2 + d^2)), Int[(a + b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e
+ f*x])^(n + 1)*Simp[A*d*(b*d*m - a*c*(n + 1)) + (c*C - B*d)*(b*c*m + a*d*(
n + 1)) - d*(n + 1)*((A - C)*(b*c - a*d) + B*(a*c + b*d))*Tan[e + f*x] - b*
(d*(B*c - A*d)*(m + n + 1) - C*(c^2*m - d^2*(n + 1)))*Tan[e + f*x]^2, x], x
], x] /; FreeQ[{a, b, c, d, e, f, A, B, C}, x] && NeQ[b*c - a*d, 0] && NeQ[
a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && GtQ[m, 0] && LtQ[n, -1]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
```

```
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{\sqrt{a+b\tan(e+fx)}\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$= -\frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{3d\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}}+$$

$$= -\frac{\sqrt{a-ib}\,(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(}}{\sqrt{a-ib}\,\sqrt{c+d\tan(}}\right.}{(c-id)^{5/2}f}$$

Mathematica [A] time = 7.02, size = 609, normalized size = 1.63

$$-\frac{C\sqrt{a+b\tan(e+fx)}}{df(c+d\tan(e+fx))^{3/2}}-\frac{-\frac{2\sqrt{a+b\tan(e+fx)}\left(\frac{1}{2}d^2(-ad(2A-3C)-bcC)-c\left(-\left(d^2(aB+Ab-bC)\right)-\frac{1}{2}c(aCd-2bBd-bcC)\right)\right)}{3f\left(c^2+d^2\right)(ad-bc)(c+d\tan(e+fx))^{3/2}}-\frac{2\left(-\frac{2\sqrt{a+b\tan(e+fx)}}{}\right.}{}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(Sqrt[a + b*Tan[e + f*x]]*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)
)/(c + d*Tan[e + f*x])^(5/2),x]
```

```
[Out] -((C*Sqrt[a + b*Tan[e + f*x]])/(d*f*(c + d*Tan[e + f*x])^(3/2))) - ((-2*((d
^2*(-(b*c*C) - a*(2*A - 3*C)*d))/2 - c*(-((A*b + a*B - b*C)*d^2) - (c*(-(b*
c*C) - 2*b*B*d + a*C*d))/2))*Sqrt[a + b*Tan[e + f*x]])/(3*(-(b*c) + a*d)*(c
^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2)) - (2*((-3*d*(b*c - a*d)^2*((Sqrt[-a
 + I*b]*(I*A + B - I*C)*(c + I*d)^2*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*Tan[
e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/Sqrt[-c + I*d] + (Sq
rt[a + I*b]*(B - I*(A - C))*(c - I*d)^2*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*T
an[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/Sqrt[c + I*d]))/(2
*(-(b*c) + a*d)*(c^2 + d^2)*f) - (2*(-1/2*(d^2*(b*c - a*d)*(3*a*d*(A*c - c*
C + B*d) + b*(c^2*C - B*c*d + A*d^2))) - c*((-3*d^2*(b*c - a*d)*(A*b*c + a*
B*c - b*c*C - a*A*d + b*B*d + a*C*d))/2 + (b*c*(b*c - a*d)*(c^2*C + 2*B*c*d
 - (2*A - 3*C)*d^2))/2))*Sqrt[a + b*Tan[e + f*x]])/((-(b*c) + a*d)*(c^2 + d
^2)*f*Sqrt[c + d*Tan[e + f*x]])))/(3*(-(b*c) + a*d)*(c^2 + d^2)))/d
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{a+b\tan\left(fx+e\right)}\,\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))
^(5/2),x)
```

maxima [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: ValueError

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Exception raised: ValueError >> Computation failed since Maxima requested a
dditional constraints; using the 'assume' command before evaluation *may* h
elp (example of legal syntax is 'assume(((2*b*d+2*a*c)^2>0)', see `assume?`
 for more details)Is ((2*b*d+2*a*c)^2    -4*((a*c-b*d)^2       -((-a*d)-b*c
)        *(a*d+b*c)))    ^2 positive or zero?
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{\sqrt{a+b\tan\left(e+f\,x\right)}\,\left(C\tan\left(e+f\,x\right)^2+B\tan\left(e+f\,x\right)+A\right)}{{\left(c+d\tan\left(e+f\,x\right)\right)}^{5/2}} \,d x$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2),x)
```

```
[Out] int(((a + b*tan(e + f*x))^(1/2)*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c
 + d*tan(e + f*x))^(5/2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\sqrt{a+b\tan\left(e+fx\right)}\left(A+B\tan\left(e+fx\right)+C\tan^{2}\left(e+fx\right)\right)}{\left(c+d\tan\left(e+fx\right)\right)^{\frac{5}{2}}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**(1/2)*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral(sqrt(a + b*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/
(c + d*tan(e + f*x))**(5/2), x)
```

3.162 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=379

$$\frac{2\left(Ad^2-Bcd+c^2C\right)\sqrt{a+b\tan(e+fx)}}{3f\left(c^2+d^2\right)(bc-ad)(c+d\tan(e+fx))^{3/2}}+\frac{2\sqrt{a+b\tan(e+fx)}\,\left(b\left(4c^2d^2(2A-C)+2Ad^4-5Bc^3d+Bcd^3+2\right.\right.}{3f\left(c^2+d^2\right)^2(bc-ad)^2\sqrt{c+d\tan(}}$$

```
[Out] -(B+I*(A-C))*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(c-I*d)^(5/2)/f/(a-I*b)^(1/2)+(I*A-B-I*C)*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(c+I
*d)^(5/2)/f/(a+I*b)^(1/2)+2/3*(b*(2*c^4*C-5*B*c^3*d+4*c^2*(2*A-C)*d^2+B*c*d
^3+2*A*d^4)-3*a*d^2*(2*c*(A-C)*d-B*(c^2-d^2)))*(a+b*tan(f*x+e))^(1/2)/(-a*d
+b*c)^2/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)+2/3*(A*d^2-B*c*d+C*c^2)*(a+b*t
an(f*x+e))^(1/2)/(-a*d+b*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] time = 1.81, antiderivative size = 379, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 5, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used = {3649, 3616, 3615, 93, 208}

$$\frac{2\left(Ad^2-Bcd+c^2C\right)\sqrt{a+b\tan(e+fx)}}{3f\left(c^2+d^2\right)(bc-ad)(c+d\tan(e+fx))^{3/2}}+\frac{2\sqrt{a+b\tan(e+fx)}\,\left(b\left(4c^2d^2(2A-C)+2Ad^4-5Bc^3d+Bcd^3+2\right.\right.}{3f\left(c^2+d^2\right)^2(bc-ad)^2\sqrt{c+d\tan(}}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]*(c +
d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] -(((B + I*(A - C))*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a - I*b]*(c - I*d)^(5/2)*f)) + ((
I*A - B - I*C)*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I
*b]*Sqrt[c + d*Tan[e + f*x]])]])/(Sqrt[a + I*b]*(c + I*d)^(5/2)*f) + (2*(c^2
*C - B*c*d + A*d^2)*Sqrt[a + b*Tan[e + f*x]])/(3*(b*c - a*d)*(c^2 + d^2)*f*
(c + d*Tan[e + f*x])^(3/2)) + (2*(b*(2*c^4*C - 5*B*c^3*d + 4*c^2*(2*A - C)*
d^2 + B*c*d^3 + 2*A*d^4) - 3*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)))*Sqrt[a
+ b*Tan[e + f*x]])/(3*(b*c - a*d)^2*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]
])
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
```

```
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{5/2}}\,dx = \frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{3(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + 2\int \frac{\frac{1}{2}\left(2Abd^2+3Ac(b\ldots\right.}{\ldots}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{3(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\left(b\left(2c^4C-5Bc\ldots\right.\right.}{\ldots}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{3(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\left(b\left(2c^4C-5Bc\ldots\right.\right.}{\ldots}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{3(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\left(b\left(2c^4C-5Bc\ldots\right.\right.}{\ldots}$$

$$= \frac{2\left(c^2C-Bcd+Ad^2\right)\sqrt{a+b\tan(e+fx)}}{3(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^{3/2}} + \frac{2\left(b\left(2c^4C-5Bc\ldots\right.\right.}{\ldots}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\sqrt{c+d\tan(e+fx)}}\right)}{\sqrt{a-ib}\,(c-id)^{5/2}f} - \frac{(B-i(A-C))\,\mathrm{t}\ldots}{\sqrt{a\ldots}}$$

Mathematica [A] time = 5.59, size = 403, normalized size = 1.06

$$\frac{\frac{2\left(c^2+d^2\right)(bc-ad)\left(Ad^2-Bcd+c^2C\right)\sqrt{a+b\tan(e+fx)}}{(c+d\tan(e+fx))^{3/2}} + \frac{2\sqrt{a+b\tan(e+fx)}\left(3ad^2\left(2cd(C-A)+B\left(c^2-d^2\right)\right)+b\left(4c^2d^2(2A-C)+2Ad^4-5Bc^3d+Bcd^3+2c^4C\right)\right)}{\sqrt{c+d\tan(e+fx)}} \ldots}{3f\left(c^2+d^2\right)^2(bc-ad\ldots}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/(Sqrt[a + b*Tan[e + f*x]]
*(c + d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] (3*(b*c - a*d)^2*(((I*A + B - I*C)*(c + I*d)^2*ArcTanh[(Sqrt[-c + I*d]*Sqrt
[a + b*Tan[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a
+ I*b]*Sqrt[-c + I*d]) + (I*(A + I*B - C)*(c - I*d)^2*ArcTanh[(Sqrt[c + I*d
]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqr
t[a + I*b]*Sqrt[c + I*d])) + (2*(b*c - a*d)*(c^2 + d^2)*(c^2*C - B*c*d + A*
```

```
d^2)*Sqrt[a + b*Tan[e + f*x]])/(c + d*Tan[e + f*x])^(3/2) + (2*(b*(2*c^4*C
- 5*B*c^3*d + 4*c^2*(2*A - C)*d^2 + B*c*d^3 + 2*A*d^4) + 3*a*d^2*(2*c*(-A +
 C)*d + B*(c^2 - d^2)))*Sqrt[a + b*Tan[e + f*x]])/Sqrt[c + d*Tan[e + f*x]])
/(3*(b*c - a*d)^2*(c^2 + d^2)^2*f)
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)}{\sqrt{a+b\tan\left(fx+e\right)}\,\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f*x+e))
^(5/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(1/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(1/2)*(c
+ d*tan(e + f*x))^(5/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^{2}\left(e + fx\right)}{\sqrt{a + b\tan\left(e + fx\right)}\left(c + d\tan\left(e + fx\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(1/2)/(c+d*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(sqrt(a + b*tan(e + f*x))
*(c + d*tan(e + f*x))**(5/2)), x)
```

3.163 $\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx$

Optimal. Leaf size=651

$$-\frac{2d\sqrt{a+b\tan(e+fx)}\left(A\left(a^2d^2+b^2\left(3c^2+4d^2\right)\right)+a^2\left(-Bcd+4c^2C+3Cd^2\right)-3abB\left(c^2+d^2\right)+b^2c(cC-Bd}{3f\left(a^2+b^2\right)\left(c^2+d^2\right)(bc-ad)^2(c+d\tan(e+fx))^{3/2}}$$

```
[Out] -(I*A+B-I*C)*arctanh((c-I*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a-I*b)^(1/2)/(c+
d*tan(f*x+e))^(1/2))/(a-I*b)^(3/2)/(c-I*d)^(5/2)/f-(B-I*(A-C))*arctanh((c+I
*d)^(1/2)*(a+b*tan(f*x+e))^(1/2)/(a+I*b)^(1/2)/(c+d*tan(f*x+e))^(1/2))/(a+I
*b)^(3/2)/(c+I*d)^(5/2)/f-2/3*d*(b^3*c*(-8*B*c^2*d-2*B*d^3+5*C*c^3-C*c*d^2)
+a^2*b*(-8*B*c^3*d-2*B*c*d^3+8*C*c^4+5*C*c^2*d^2+3*C*d^4)+3*a^3*d^2*(2*c*C*
d+B*(c^2-d^2))+3*a*b^2*(2*c*C*d^3-B*(c^4+c^2*d^2+2*d^4))-A*(6*a^3*c*d^3+6*a
*b^2*c*d^3-a^2*b*d^2*(11*c^2+5*d^2)-b^3*(3*c^4+17*c^2*d^2+8*d^4)))*(a+b*tan
(f*x+e))^(1/2)/(a^2+b^2)/(-a*d+b*c)^3/(c^2+d^2)^2/f/(c+d*tan(f*x+e))^(1/2)-
2*(A*b^2-a*(B*b-C*a))/(a^2+b^2)/(-a*d+b*c)/f/(a+b*tan(f*x+e))^(1/2)/(c+d*ta
n(f*x+e))^(3/2)-2/3*d*(b^2*c*(-B*d+C*c)-3*a*b*B*(c^2+d^2)+a^2*(-B*c*d+4*C*c
^2+3*C*d^2)+A*(a^2*d^2+b^2*(3*c^2+4*d^2)))*(a+b*tan(f*x+e))^(1/2)/(a^2+b^2)
/(-a*d+b*c)^2/(c^2+d^2)/f/(c+d*tan(f*x+e))^(3/2)
```

Rubi [A] time = 3.43, antiderivative size = 650, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 5, integrand size = 49, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.102, Rules used = {3649, 3616, 3615, 93, 208}

$$-\frac{2d\sqrt{a+b\tan(e+fx)}\left(-A\left(-a^2bd^2\left(11c^2+5d^2\right)+6a^3cd^3+6ab^2cd^3+b^3\left(-\left(17c^2d^2+3c^4+8d^4\right)\right)\right)+a^2b\left(-8\right.}{3f\left(a^2\right.}$$

Antiderivative was successfully verified.

```
[In] Int[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/2)*(c
+ d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] -(((I*A + B - I*C)*ArcTanh[(Sqrt[c - I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a
 - I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a - I*b)^(3/2)*(c - I*d)^(5/2)*f)) -
((B - I*(A - C))*ArcTanh[(Sqrt[c + I*d]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a +
 I*b]*Sqrt[c + d*Tan[e + f*x]])])/((a + I*b)^(3/2)*(c + I*d)^(5/2)*f) - (2*
(A*b^2 - a*(b*B - a*C)))/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e + f*x]
]*(c + d*Tan[e + f*x])^(3/2)) - (2*d*(a^2*A*d^2 + b^2*c*(c*C - B*d) - 3*a*b
*B*(c^2 + d^2) + A*b^2*(3*c^2 + 4*d^2) + a^2*(4*c^2*C - B*c*d + 3*C*d^2))*S
qrt[a + b*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c - a*d)^2*(c^2 + d^2)*f*(c + d*
```

```
Tan[e + f*x])^(3/2)) - (2*d*(b^3*c*(5*c^3*C - 8*B*c^2*d - c*C*d^2 - 2*B*d^3
) + a^2*b*(8*c^4*C - 8*B*c^3*d + 5*c^2*C*d^2 - 2*B*c*d^3 + 3*C*d^4) + 3*a^3
*d^2*(2*c*C*d + B*(c^2 - d^2)) + 3*a*b^2*(2*c*C*d^3 - B*(c^4 + c^2*d^2 + 2*
d^4)) - A*(6*a^3*c*d^3 + 6*a*b^2*c*d^3 - a^2*b*d^2*(11*c^2 + 5*d^2) - b^3*(
3*c^4 + 17*c^2*d^2 + 8*d^4)))*Sqrt[a + b*Tan[e + f*x]])/(3*(a^2 + b^2)*(b*c
 - a*d)^3*(c^2 + d^2)^2*f*Sqrt[c + d*Tan[e + f*x]])
```

Rule 93

```
Int[(((a_.) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_))/((e_.) + (f_.)*(x
_)), x_Symbol] :> With[{q = Denominator[m]}, Dist[q, Subst[Int[x^(q*(m + 1)
 - 1)/(b*e - a*f - (d*e - c*f)*x^q), x], x, (a + b*x)^(1/q)/(c + d*x)^(1/q)
], x]] /; FreeQ[{a, b, c, d, e, f}, x] && EqQ[m + n + 1, 0] && RationalQ[n]
 && LtQ[-1, m, 0] && SimplerQ[a + b*x, c + d*x]
```

Rule 208

```
Int[((a_) + (b_.)*(x_)^2)^(-1), x_Symbol] :> Simp[(Rt[-(a/b), 2]*ArcTanh[x/
Rt[-(a/b), 2]])/a, x] /; FreeQ[{a, b}, x] && NegQ[a/b]
```

Rule 3615

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[A^2/f, Subst[Int[((a + b*x)^m*(c + d*x)^n)/(A - B*x), x], x, Tan[e + f*x
]], x] /; FreeQ[{a, b, c, d, e, f, A, B, m, n}, x] && NeQ[b*c - a*d, 0] &&
NeQ[a^2 + b^2, 0] && EqQ[A^2 + B^2, 0]
```

Rule 3616

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((A_.) + (B_.)*tan[(e_.) +
 (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_), x_Symbol] :> Di
st[(A + I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(1 - I*Ta
n[e + f*x]), x], x] + Dist[(A - I*B)/2, Int[(a + b*Tan[e + f*x])^m*(c + d*T
an[e + f*x])^n*(1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A,
B, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[A^2 + B^2, 0]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
```

```
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int \frac{A+B\tan(e+fx)+C\tan^2(e+fx)}{(a+b\tan(e+fx))^{3/2}(c+d\tan(e+fx))^{5/2}}\,dx = -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))}$$

$$= -\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))}$$

$$= -\frac{(iA+B-iC)\tanh^{-1}\left(\frac{\sqrt{c-id}\,\sqrt{a+b\tan(e+fx)}}{\sqrt{a-ib}\,\sqrt{c+d\tan(e+fx)}}\right)}{(a-ib)^{3/2}(c-id)^{5/2}f} - \frac{(B-i(A-C}{(a}$$

Mathematica [A] time = 6.99, size = 903, normalized size = 1.39

$$-\frac{2\left(Ab^2-a(bB-aC)\right)}{\left(a^2+b^2\right)(bc-ad)f\sqrt{a+b\tan(e+fx)}\,(c+d\tan(e+fx))^{3/2}}-\frac{2\left(-\frac{2\sqrt{a+b\tan(e+fx)}\left(\frac{1}{2}d^2\left(4Adb^2-aA(bc-ad)-(bB-aC)(bc+3a\right.\right.}{3(ad-bc)\left(c^2+d^2\right)f(c-}\right.}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2)/((a + b*Tan[e + f*x])^(3/
2)*(c + d*Tan[e + f*x])^(5/2)),x]
```

```
[Out] (-2*(A*b^2 - a*(b*B - a*C)))/((a^2 + b^2)*(b*c - a*d)*f*Sqrt[a + b*Tan[e +
f*x]]*(c + d*Tan[e + f*x])^(3/2)) - (2*((-2*(-(c*(-2*c*(A*b^2 - a*(b*B - a*
C))*d + ((A*b - a*B - b*C)*d*(b*c - a*d))/2)) + (d^2*(4*A*b^2*d - a*A*(b*c
- a*d) - (b*B - a*C)*(b*c + 3*a*d)))/2)*Sqrt[a + b*Tan[e + f*x]])/(3*(-(b*c
) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^(3/2)) - (2*((3*(b*c - a*d)^3*(
((a + I*b)*(I*A + B - I*C)*(c + I*d)^2*ArcTanh[(Sqrt[-c + I*d]*Sqrt[a + b*T
an[e + f*x]])/(Sqrt[-a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqrt[-a + I*b]*S
qrt[-c + I*d]) + ((I*a + b)*(A + I*B - C)*(c - I*d)^2*ArcTanh[(Sqrt[c + I*d
]*Sqrt[a + b*Tan[e + f*x]])/(Sqrt[a + I*b]*Sqrt[c + d*Tan[e + f*x]])])/(Sqr
t[a + I*b]*Sqrt[c + I*d])))/(4*(-(b*c) + a*d)*(c^2 + d^2)*f) - (2*(d^2*(((b
*c)/2 - (3*a*d)/2)*(-2*c*(A*b^2 - a*(b*B - a*C))*d + ((A*b - a*B - b*C)*d*(
b*c - a*d))/2) + ((b*d^2 - (3*c*(-(b*c) + a*d))/2)*(4*A*b^2*d - a*A*(b*c -
a*d) - (b*B - a*C)*(b*c + 3*a*d)))/2) - c*((3*d*(-(b*c) + a*d)*(-2*(A*b^2 -
 a*(b*B - a*C))*d^2 - (c*(A*b - a*B - b*C)*(b*c - a*d))/2 + (d*(4*A*b^2*d -
 a*A*(b*c - a*d) - (b*B - a*C)*(b*c + 3*a*d)))/2))/2 - b*c*(-(c*(-2*c*(A*b^
2 - a*(b*B - a*C))*d + ((A*b - a*B - b*C)*d*(b*c - a*d))/2)) + (d^2*(4*A*b^
2*d - a*A*(b*c - a*d) - (b*B - a*C)*(b*c + 3*a*d)))/2)))*Sqrt[a + b*Tan[e +
 f*x]])/((-(b*c) + a*d)*(c^2 + d^2)*f*Sqrt[c + d*Tan[e + f*x]])))/(3*(-(b*c
) + a*d)*(c^2 + d^2))))/((a^2 + b^2)*(b*c - a*d))
```

fricas [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="fricas")
```

```
[Out] Timed out
```

giac [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="giac")
```

```
[Out] Timed out
```

maple [F(-1)] time = 180.00, size = 0, normalized size = 0.00

$$\int \frac{A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)}{\left(a+b\tan\left(fx+e\right)\right)^{\frac{3}{2}}\left(c+d\tan\left(fx+e\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f*x+e))
^(5/2),x)
```

```
[Out] int((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f*x+e))
^(5/2),x)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(a+b*tan(f*x+e))^(3/2)/(c+d*tan(f
*x+e))^(5/2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F(-1)] time = 0.00, size = -1, normalized size = -0.00

Hanged

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((A + B*tan(e + f*x) + C*tan(e + f*x)^2)/((a + b*tan(e + f*x))^(3/2)*(c
+ d*tan(e + f*x))^(5/2)),x)
```

```
[Out] \text{Hanged}
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{A + B\tan\left(e + fx\right) + C\tan^2\left(e + fx\right)}{\left(a + b\tan\left(e + fx\right)\right)^{\frac{3}{2}}\left(c + d\tan\left(e + fx\right)\right)^{\frac{5}{2}}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(a+b*tan(f*x+e))**(3/2)/(c+d*tan
(f*x+e))**(5/2),x)
```

```
[Out] Integral((A + B*tan(e + f*x) + C*tan(e + f*x)**2)/((a + b*tan(e + f*x))**(3
/2)*(c + d*tan(e + f*x))**(5/2)), x)
```

3.164 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n \left(A+B\tan(e+\right.$

Optimal. Leaf size=376

$$-\frac{(B+i(A-C))(a+b\tan(e+fx))^{m+1}(c+d\tan(e+fx))^n\left(\frac{b(c+d\tan(e+fx))}{bc-ad}\right)^{-n}F_1\left(m+1;-n,1;m+2;-\frac{d(a+b\tan(e}{bc-ad}\right.}{2f(m+1)(a-ib)}$$

```
[Out] -1/2*(B+I*(A-C))*AppellF1(1+m,1,-n,2+m,(a+b*tan(f*x+e))/(a-I*b),-d*(a+b*tan
(f*x+e))/(-a*d+b*c))*(a+b*tan(f*x+e))^(1+m)*(c+d*tan(f*x+e))^n/(a-I*b)/f/(1
+m)/((b*(c+d*tan(f*x+e))/(-a*d+b*c))^n)-1/2*(A+I*B-C)*AppellF1(1+m,1,-n,2+m
,(a+b*tan(f*x+e))/(a+I*b),-d*(a+b*tan(f*x+e))/(-a*d+b*c))*(a+b*tan(f*x+e))^
(1+m)*(c+d*tan(f*x+e))^n/(I*a-b)/f/(1+m)/((b*(c+d*tan(f*x+e))/(-a*d+b*c))^n
)+C*hypergeom([-n, 1+m],[2+m],-d*(a+b*tan(f*x+e))/(-a*d+b*c))*(a+b*tan(f*x+
e))^(1+m)*(c+d*tan(f*x+e))^n/b/f/(1+m)/((b*(c+d*tan(f*x+e))/(-a*d+b*c))^n)
```

Rubi [A] time = 0.90, antiderivative size = 376, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3655, 6725, 70, 69, 137, 136}

$$-\frac{(B+i(A-C))(a+b\tan(e+fx))^{m+1}(c+d\tan(e+fx))^n\left(\frac{b(c+d\tan(e+fx))}{bc-ad}\right)^{-n}F_1\left(m+1;-n,1;m+2;-\frac{d(a+b\tan(e}{bc-ad}\right.}{2f(m+1)(a-ib)}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] -((B + I*(A - C))*AppellF1[1 + m, -n, 1, 2 + m, -((d*(a + b*Tan[e + f*x]))/
(b*c - a*d)), (a + b*Tan[e + f*x])/(a - I*b)]*(a + b*Tan[e + f*x])^(1 + m)*
(c + d*Tan[e + f*x])^n)/(2*(a - I*b)*f*(1 + m)*((b*(c + d*Tan[e + f*x]))/(b
*c - a*d))^n) - ((A + I*B - C)*AppellF1[1 + m, -n, 1, 2 + m, -((d*(a + b*Ta
n[e + f*x]))/(b*c - a*d)), (a + b*Tan[e + f*x])/(a + I*b)]*(a + b*Tan[e + f
*x])^(1 + m)*(c + d*Tan[e + f*x])^n)/(2*(I*a - b)*f*(1 + m)*((b*(c + d*Tan[
e + f*x]))/(b*c - a*d))^n) + (C*Hypergeometric2F1[1 + m, -n, 2 + m, -((d*(a
 + b*Tan[e + f*x]))/(b*c - a*d))]*(a + b*Tan[e + f*x])^(1 + m)*(c + d*Tan[e
 + f*x])^n)/(b*f*(1 + m)*((b*(c + d*Tan[e + f*x]))/(b*c - a*d))^n)
```

Rule 69

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
a + b*x)^(m + 1)*Hypergeometric2F1[-n, m + 1, m + 2, -((d*(a + b*x))/(b*c -
 a*d))])/(b*(m + 1)*(b/(b*c - a*d))^n), x] /; FreeQ[{a, b, c, d, m, n}, x]
&& NeQ[b*c - a*d, 0] &&  !IntegerQ[m] &&  !IntegerQ[n] && GtQ[b/(b*c - a*d)
```

```
, 0] && (RationalQ[m] ||  !(RationalQ[n] && GtQ[-(d/(b*c - a*d)), 0]))
```

Rule 70

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Dist[(c
 + d*x)^FracPart[n]/((b/(b*c - a*d))^IntPart[n]*((b*(c + d*x))/(b*c - a*d))
^FracPart[n]), Int[(a + b*x)^m*Simp[(b*c)/(b*c - a*d) + (b*d*x)/(b*c - a*d)
, x]^n, x], x] /; FreeQ[{a, b, c, d, m, n}, x] && NeQ[b*c - a*d, 0] &&  !In
tegerQ[m] &&  !IntegerQ[n] && (RationalQ[m] ||  !SimplerQ[n + 1, m + 1])
```

Rule 136

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_)*((e_.) + (f_.)*(x_))
^(p_), x_Symbol] :> Simp[((b*e - a*f)^p*(a + b*x)^(m + 1)*AppellF1[m + 1, -
n, -p, m + 2, -((d*(a + b*x))/(b*c - a*d)), -((f*(a + b*x))/(b*e - a*f))])/
(b^(p + 1)*(m + 1)*(b/(b*c - a*d))^n), x] /; FreeQ[{a, b, c, d, e, f, m, n}
, x] &&  !IntegerQ[m] &&  !IntegerQ[n] && IntegerQ[p] && GtQ[b/(b*c - a*d),
 0] &&  !(GtQ[d/(d*a - c*b), 0] && SimplerQ[c + d*x, a + b*x])
```

Rule 137

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_.) + (d_.)*(x_))^(n_)*((e_.) + (f_.)*(x_))
^(p_), x_Symbol] :> Dist[(c + d*x)^FracPart[n]/((b/(b*c - a*d))^IntPart[n]*
((b*(c + d*x))/(b*c - a*d))^FracPart[n]), Int[(a + b*x)^m*((b*c)/(b*c - a*d
) + (b*d*x)/(b*c - a*d))^n*(e + f*x)^p, x], x] /; FreeQ[{a, b, c, d, e, f,
m, n}, x] &&  !IntegerQ[m] &&  !IntegerQ[n] && IntegerQ[p] &&  !GtQ[b/(b*c
- a*d), 0] &&  !SimplerQ[c + d*x, a + b*x]
```

Rule 3655

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> With[{ff = FreeFactors[Tan[e + f*x], x]}, D
ist[ff/f, Subst[Int[((a + b*ff*x)^m*(c + d*ff*x)^n*(A + B*ff*x + C*ff^2*x^2
))/(1 + ff^2*x^2), x], x, Tan[e + f*x]/ff], x]] /; FreeQ[{a, b, c, d, e, f,
 A, B, C, m, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 +
d^2, 0]
```

Rule 6725

```
Int[(u_)/((a_) + (b_.)*(x_)^(n_)), x_Symbol] :> With[{v = RationalFunctionE
xpand[u/(a + b*x^n), x]}, Int[v, x] /; SumQ[v]] /; FreeQ[{a, b}, x] && IGtQ
[n, 0]
```

Rubi steps

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{\text{Subst}\left(\int \frac{(a+bx)^m(c+dx)^n(A}{1+x^2}\right.}{}$$

$$= \frac{\text{Subst}\left(\int \left(C(a+bx)^m(c\right.\right.}{}$$

$$= \frac{(-B+i(A-C))\,\text{Subst}\left(\right.}{}$$

$$= \frac{\left((-B+i(A-C))(c+d\,\mathrm{t}\right.}{}$$

$$= -\frac{(B+i(A-C))F_1\left(1+\right.}{}$$

Mathematica [F] time = 25.50, size = 0, normalized size = 0.00

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^n\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx$$

Verification is Not applicable to the result.

```
[In] Integrate[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] Integrate[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^n*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2), x]
```

fricas [F] time = 1.70, size = 0, normalized size = 0.00

$$\text{integral}\left(\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^m\left(d\tan\left(fx+e\right)+c\right)^n,x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^n*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] integral((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m*(d*
tan(f*x + e) + c)^n, x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}\left(d\tan\left(fx+e\right)+c\right)^{n}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^n*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m*(d
*tan(f*x + e) + c)^n, x)
```

maple [F] time = 4.34, size = 0, normalized size = 0.00

$$\int \left(a+b\tan\left(fx+e\right)\right)^{m}\left(c+d\tan\left(fx+e\right)\right)^{n}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^n*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

```
[Out] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^n*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}\left(d\tan\left(fx+e\right)+c\right)^{n}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^n*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m*(d
*tan(f*x + e) + c)^n, x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int {\left(a+b\,\tan\left(e+f\,x\right)\right)}^m\,{\left(c+d\,\tan\left(e+f\,x\right)\right)}^n\,\left(C\,{\tan\left(e+f\,x\right)}^2+B\,\tan\left(e+f\,x\right)+A\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^n*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^n*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2), x)
```

sympy [**F(-2)**] time = 0.00, size = 0, normalized size = 0.00

Exception raised: HeuristicGCDFailed

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**m*(c+d*tan(f*x+e))**n*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
```

```
[Out] Exception raised: HeuristicGCDFailed
```

3.165 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\left(A+B\tan(e+f\right.$

Optimal. Leaf size=560

$$\frac{d\tan(e+fx)(a+b\tan(e+fx))^{m+1}\left(b^2d(m+3)(m+4)(d(A-C)+Bc)-2(bc-ad)(3aCd-b(Bd(m+4)+3cC\right.}{b^3f(m+2)(m+3)(m+4)}$$

```
[Out] (b*c*(2+m)*(b^2*d*(B*c+(A-C)*d)*(3+m)*(4+m)-2*(-a*d+b*c)*(3*a*C*d-b*(3*c*C+
B*d*(4+m))))+d*(b^3*(2*c*(A-C)*d+B*(c^2-d^2))*(2+m)*(3+m)*(4+m)-a*(b^2*d*(B
*c+(A-C)*d)*(3+m)*(4+m)-2*(-a*d+b*c)*(3*a*C*d-b*(3*c*C+B*d*(4+m))))))*(a+b*
tan(f*x+e))^(1+m)/b^4/f/(1+m)/(2+m)/(3+m)/(4+m)+1/2*(A-I*B-C)*(c-I*d)^3*hyp
ergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a
+b)/f/(1+m)-1/2*(A+I*B-C)*(c+I*d)^3*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e
))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a-b)/f/(1+m)+d*(b^2*d*(B*c+(A-C)*d)*(
3+m)*(4+m)-2*(-a*d+b*c)*(3*a*C*d-b*(3*c*C+B*d*(4+m))))*tan(f*x+e)*(a+b*tan(
f*x+e))^(1+m)/b^3/f/(2+m)/(3+m)/(4+m)-(3*a*C*d-b*(3*c*C+B*d*(4+m)))*(a+b*ta
n(f*x+e))^(1+m)*(c+d*tan(f*x+e))^2/b^2/f/(3+m)/(4+m)+C*(a+b*tan(f*x+e))^(1+
m)*(c+d*tan(f*x+e))^3/b/f/(4+m)
```

Rubi [A] time = 2.38, antiderivative size = 551, normalized size of antiderivative = 0.98, number of steps used = 9, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3647, 3637, 3630, 3539, 3537, 68}

$$\frac{(a+b\tan(e+fx))^{m+1}\left(d\left(b^3(m+2)(m+3)(m+4)\left(2cd(A-C)+B\left(c^2-d^2\right)\right)-a\left(2(bc-ad)(-3aCd+bBd(m\right.\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] ((b*c*(2 + m)*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b
*c*C - 3*a*C*d + b*B*d*(4 + m))) + d*(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))*(
2 + m)*(3 + m)*(4 + m) - a*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*
c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m)))))*(a + b*Tan[e + f*x])^(1 + m
))/(b^4*f*(1 + m)*(2 + m)*(3 + m)*(4 + m)) + ((A - I*B - C)*(c - I*d)^3*Hyp
ergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a - I*b)]*(a + b*Tan[
e + f*x])^(1 + m))/(2*(I*a + b)*f*(1 + m)) - ((A + I*B - C)*(c + I*d)^3*Hyp
ergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a + I*b)]*(a + b*Tan[
e + f*x])^(1 + m))/(2*(I*a - b)*f*(1 + m)) + (d*(b^2*d*(B*c + (A - C)*d)*(3
 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m)))*Tan[e +
f*x]*(a + b*Tan[e + f*x])^(1 + m))/(b^3*f*(2 + m)*(3 + m)*(4 + m)) + ((3*b*
c*C - 3*a*C*d + b*B*d*(4 + m))*(a + b*Tan[e + f*x])^(1 + m)*(c + d*Tan[e +
```

```
f*x])^2)/(b^2*f*(3 + m)*(4 + m)) + (C*(a + b*Tan[e + f*x])^(1 + m)*(c + d*T
an[e + f*x])^3)/(b*f*(4 + m))
```

Rule 68

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
b*c - a*d)^n*(a + b*x)^(m + 1)*Hypergeometric2F1[-n, m + 1, m + 2, -((d*(a
+ b*x))/(b*c - a*d))])/(b^(n + 1)*(m + 1)), x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^3\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)dx = \frac{C(a+b\tan(e+fx))^{1+m}(c \cdots}{bf(4+m \cdots}$$

$$= \frac{(3bcC-3aCd+bBd(4+ \cdots}{\cdots}$$

$$= \frac{d\left(b^2d(Bc+(A-C)d)(3 \cdots\right.}{\cdots}$$

$$= \frac{\left(bc(2+m)\left(b^2d(Bc+(A \cdots\right.\right.}{\cdots}$$

$$= \frac{\left(bc(2+m)\left(b^2d(Bc+(A \cdots\right.\right.}{\cdots}$$

$$= \frac{\left(bc(2+m)\left(b^2d(Bc+(A \cdots\right.\right.}{\cdots}$$

$$= \frac{\left(bc(2+m)\left(b^2d(Bc+(A \cdots\right.\right.}{\cdots}$$

Mathematica [B] time = 6.41, size = 1390, normalized size = 2.48

$$\frac{C(c+d\tan(e+fx))^3(a+b\tan(e+fx))^{m+1}}{bf(m+4)}+\frac{\frac{(3bcC-3adC+bBd(m+4))(c+d\tan(e+fx))^2(a+b\tan(e+fx))^{m+1}}{bf(m+3)}+\frac{d\left(d(Bc+(A-C)d)(m+3)(m \cdots\right.}{\cdots}}{\cdots}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^3*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(1 + m)*(c + d*Tan[e + f*x])^3)/(b*f*(4 + m)) + (((
3*b*c*C - 3*a*C*d + b*B*d*(4 + m))*(a + b*Tan[e + f*x])^(1 + m)*(c + d*Tan[
e + f*x])^2)/(b*f*(3 + m)) + ((d*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) +
 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m)))*Tan[e + f*x]*(a + b*Tan
[e + f*x])^(1 + m))/(b*f*(2 + m)) - (((-(b*c*(2 + m)*(b^2*d*(B*c + (A - C)*
d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m)))) +
d*(-(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))*(2 + m)*(3 + m)*(4 + m)) + a*(b^2*
d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*
B*d*(4 + m)))))*(a + b*Tan[e + f*x])^(1 + m))/(b*f*(1 + m)) + ((I/2)*(a*d*(
b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d
+ b*B*d*(4 + m))) + b*c*(2 + m)*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) +
2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m))) - b*c*(2 + m)*(-((2*a*d
+ b*c*(1 + m))*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m))) + b*c*(3 + m)*(A*b*c*(4
 + m) - C*(3*a*d + b*c*(1 + m)))) - d*(-(b^3*(2*c*(A - C)*d + B*(c^2 - d^2)
)*(2 + m)*(3 + m)*(4 + m)) + a*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2
*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m)))) - I*b*(2 + m)*(b^2*c*(2*
c*(A - C)*d + B*(c^2 - d^2))*(3 + m)*(4 + m) - d*(b^2*d*(B*c + (A - C)*d)*(
3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m))) + d*(-(
(2*a*d + b*c*(1 + m))*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m))) + b*c*(3 + m)*(A
*b*c*(4 + m) - C*(3*a*d + b*c*(1 + m))))))*Hypergeometric2F1[1, 1 + m, 2 +
m, ((-I)*a - I*b*Tan[e + f*x])/((-I)*a + b)]*(a + b*Tan[e + f*x])^(1 + m))/
((a + I*b)*f*(1 + m)) - ((I/2)*(a*d*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m
) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*d*(4 + m))) + b*c*(2 + m)*(b^2*d
*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B
*d*(4 + m))) - b*c*(2 + m)*(-((2*a*d + b*c*(1 + m))*(3*b*c*C - 3*a*C*d + b*
B*d*(4 + m))) + b*c*(3 + m)*(A*b*c*(4 + m) - C*(3*a*d + b*c*(1 + m)))) - d*
(-(b^3*(2*c*(A - C)*d + B*(c^2 - d^2))*(2 + m)*(3 + m)*(4 + m)) + a*(b^2*d*
(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*C - 3*a*C*d + b*B*
d*(4 + m)))) + I*b*(2 + m)*(b^2*c*(2*c*(A - C)*d + B*(c^2 - d^2))*(3 + m)*(
4 + m) - d*(b^2*d*(B*c + (A - C)*d)*(3 + m)*(4 + m) + 2*(b*c - a*d)*(3*b*c*
C - 3*a*C*d + b*B*d*(4 + m))) + d*(-((2*a*d + b*c*(1 + m))*(3*b*c*C - 3*a*C
*d + b*B*d*(4 + m))) + b*c*(3 + m)*(A*b*c*(4 + m) - C*(3*a*d + b*c*(1 + m))
))))*Hypergeometric2F1[1, 1 + m, 2 + m, -((I*a + I*b*Tan[e + f*x])/((-I)*a
- b))]*(a + b*Tan[e + f*x])^(1 + m))/((a - I*b)*f*(1 + m)))/(b*(2 + m)))/(b
*(3 + m)))/(b*(4 + m))
```

fricas [F] time = 0.91, size = 0, normalized size = 0.00

$$\text{integral}\left(\left(Cd^3\tan\left(fx+e\right)^5+\left(3\,Ccd^2+Bd^3\right)\tan\left(fx+e\right)^4+Ac^3+\left(3\,Cc^2d+3\,Bcd^2+Ad^3\right)\tan\left(fx+e\right)^3+\right.\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] integral((C*d^3*tan(f*x + e)^5 + (3*C*c*d^2 + B*d^3)*tan(f*x + e)^4 + A*c^3
 + (3*C*c^2*d + 3*B*c*d^2 + A*d^3)*tan(f*x + e)^3 + (C*c^3 + 3*B*c^2*d + 3*
A*c*d^2)*tan(f*x + e)^2 + (B*c^3 + 3*A*c^2*d)*tan(f*x + e))*(b*tan(f*x + e)
 + a)^m, x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^{2} + B\tan\left(fx+e\right) + A\right)\left(d\tan\left(fx+e\right)+c\right)^{3}\left(b\tan\left(fx+e\right)+a\right)^{m}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)^3*(b
*tan(f*x + e) + a)^m, x)
```

maple [F] time = 2.62, size = 0, normalized size = 0.00

$$\int \left(a+b\tan\left(fx+e\right)\right)^{m}\left(c+d\tan\left(fx+e\right)\right)^{3}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

```
[Out] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^3*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int {\left(a+b\,\tan\left(e+f\,x\right)\right)}^m\,{\left(c+d\,\tan\left(e+f\,x\right)\right)}^3\,\left(C\,{\tan\left(e+f\,x\right)}^2+B\,\tan\left(e+f\,x\right)+A\right) \,d x$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^3*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a + b \tan \left(e + fx\right)\right)^{m} \left(c + d \tan \left(e + fx\right)\right)^{3} \left(A + B \tan \left(e + fx\right) + C \tan^{2} \left(e + fx\right)\right)\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**m*(c+d*tan(f*x+e))**3*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**m*(c + d*tan(e + f*x))**3*(A + B*tan(e + f*x
) + C*tan(e + f*x)**2), x)
```

3.166 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2\left(A+B\tan(e+f\right.$

Optimal. Leaf size=363

$$\frac{(a+b\tan(e+fx))^{m+1}\left(2a^2Cd^2-abd(m+3)(Bd+2cC)+b^2(m+2)\left(d^2(m+3)(A-C)+2Bcd(m+3)+2c^2C\right)\right.}{b^3f(m+1)(m+2)(m+3)}$$

```
[Out] (2*a^2*C*d^2-a*b*d*(B*d+2*C*c)*(3+m)+b^2*(2+m)*(2*c^2*C+2*B*c*d*(3+m)+(A-C)
*d^2*(3+m)))*(a+b*tan(f*x+e))^(1+m)/b^3/f/(1+m)/(2+m)/(3+m)+1/2*(A-I*B-C)*(
c-I*d)^2*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f*x+e)
)^(1+m)/(I*a+b)/f/(1+m)+1/2*(I*A-B-I*C)*(c+I*d)^2*hypergeom([1, 1+m],[2+m],
(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(a+I*b)/f/(1+m)-d*(2*a*C*d
-b*(2*c*C+B*d*(3+m)))*tan(f*x+e)*(a+b*tan(f*x+e))^(1+m)/b^2/f/(2+m)/(3+m)+C
*(a+b*tan(f*x+e))^(1+m)*(c+d*tan(f*x+e))^2/b/f/(3+m)
```

Rubi [A] time = 1.15, antiderivative size = 360, normalized size of antiderivative = 0.99, number of steps used = 8, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3647, 3637, 3630, 3539, 3537, 68}

$$\frac{(a+b\tan(e+fx))^{m+1}\left(2a^2Cd^2-abd(m+3)(Bd+2cC)+b^2(m+2)\left(d^2(m+3)(A-C)+2Bcd(m+3)+2c^2C\right)\right.}{b^3f(m+1)(m+2)(m+3)}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x] + C*T
an[e + f*x]^2),x]
```

```
[Out] ((2*a^2*C*d^2 - a*b*d*(2*c*C + B*d)*(3 + m) + b^2*(2 + m)*(2*c^2*C + 2*B*c*
d*(3 + m) + (A - C)*d^2*(3 + m)))*(a + b*Tan[e + f*x])^(1 + m))/(b^3*f*(1 +
 m)*(2 + m)*(3 + m)) + ((A - I*B - C)*(c - I*d)^2*Hypergeometric2F1[1, 1 +
m, 2 + m, (a + b*Tan[e + f*x])/(a - I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*
(I*a + b)*f*(1 + m)) + ((I*A - B - I*C)*(c + I*d)^2*Hypergeometric2F1[1, 1
+ m, 2 + m, (a + b*Tan[e + f*x])/(a + I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(
2*(a + I*b)*f*(1 + m)) + (d*(2*b*c*C - 2*a*C*d + b*B*d*(3 + m))*Tan[e + f*x
]*(a + b*Tan[e + f*x])^(1 + m))/(b^2*f*(2 + m)*(3 + m)) + (C*(a + b*Tan[e +
 f*x])^(1 + m)*(c + d*Tan[e + f*x])^2)/(b*f*(3 + m))
```

Rule 68

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
b*c - a*d)^n*(a + b*x)^(m + 1)*Hypergeometric2F1[-n, m + 1, m + 2, -((d*(a
+ b*x))/(b*c - a*d))])/(b^(n + 1)*(m + 1)), x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rule 3647

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.
) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a + b*Tan[e + f*x])^m*(c + d*Tan[
e + f*x])^(n + 1))/(d*f*(m + n + 1)), x] + Dist[1/(d*(m + n + 1)), Int[(a +
 b*Tan[e + f*x])^(m - 1)*(c + d*Tan[e + f*x])^n*Simp[a*A*d*(m + n + 1) - C*
(b*c*m + a*d*(n + 1)) + d*(A*b + a*B - b*C)*(m + n + 1)*Tan[e + f*x] - (C*m
*(b*c - a*d) - b*B*d*(m + n + 1))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] &&
NeQ[c^2 + d^2, 0] && GtQ[m, 0] &&  !(IGtQ[n, 0] && ( !IntegerQ[m] || (EqQ[c
```

```
, 0] && NeQ[a, 0])))
```

Rubi steps

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))^2\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{C(a+b\tan(e+fx))^{1+m}(c}{bf(3+m}$$

$$= \frac{d(2bcC-2aCd+bBd(3+}{b^2}$$

$$= \frac{\left(2a^2Cd^2-abd(2cC+Bd)\right.}{}$$

$$= \frac{\left(2a^2Cd^2-abd(2cC+Bd)\right.}{}$$

$$= \frac{\left(2a^2Cd^2-abd(2cC+Bd)\right.}{}$$

$$= \frac{\left(2a^2Cd^2-abd(2cC+Bd)\right.}{}$$

Mathematica [A] time = 6.35, size = 505, normalized size = 1.39

$$\frac{C(c+d\tan(e+fx))^2(a+b\tan(e+fx))^{m+1}}{bf(m+3)}+\frac{\frac{d\tan(e+fx)(-2aCd+bBd(m+3)+2bcC)(a+b\tan(e+fx))^{m+1}}{bf(m+2)}-\frac{i(a+b\tan(e+fx))^{m+1}\left(-\left(b^2(m\right.\right.}{}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])^2*(A + B*Tan[e + f*x]
 + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(1 + m)*(c + d*Tan[e + f*x])^2)/(b*f*(3 + m)) + ((d
*(2*b*c*C - 2*a*C*d + b*B*d*(3 + m))*Tan[e + f*x]*(a + b*Tan[e + f*x])^(1 +
 m))/(b*f*(2 + m)) - (((-(b*c*(2 + m)*(2*b*c*C - 2*a*C*d + b*B*d*(3 + m)))
- d*(b^2*(B*c + (A - C)*d)*(2 + m)*(3 + m) - a*(2*b*c*C - 2*a*C*d + b*B*d*(
3 + m))))*(a + b*Tan[e + f*x])^(1 + m))/(b*f*(1 + m)) + ((I/2)*(-(b^2*(A*c^
2 - c^2*C - 2*B*c*d - A*d^2 + C*d^2)*(2 + m)*(3 + m)) - I*b^2*(2*c*(A - C)*
d + B*(c^2 - d^2))*(2 + m)*(3 + m))*Hypergeometric2F1[1, 1 + m, 2 + m, ((-I
)*a - I*b*Tan[e + f*x])/((-I)*a + b)]*(a + b*Tan[e + f*x])^(1 + m))/((a + I
*b)*f*(1 + m)) - ((I/2)*(-(b^2*(A*c^2 - c^2*C - 2*B*c*d - A*d^2 + C*d^2)*(2
```

```
 + m)*(3 + m)) + I*b^2*(2*c*(A - C)*d + B*(c^2 - d^2))*(2 + m)*(3 + m))*Hyp
ergeometric2F1[1, 1 + m, 2 + m, -((I*a + I*b*Tan[e + f*x])/((-I)*a - b))]*(
a + b*Tan[e + f*x])^(1 + m))/((a - I*b)*f*(1 + m)))/(b*(2 + m)))/(b*(3 + m)
)
```

fricas [F] time = 0.65, size = 0, normalized size = 0.00

$$\text{integral}\left(\left(Cd^2\tan\left(fx+e\right)^4+\left(2\,Ccd+Bd^2\right)\tan\left(fx+e\right)^3+Ac^2+\left(Cc^2+2\,Bcd+Ad^2\right)\tan\left(fx+e\right)^2+\left(Bc^2\right.\right.\right.$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="fricas")
```

```
[Out] integral((C*d^2*tan(f*x + e)^4 + (2*C*c*d + B*d^2)*tan(f*x + e)^3 + A*c^2 +
 (C*c^2 + 2*B*c*d + A*d^2)*tan(f*x + e)^2 + (B*c^2 + 2*A*c*d)*tan(f*x + e))
*(b*tan(f*x + e) + a)^m, x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\left(d\tan\left(fx+e\right)+c\right)^2\left(b\tan\left(fx+e\right)+a\right)^m\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)^2*(b
*tan(f*x + e) + a)^m, x)
```

maple [F] time = 2.19, size = 0, normalized size = 0.00

$$\int\left(a+b\tan\left(fx+e\right)\right)^m\left(c+d\tan\left(fx+e\right)\right)^2\left(A+B\tan\left(fx+e\right)+C\left(\tan^2\left(fx+e\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

```
[Out] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x
)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))^2*(A+B*tan(f*x+e)+C*tan(f*x+e
)^2),x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \left(a+b\tan \left(e+f\,x\right)\right)^m\,\left(c+d\tan \left(e+f\,x\right)\right)^2\,\left(C\tan \left(e+f\,x\right)^2+B\tan \left(e+f\,x\right)+A\right) \,d x$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))^2*(A + B*tan(e + f*x) + C*t
an(e + f*x)^2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a + b \tan{\left(e + f x \right)}\right)^{m} \left(c + d \tan{\left(e + f x \right)}\right)^{2} \left(A + B \tan{\left(e + f x \right)} + C \tan^{2}{\left(e + f x \right)}\right)\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**m*(c+d*tan(f*x+e))**2*(A+B*tan(f*x+e)+C*tan(f*x
+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**m*(c + d*tan(e + f*x))**2*(A + B*tan(e + f*x
) + C*tan(e + f*x)**2), x)
```

3.167 $\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(A+B\tan(e+f$

Optimal. Leaf size=247

$$\frac{(c-id)(A-iB-C)(a+b\tan(e+fx))^{m+1}\,{}_2F_1\left(1,m+1;m+2;\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(b+ia)}-\frac{(c+id)(A+iB-C)(a+b\tan(e}{2f(}$$

[Out] -(a*C*d-b*(B*d+C*c)*(2+m))*(a+b*tan(f*x+e))^(1+m)/b^2/f/(1+m)/(2+m)+1/2*(A-I*B-C)*(c-I*d)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a+b)/f/(1+m)-1/2*(A+I*B-C)*(c+I*d)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a-b)/f/(1+m)+C*d*tan(f*x+e)*(a+b*tan(f*x+e))^(1+m)/b/f/(2+m)

Rubi [A] time = 0.53, antiderivative size = 247, normalized size of antiderivative = 1.00, number of steps used = 7, number of rules used = 5, integrand size = 43, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.116, Rules used = {3637, 3630, 3539, 3537, 68}

$$\frac{(c-id)(A-iB-C)(a+b\tan(e+fx))^{m+1}\,{}_2F_1\left(1,m+1;m+2;\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(b+ia)}-\frac{(c+id)(A+iB-C)(a+b\tan(e}{2f(}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] + C*Tan
[e + f*x]^2),x]
```

```
[Out] -(((a*C*d - b*(c*C + B*d)*(2 + m))*(a + b*Tan[e + f*x])^(1 + m))/(b^2*f*(1
+ m)*(2 + m))) + ((A - I*B - C)*(c - I*d)*Hypergeometric2F1[1, 1 + m, 2 + m
, (a + b*Tan[e + f*x])/(a - I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(I*a + b
)*f*(1 + m)) - ((A + I*B - C)*(c + I*d)*Hypergeometric2F1[1, 1 + m, 2 + m,
(a + b*Tan[e + f*x])/(a + I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(I*a - b)*
f*(1 + m)) + (C*d*Tan[e + f*x]*(a + b*Tan[e + f*x])^(1 + m))/(b*f*(2 + m))
```

Rule 68

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
b*c - a*d)^n*(a + b*x)^(m + 1)*Hypergeometric2F1[-n, m + 1, m + 2, -((d*(a
+ b*x))/(b*c - a*d))])/(b^(n + 1)*(m + 1)), x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
```

```
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rule 3637

```
Int[((a_) + (b_.)*tan[(e_.) + (f_.)*(x_)])*((c_.) + (d_.)*tan[(e_.) + (f_.)
*(x_)])^(n_.)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.) + (f
_.)*(x_)]^2), x_Symbol] :> Simp[(b*C*Tan[e + f*x]*(c + d*Tan[e + f*x])^(n +
 1))/(d*f*(n + 2)), x] - Dist[1/(d*(n + 2)), Int[(c + d*Tan[e + f*x])^n*Sim
p[b*c*C - a*A*d*(n + 2) - (A*b + a*B - b*C)*d*(n + 2)*Tan[e + f*x] - (a*C*d
*(n + 2) - b*(c*C - B*d*(n + 2)))*Tan[e + f*x]^2, x], x], x] /; FreeQ[{a, b
, c, d, e, f, A, B, C, n}, x] && NeQ[b*c - a*d, 0] && NeQ[c^2 + d^2, 0] &&
 !LtQ[n, -1]
```

Rubi steps

$$\int (a+b\tan(e+fx))^m(c+d\tan(e+fx))\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)\,dx = \frac{Cd\tan(e+fx)(a+b\tan}{bf(2+m)}$$
$$= -\frac{(aCd-b(cC+Bd)(2+}{b^2f(1+}$$
$$= -\frac{(aCd-b(cC+Bd)(2+}{b^2f(1+}$$
$$= -\frac{(aCd-b(cC+Bd)(2+}{b^2f(1+}$$
$$= -\frac{(aCd-b(cC+Bd)(2+}{b^2f(1+}$$

Mathematica [A] time = 3.00, size = 202, normalized size = 0.82

$$\frac{(a+b\tan(e+fx))^{m+1}\left(-\frac{ib(m+2)(c-id)(A-iB-C)\,{}_2F_1\left(1,m+1;m+2;\frac{a+b\tan(e+fx)}{a-ib}\right)}{(m+1)(a-ib)}+\frac{ib(m+2)(c+id)(A+iB-C)\,{}_2F_1\left(1,m+1;m+2;\frac{a+b\tan(e+fx)}{a+ib}\right.}{(m+1)(a+ib)}}{2bf(m+2)}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^m*(c + d*Tan[e + f*x])*(A + B*Tan[e + f*x] +
 C*Tan[e + f*x]^2),x]
```

```
[Out] ((a + b*Tan[e + f*x])^(1 + m)*((-2*a*C*d + 2*b*(c*C + B*d)*(2 + m))/(b*(1 +
 m)) - (I*b*(A - I*B - C)*(c - I*d)*(2 + m)*Hypergeometric2F1[1, 1 + m, 2 +
 m, (a + b*Tan[e + f*x])/(a - I*b)])/((a - I*b)*(1 + m)) + (I*b*(A + I*B -
C)*(c + I*d)*(2 + m)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x]
)/(a + I*b)])/((a + I*b)*(1 + m)) + 2*C*d*Tan[e + f*x]))/(2*b*f*(2 + m))
```

fricas [F] time = 1.01, size = 0, normalized size = 0.00

$$\text{integral}\left(\left(Cd\tan\left(fx+e\right)^3+(Cc+Bd)\tan\left(fx+e\right)^2+Ac+(Bc+Ad)\tan\left(fx+e\right)\right)\left(b\tan\left(fx+e\right)+a\right)^m,x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="fricas")
```

```
[Out] integral((C*d*tan(f*x + e)^3 + (C*c + B*d)*tan(f*x + e)^2 + A*c + (B*c + A*
d)*tan(f*x + e))*(b*tan(f*x + e) + a)^m, x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(d\tan\left(fx+e\right)+c\right)\left(b\tan\left(fx+e\right)+a\right)^{m}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)*(b*t
an(f*x + e) + a)^m, x)
```

maple [F] time = 1.74, size = 0, normalized size = 0.00

$$\int \left(a+b\tan\left(fx+e\right)\right)^{m}\left(c+d\tan\left(fx+e\right)\right)\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] int((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(d\tan\left(fx+e\right)+c\right)\left(b\tan\left(fx+e\right)+a\right)^{m}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)^
2),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(d*tan(f*x + e) + c)*(b*t
an(f*x + e) + a)^m, x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int {\left(a+b\,\mathrm{tan}\left(e+f\,x\right)\right)}^m\,\left(c+d\,\mathrm{tan}\left(e+f\,x\right)\right)\,\left(C\,{\mathrm{tan}\left(e+f\,x\right)}^2+B\,\mathrm{tan}\left(e+f\,x\right)+A\right) \,d x$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^m*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) + C*tan
(e + f*x)^2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a+b\tan\left(e+fx\right)\right)^m \left(c+d\tan\left(e+fx\right)\right)\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**m*(c+d*tan(f*x+e))*(A+B*tan(f*x+e)+C*tan(f*x+e)
**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**m*(c + d*tan(e + f*x))*(A + B*tan(e + f*x) +
 C*tan(e + f*x)**2), x)
```

3.168 $\int (a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e+f\right.$

Optimal. Leaf size=178

$$\frac{(A-iB-C)(a+b\tan(e+fx))^{m+1}\,{}_2F_1\left(1,m+1;m+2;\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(b+ia)}+\frac{(iA-B-iC)(a+b\tan(e+fx))^{m+1}\,{}_2F_1\left(\right.}{2f(m+1)(a+i}$$

```
[Out] C*(a+b*tan(f*x+e))^(1+m)/b/f/(1+m)+1/2*(A-I*B-C)*hypergeom([1, 1+m],[2+m],(
a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a+b)/f/(1+m)+1/2*(I*A-B-
I*C)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1
+m)/(a+I*b)/f/(1+m)
```

Rubi [A] time = 0.18, antiderivative size = 178, normalized size of antiderivative = 1.00, number of steps used = 6, number of rules used = 4, integrand size = 33, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.121, Rules used = {3630, 3539, 3537, 68}

$$\frac{(A-iB-C)(a+b\tan(e+fx))^{m+1}\,{}_2F_1\left(1,m+1;m+2;\frac{a+b\tan(e+fx)}{a-ib}\right)}{2f(m+1)(b+ia)}+\frac{(iA-B-iC)(a+b\tan(e+fx))^{m+1}\,{}_2F_1\left(\right.}{2f(m+1)(a+i}$$

Antiderivative was successfully verified.

```
[In] Int[(a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (C*(a + b*Tan[e + f*x])^(1 + m))/(b*f*(1 + m)) + ((A - I*B - C)*Hypergeomet
ric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a - I*b)]*(a + b*Tan[e + f*x]
)^(1 + m))/(2*(I*a + b)*f*(1 + m)) + ((I*A - B - I*C)*Hypergeometric2F1[1,
1 + m, 2 + m, (a + b*Tan[e + f*x])/(a + I*b)]*(a + b*Tan[e + f*x])^(1 + m))
/(2*(a + I*b)*f*(1 + m))
```

Rule 68

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
b*c - a*d)^n*(a + b*x)^(m + 1)*Hypergeometric2F1[-n, m + 1, m + 2, -((d*(a
+ b*x))/(b*c - a*d))])/(b^(n + 1)*(m + 1)), x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3630

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :> Simp[(C*(a +
 b*Tan[e + f*x])^(m + 1))/(b*f*(m + 1)), x] + Int[(a + b*Tan[e + f*x])^m*Si
mp[A - C + B*Tan[e + f*x], x], x] /; FreeQ[{a, b, e, f, A, B, C, m}, x] &&
NeQ[A*b^2 - a*b*B + a^2*C, 0] &&  !LeQ[m, -1]
```

Rubi steps

$$\int (a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right) dx = \frac{C(a+b\tan(e+fx))^{1+m}}{bf(1+m)} + \int (a+b\tan(e+$$

$$= \frac{C(a+b\tan(e+fx))^{1+m}}{bf(1+m)} + \frac{1}{2}(A-iB-C)\int$$

$$= \frac{C(a+b\tan(e+fx))^{1+m}}{bf(1+m)} + \frac{(iA+B-iC)\,\mathrm{Su}}{}$$

$$= \frac{C(a+b\tan(e+fx))^{1+m}}{bf(1+m)} - \frac{(iA+B-iC)\,{}_2F}{}$$

Mathematica [A] time = 0.25, size = 135, normalized size = 0.76

$$\frac{(a+b\tan(e+fx))^{m+1}\left(-\frac{i(A-iB-C)\,{}_2F_1\left(1,m+1;m+2;\frac{a+b\tan(e+fx)}{a-ib}\right)}{a-ib}+\frac{i(A+iB-C)\,{}_2F_1\left(1,m+1;m+2;\frac{a+b\tan(e+fx)}{a+ib}\right)}{a+ib}+\frac{2C}{b}\right)}{2f(m+1)}$$

Antiderivative was successfully verified.

```
[In] Integrate[(a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2),x]
```

```
[Out] (((2*C)/b - (I*(A - I*B - C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[
e + f*x])/(a - I*b)])/(a - I*b) + (I*(A + I*B - C)*Hypergeometric2F1[1, 1 +
 m, 2 + m, (a + b*Tan[e + f*x])/(a + I*b)])/(a + I*b))*(a + b*Tan[e + f*x])
^(1 + m))/(2*f*(1 + m))
```

fricas [F] time = 1.15, size = 0, normalized size = 0.00

$$\text{integral}\left(\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m},x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
fricas")
```

```
[Out] integral((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m, x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m, x
)
```

maple [F] time = 1.37, size = 0, normalized size = 0.00

$$\int\left(a+b\tan\left(fx+e\right)\right)^{m}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

```
[Out] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2),x, algorithm="
maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m, x
)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.01

$$\int \left(a+b\tan\left(e+f\,x\right)\right)^m\,\left(C\tan\left(e+f\,x\right)^2+B\tan\left(e+f\,x\right)+A\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2),x)
```

```
[Out] int((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \left(a+b\tan\left(e+fx\right)\right)^m\left(A+B\tan\left(e+fx\right)+C\tan^2\left(e+fx\right)\right)\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**m*(A+B*tan(f*x+e)+C*tan(f*x+e)**2),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**m*(A + B*tan(e + f*x) + C*tan(e + f*x)**2),
x)
```

3.169 $\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx$

Optimal. Leaf size=258

$$\frac{\left(Ad^2-Bcd+c^2C\right)(a+b\tan(e+fx))^{m+1}\,{}_2F_1\left(1,m+1;m+2;-\frac{d(a+b\tan(e+fx))}{bc-ad}\right)}{f(m+1)\left(c^2+d^2\right)(bc-ad)}-\frac{(iA+B-iC)(a+b\tan(e+fx)}{2f(m+1}$$

```
[Out] -1/2*(I*A+B-I*C)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*ta
n(f*x+e))^(1+m)/(a-I*b)/(c-I*d)/f/(1+m)-1/2*(A+I*B-C)*hypergeom([1, 1+m],[2
+m],(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(I*a-b)/(c+I*d)/f/(1+m
)+(A*d^2-B*c*d+C*c^2)*hypergeom([1, 1+m],[2+m],-d*(a+b*tan(f*x+e))/(-a*d+b*
c))*(a+b*tan(f*x+e))^(1+m)/(-a*d+b*c)/(c^2+d^2)/f/(1+m)
```

Rubi [A] time = 0.48, antiderivative size = 258, normalized size of antiderivative = 1.00, number of steps used = 8, number of rules used = 5, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.111, Rules used = {3653, 3539, 3537, 68, 3634}

$$\frac{\left(Ad^2-Bcd+c^2C\right)(a+b\tan(e+fx))^{m+1}\,{}_2F_1\left(1,m+1;m+2;-\frac{d(a+b\tan(e+fx))}{bc-ad}\right)}{f(m+1)\left(c^2+d^2\right)(bc-ad)}-\frac{(iA+B-iC)(a+b\tan(e+fx)}{2f(m+1}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x]),x]
```

```
[Out] -((I*A + B - I*C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(
a - I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(a - I*b)*(c - I*d)*f*(1 + m)) -
 ((A + I*B - C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a
+ I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(I*a - b)*(c + I*d)*f*(1 + m)) + (
(c^2*C - B*c*d + A*d^2)*Hypergeometric2F1[1, 1 + m, 2 + m, -((d*(a + b*Tan[
e + f*x]))/(b*c - a*d))]*(a + b*Tan[e + f*x])^(1 + m))/((b*c - a*d)*(c^2 +
d^2)*f*(1 + m))
```

Rule 68

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
b*c - a*d)^n*(a + b*x)^(m + 1)*Hypergeometric2F1[-n, m + 1, m + 2, -((d*(a
+ b*x))/(b*c - a*d))])/(b^(n + 1)*(m + 1)), x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{c+d\tan(e+fx)}\,dx = \frac{\int(a+b\tan(e+fx))^m(Ac-cC+Bd+(Bc-}{c^2+d^2}$$

$$= \frac{(A-iB-C)\int(1+i\tan(e+fx))(a+b\tan(e}{2(c-id)}$$

$$= \frac{\left(c^2C-Bcd+Ad^2\right)\,{}_2F_1\left(1,1+m;2+m;-\frac{d(a}{}\right.}{(bc-ad)\left(c^2+d^2\right)}$$

$$= -\frac{(iA+B-iC)\,{}_2F_1\left(1,1+m;2+m;\frac{a+b\tan(e+}{a-ib}\right.}{2(a-ib)(c-id)f(1+}$$

Mathematica [A] time = 1.12, size = 204, normalized size = 0.79

$$\frac{(a+b\tan(e+fx))^{m+1}\left(\frac{2\left(Ad^2-Bcd+c^2C\right){}_2F_1\left(1,m+1;m+2;\frac{d(a+b\tan(e+fx))}{ad-bc}\right)}{bc-ad}+\frac{(d-ic)(A-iB-C)\,{}_2F_1\left(1,m+1;m+2;\frac{a+b\tan(e+fx)}{a-ib}\right)}{a-ib}+\frac{(d+ic)(A+}{}\right.}{2f(m+1)\left(c^2+d^2\right)}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x]),x]
```

```
[Out] ((((A - I*B - C)*((-I)*c + d)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan
[e + f*x])/(a - I*b)])/(a - I*b) + ((A + I*B - C)*(I*c + d)*Hypergeometric2
F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a + I*b)])/(a + I*b) + (2*(c^2*C
- B*c*d + A*d^2)*Hypergeometric2F1[1, 1 + m, 2 + m, (d*(a + b*Tan[e + f*x])
)/(-(b*c) + a*d)])/(b*c - a*d))*(a + b*Tan[e + f*x])^(1 + m))/(2*(c^2 + d^2
)*f*(1 + m))
```

fricas [F] time = 0.74, size = 0, normalized size = 0.00

$$\text{integral}\left(\frac{\left(C\tan\left(fx+e\right)^2+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^m}{d\tan\left(fx+e\right)+c},x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="fricas")
```

```
[Out] integral((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d*
tan(f*x + e) + c), x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}}{d\tan\left(fx+e\right)+c}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d
*tan(f*x + e) + c), x)
```

maple [F] time = 4.92, size = 0, normalized size = 0.00

$$\int \frac{\left(a+b\tan\left(fx+e\right)\right)^{m}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{c+d\tan\left(fx+e\right)}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x)
```

```
[Out] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e)),x)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}}{d\tan\left(fx+e\right)+c}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
)),x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d
*tan(f*x + e) + c), x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{{\left(a+b\tan\left(e+f\,x\right)\right)}^{m}\,\left(C\,{\tan\left(e+f\,x\right)}^{2}+B\,\tan\left(e+f\,x\right)+A\right)}{c+d\,\tan\left(e+f\,x\right)}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x)),x)
```

```
[Out] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x)), x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{(a + b\tan(e + fx))^m \left(A + B\tan(e + fx) + C\tan^2(e + fx)\right)}{c + d\tan(e + fx)}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**m*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e)),x)
```

```
[Out] Integral((a + b*tan(e + f*x))**m*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x)), x)
```

3.170 $\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx$

Optimal. Leaf size=403

$$-\frac{(a+b\tan(e+fx))^{m+1}\left(ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)-b\left(Ad^2\left(c^2(2-m)-d^2m\right)-Bcd\left(c^2(1-m)-d^2(m+\right.\right.\right.}{f(m+1)\left(c^2+d^2\right)^2(bc-ad)^2}$$

```
[Out] 1/2*(A-I*B-C)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f
*x+e))^(1+m)/(I*a+b)/(c-I*d)^2/f/(1+m)+1/2*(I*A-B-I*C)*hypergeom([1, 1+m],[
2+m],(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(a+I*b)/(c+I*d)^2/f/(
1+m)-(a*d^2*(2*c*(A-C)*d-B*(c^2-d^2))-b*(A*d^2*(c^2*(2-m)-d^2*m)-B*c*d*(c^2
*(1-m)-d^2*(1+m))-c^2*C*(c^2*m+d^2*(2+m))))*hypergeom([1, 1+m],[2+m],-d*(a+
b*tan(f*x+e))/(-a*d+b*c))*(a+b*tan(f*x+e))^(1+m)/(-a*d+b*c)^2/(c^2+d^2)^2/f
/(1+m)+(A*d^2-B*c*d+C*c^2)*(a+b*tan(f*x+e))^(1+m)/(-a*d+b*c)/(c^2+d^2)/f/(c
+d*tan(f*x+e))
```

Rubi [A] time = 1.22, antiderivative size = 402, normalized size of antiderivative = 1.00, number of steps used = 9, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}}$ = 0.133, Rules used = {3649, 3653, 3539, 3537, 68, 3634}

$$-\frac{(a+b\tan(e+fx))^{m+1}\left(ad^2\left(2cd(A-C)-B\left(c^2-d^2\right)\right)-b\left(Ac^2d^2(2-m)-Ad^4m-B\left(c^3d(1-m)-cd^3(m+\right.\right.\right.}{f(m+1)\left(c^2+d^2\right)^2(bc-ad)^2}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^2,x]
```

```
[Out] ((A - I*B - C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a -
 I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(I*a + b)*(c - I*d)^2*f*(1 + m)) +
((I*A - B - I*C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a
 + I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(a + I*b)*(c + I*d)^2*f*(1 + m))
- ((a*d^2*(2*c*(A - C)*d - B*(c^2 - d^2)) - b*(A*c^2*d^2*(2 - m) - c^4*C*m
- A*d^4*m - c^2*C*d^2*(2 + m) - B*(c^3*d*(1 - m) - c*d^3*(1 + m))))*Hyperge
ometric2F1[1, 1 + m, 2 + m, -((d*(a + b*Tan[e + f*x]))/(b*c - a*d))]*(a + b
*Tan[e + f*x])^(1 + m))/((b*c - a*d)^2*(c^2 + d^2)^2*f*(1 + m)) + ((c^2*C -
 B*c*d + A*d^2)*(a + b*Tan[e + f*x])^(1 + m))/((b*c - a*d)*(c^2 + d^2)*f*(c
 + d*Tan[e + f*x]))
```

Rule 68

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
b*c - a*d)^n*(a + b*x)^(m + 1)*Hypergeometric2F1[-n, m + 1, m + 2, -((d*(a
```

```
+ b*x))/(b*c - a*d))])/(b^(n + 1)*(m + 1)), x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
```

```
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^m \left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^2}\,dx = \frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{1+m}}{(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))}$$

$$= \frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{1+m}}{(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))}$$

$$= \frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{1+m}}{(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))}$$

$$= -\frac{\left(ad^2\left(2c(A-C)d-B\left(c^2-d^2\right)\right)-b\left(Ac^2d\right.\right.}{}$$

$$= -\frac{(iA+B-iC)\,{}_2F_1\left(1,1+m;2+m;\frac{a+b\tan(}{a-ib}\right.}{2(a-ib)(c-id)^2f(1}$$

Mathematica [A] time = 6.20, size = 563, normalized size = 1.40

$$-\frac{\left(Ad^2-c(Bd-cC)\right)(a+b\tan(e+fx))^{m+1}}{f\left(c^2+d^2\right)(ad-bc)(c+d\tan(e+fx))}-\frac{-\frac{(a+b\tan(e+fx))^{m+1}\left(d^2\left((cC-Bd)(ad-bc(m+1))-A\left(acd-b\left(c^2-d^2m\right)\right)\right)-cd(bc-ad)(Bc-\right.}{f(m+1)\left(c^2+d^2\right)(ad-b}}{}$$

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^2,x]
```

```
[Out] -(((A*d^2 - c*(-(c*C) + B*d))*(a + b*Tan[e + f*x])^(1 + m))/((-(b*c) + a*d)
*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))) - (-(((-(c*d*(b*c - a*d)*(B*c - (A -
C)*d)) - b*c^2*(c^2*C - B*c*d + A*d^2)*m + d^2*((c*C - B*d)*(a*d - b*c*(1 +
 m)) - A*(a*c*d - b*(c^2 - d^2*m))))*Hypergeometric2F1[1, 1 + m, 2 + m, (d*
```

```
(a + b*Tan[e + f*x]))/(-(b*c) + a*d)]*(a + b*Tan[e + f*x])^(1 + m))/((-(b*c
) + a*d)*(c^2 + d^2)*f*(1 + m))) + (((I/2)*(-((b*c - a*d)*(c^2*C - 2*B*c*d
- C*d^2 - A*(c^2 - d^2))) - I*(b*c - a*d)*(2*c*(A - C)*d - B*(c^2 - d^2)))*
Hypergeometric2F1[1, 1 + m, 2 + m, ((-I)*a - I*b*Tan[e + f*x])/((-I)*a + b)
]*(a + b*Tan[e + f*x])^(1 + m))/((a + I*b)*f*(1 + m)) - ((I/2)*(-((b*c - a*
d)*(c^2*C - 2*B*c*d - C*d^2 - A*(c^2 - d^2))) + I*(b*c - a*d)*(2*c*(A - C)*
d - B*(c^2 - d^2)))*Hypergeometric2F1[1, 1 + m, 2 + m, -((I*a + I*b*Tan[e +
 f*x])/((-I)*a - b))]*(a + b*Tan[e + f*x])^(1 + m))/((a - I*b)*f*(1 + m)))/
(c^2 + d^2))/((-(b*c) + a*d)*(c^2 + d^2))
```

fricas [F] time = 1.55, size = 0, normalized size = 0.00

$$\text{integral}\left(\frac{\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}}{d^{2}\tan\left(fx+e\right)^{2}+2\,cd\tan\left(fx+e\right)+c^{2}},x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="fricas")
```

```
[Out] integral((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d^
2*tan(f*x + e)^2 + 2*c*d*tan(f*x + e) + c^2), x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\frac{\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}}{\left(d\tan\left(fx+e\right)+c\right)^{2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d
*tan(f*x + e) + c)^2, x)
```

maple [F] time = 4.56, size = 0, normalized size = 0.00

$$\int\frac{\left(a+b\tan\left(fx+e\right)\right)^{m}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(c+d\tan\left(fx+e\right)\right)^{2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x
)
```

```
[Out] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^2,x
)
```

maxima [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}}{\left(d\tan\left(fx+e\right)+c\right)^{2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^2,x, algorithm="maxima")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d
*tan(f*x + e) + c)^2, x)
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{\left(a+b\tan\left(e+f\,x\right)\right)^{m}\,\left(C\tan\left(e+f\,x\right)^{2}+B\tan\left(e+f\,x\right)+A\right)}{\left(c+d\tan\left(e+f\,x\right)\right)^{2}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^2,x)
```

```
[Out] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^2, x)
```

sympy [F(-2)] time = 0.00, size = 0, normalized size = 0.00

Exception raised: HeuristicGCDFailed

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**m*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**2,x)
```

```
[Out] Exception raised: HeuristicGCDFailed
```

3.171 $\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx$

Optimal. Leaf size=702

$$\frac{(a+b\tan(e+fx))^{m+1}\left(2a^2d^3\left(d(A-C)\left(3c^2-d^2\right)-B\left(c^3-3cd^2\right)\right)-2abd^2\left(2cd(A-C)\left(c^2(3-m)-d^2(m+1)\right)\right.\right.}{}$$

```
[Out] 1/2*(A-I*B-C)*hypergeom([1, 1+m],[2+m],(a+b*tan(f*x+e))/(a-I*b))*(a+b*tan(f
*x+e))^(1+m)/(I*a+b)/(c-I*d)^3/f/(1+m)+1/2*(A+I*B-C)*hypergeom([1, 1+m],[2+
m],(a+b*tan(f*x+e))/(a+I*b))*(a+b*tan(f*x+e))^(1+m)/(a+I*b)/(I*c-d)^3/f/(1+
m)+1/2*(2*a^2*d^3*((A-C)*d*(3*c^2-d^2)-B*(c^3-3*c*d^2))-2*a*b*d^2*(B*(6*c^2
*d^2-c^4*(2-m)-d^4*m)+2*c*(A-C)*d*(c^2*(3-m)-d^2*(1+m)))-b^2*(A*d^2*(d^4*(1
-m)*m+2*c^2*d^2*(-m^2+3*m+1)-c^4*(m^2-5*m+6))+B*c*d*(d^4*m*(1+m)-2*c^2*d^2*
(-m^2+m+3)+c^4*(m^2-3*m+2))+c^2*C*(c^4*(1-m)*m+2*c^2*d^2*(-m^2-m+3)-d^4*(m^
2+3*m+2))))*hypergeom([1, 1+m],[2+m],-d*(a+b*tan(f*x+e))/(-a*d+b*c))*(a+b*t
an(f*x+e))^(1+m)/(-a*d+b*c)^3/(c^2+d^2)^3/f/(1+m)+1/2*(A*d^2-B*c*d+C*c^2)*(
a+b*tan(f*x+e))^(1+m)/(-a*d+b*c)/(c^2+d^2)/f/(c+d*tan(f*x+e))^2-1/2*(2*a*d^
2*(2*c*(A-C)*d-B*(c^2-d^2))-b*(c^4*C*(1-m)+A*d^4*(1-m)-B*c^3*d*(3-m)+B*c*d^
3*(1+m)+c^2*d^2*(A*(5-m)-C*(3+m))))*(a+b*tan(f*x+e))^(1+m)/(-a*d+b*c)^2/(c^
2+d^2)^2/f/(c+d*tan(f*x+e))
```

Rubi [A] time = 2.94, antiderivative size = 702, normalized size of antiderivative = 1.00, number of steps used = 10, number of rules used = 6, integrand size = 45, $\frac{\text{number of rules}}{\text{integrand size}} = 0.133$, Rules used = {3649, 3653, 3539, 3537, 68, 3634}

$$\frac{(a+b\tan(e+fx))^{m+1}\left(2a^2d^3\left(d(A-C)\left(3c^2-d^2\right)-B\left(c^3-3cd^2\right)\right)-2abd^2\left(2cd(A-C)\left(c^2(3-m)-d^2(m+1)\right)\right.\right.}{}$$

Antiderivative was successfully verified.

```
[In] Int[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/(c + d
*Tan[e + f*x])^3,x]
```

```
[Out] ((A - I*B - C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a -
 I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(I*a + b)*(c - I*d)^3*f*(1 + m)) +
((A + I*B - C)*Hypergeometric2F1[1, 1 + m, 2 + m, (a + b*Tan[e + f*x])/(a +
 I*b)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(a + I*b)*(I*c - d)^3*f*(1 + m)) +
((2*a^2*d^3*((A - C)*d*(3*c^2 - d^2) - B*(c^3 - 3*c*d^2)) - 2*a*b*d^2*(B*(6
*c^2*d^2 - c^4*(2 - m) - d^4*m) + 2*c*(A - C)*d*(c^2*(3 - m) - d^2*(1 + m))
) - b^2*(A*d^2*(d^4*(1 - m)*m + 2*c^2*d^2*(1 + 3*m - m^2) - c^4*(6 - 5*m +
m^2)) + B*(c*d^5*m*(1 + m) - 2*c^3*d^3*(3 + m - m^2) + c^5*d*(2 - 3*m + m^2
```

```
)) + c^2*C*(c^4*(1 - m)*m + 2*c^2*d^2*(3 - m - m^2) - d^4*(2 + 3*m + m^2)))
)*Hypergeometric2F1[1, 1 + m, 2 + m, -((d*(a + b*Tan[e + f*x]))/(b*c - a*d)
)]*(a + b*Tan[e + f*x])^(1 + m))/(2*(b*c - a*d)^3*(c^2 + d^2)^3*f*(1 + m))
+ ((c^2*C - B*c*d + A*d^2)*(a + b*Tan[e + f*x])^(1 + m))/(2*(b*c - a*d)*(c^
2 + d^2)*f*(c + d*Tan[e + f*x])^2) - ((2*a*d^2*(2*c*(A - C)*d - B*(c^2 - d^
2)) - b*(c^4*C*(1 - m) + A*d^4*(1 - m) - B*c^3*d*(3 - m) + B*c*d^3*(1 + m)
+ c^2*d^2*(A*(5 - m) - C*(3 + m))))*(a + b*Tan[e + f*x])^(1 + m))/(2*(b*c -
 a*d)^2*(c^2 + d^2)^2*f*(c + d*Tan[e + f*x]))
```

Rule 68

```
Int[((a_) + (b_.)*(x_))^(m_)*((c_) + (d_.)*(x_))^(n_), x_Symbol] :> Simp[((
b*c - a*d)^n*(a + b*x)^(m + 1)*Hypergeometric2F1[-n, m + 1, m + 2, -((d*(a
+ b*x))/(b*c - a*d))])/(b^(n + 1)*(m + 1)), x] /; FreeQ[{a, b, c, d, m}, x]
 && NeQ[b*c - a*d, 0] &&  !IntegerQ[m] && IntegerQ[n]
```

Rule 3537

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_) + (d_.)*tan[(e_.) +
(f_.)*(x_)]), x_Symbol] :> Dist[(c*d)/f, Subst[Int[(a + (b*x)/d)^m/(d^2 + c
*x), x], x, d*Tan[e + f*x]], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b
*c - a*d, 0] && NeQ[a^2 + b^2, 0] && EqQ[c^2 + d^2, 0]
```

Rule 3539

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)]), x_Symbol] :> Dist[(c + I*d)/2, Int[(a + b*Tan[e + f*x])^m*(1
 - I*Tan[e + f*x]), x], x] + Dist[(c - I*d)/2, Int[(a + b*Tan[e + f*x])^m*(
1 + I*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, m}, x] && NeQ[b*c -
 a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&  !IntegerQ[m]
```

Rule 3634

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_.)*((c_.) + (d_.)*tan[(e_.)
+ (f_.)*(x_)])^(n_.)*((A_) + (C_.)*tan[(e_.) + (f_.)*(x_)]^2), x_Symbol] :>
 Dist[A/f, Subst[Int[(a + b*x)^m*(c + d*x)^n, x], x, Tan[e + f*x]], x] /; F
reeQ[{a, b, c, d, e, f, A, C, m, n}, x] && EqQ[A, C]
```

Rule 3649

```
Int[((a_.) + (b_.)*tan[(e_.) + (f_.)*(x_)])^(m_)*((c_.) + (d_.)*tan[(e_.) +
 (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.) + (f_.)*(x_)] + (C_.)*tan[(e_.)
 + (f_.)*(x_)]^2), x_Symbol] :> Simp[((A*b^2 - a*(b*B - a*C))*(a + b*Tan[e
+ f*x])^(m + 1)*(c + d*Tan[e + f*x])^(n + 1))/(f*(m + 1)*(b*c - a*d)*(a^2 +
 b^2)), x] + Dist[1/((m + 1)*(b*c - a*d)*(a^2 + b^2)), Int[(a + b*Tan[e + f
*x])^(m + 1)*(c + d*Tan[e + f*x])^n*Simp[A*(a*(b*c - a*d)*(m + 1) - b^2*d*(
```

```
m + n + 2)) + (b*B - a*C)*(b*c*(m + 1) + a*d*(n + 1)) - (m + 1)*(b*c - a*d)
*(A*b - a*B - b*C)*Tan[e + f*x] - d*(A*b^2 - a*(b*B - a*C))*(m + n + 2)*Tan
[e + f*x]^2, x], x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C, n}, x] && NeQ[
b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] && LtQ[m, -1] &&  !
(ILtQ[n, -1] && ( !IntegerQ[m] || (EqQ[c, 0] && NeQ[a, 0])))
```

Rule 3653

```
Int[(((c_.) + (d_.)*tan[(e_.) + (f_.)*(x_)])^(n_)*((A_.) + (B_.)*tan[(e_.)
+ (f_.)*(x_)] + (C_.)*tan[(e_.) + (f_.)*(x_)]^2))/((a_.) + (b_.)*tan[(e_.)
+ (f_.)*(x_)]), x_Symbol] :> Dist[1/(a^2 + b^2), Int[(c + d*Tan[e + f*x])^n
*Simp[b*B + a*(A - C) + (a*B - b*(A - C))*Tan[e + f*x], x], x], x] + Dist[(
A*b^2 - a*b*B + a^2*C)/(a^2 + b^2), Int[((c + d*Tan[e + f*x])^n*(1 + Tan[e
+ f*x]^2))/(a + b*Tan[e + f*x]), x], x] /; FreeQ[{a, b, c, d, e, f, A, B, C
, n}, x] && NeQ[b*c - a*d, 0] && NeQ[a^2 + b^2, 0] && NeQ[c^2 + d^2, 0] &&
 !GtQ[n, 0] &&  !LeQ[n, -1]
```

Rubi steps

$$\int \frac{(a+b\tan(e+fx))^m\left(A+B\tan(e+fx)+C\tan^2(e+fx)\right)}{(c+d\tan(e+fx))^3}\,dx = \frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{1+m}}{2(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} +$$

$$= \frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{1+m}}{2(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} -$$

$$= \frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{1+m}}{2(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} -$$

$$= \frac{\left(c^2C-Bcd+Ad^2\right)(a+b\tan(e+fx))^{1+m}}{2(bc-ad)\left(c^2+d^2\right)f(c+d\tan(e+fx))^2} -$$

$$= \frac{\left(2a^2d^3\left((A-C)d\left(3c^2-d^2\right)-B\left(c^3-3cd^2\right)\right)\right.}{}$$

$$= -\frac{(A-iB-C)\,{}_2F_1\left(1,1+m;2+m;\frac{a+b\tan(e+f}{a-ib}\right.}{2(a-ib)(ic+d)^3f(1-}$$

Mathematica [B] time = 6.24, size = 2238, normalized size = 3.19

Result too large to show

Antiderivative was successfully verified.

```
[In] Integrate[((a + b*Tan[e + f*x])^m*(A + B*Tan[e + f*x] + C*Tan[e + f*x]^2))/
(c + d*Tan[e + f*x])^3,x]
```

```
[Out] -1/2*((A*d^2 - c*(-(c*C) + B*d))*(a + b*Tan[e + f*x])^(1 + m))/((-(b*c) + a
*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x])^2) - (-(((-(c*(2*d*(b*c - a*d)*(B*c
- (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))) + d^2*(A*(2*c*(b*c - a
*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m))))*(a + b*Tan[e + f
*x])^(1 + m))/((-(b*c) + a*d)*(c^2 + d^2)*f*(c + d*Tan[e + f*x]))) - (-(((-
(c*d*(-(b*c) + a*d)*(-2*c*(b*c - a*d)*(B*c - (A - C)*d) - b*d*(c^2*C - B*c*
d + A*d^2)*(1 - m) + d*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(
2*a*d - b*c*(1 + m))))) - b*c^2*m*(-(c*(2*d*(b*c - a*d)*(B*c - (A - C)*d) -
 b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))) + d^2*(A*(2*c*(b*c - a*d) + b*d^2*(1
 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m)))) + d^2*((2*d*(b*c - a*d)*(B*c -
 (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))*(-(a*d) + b*c*(1 + m)) +
 (-(c*(-(b*c) + a*d)) - b*d^2*m)*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*
C - B*d)*(2*a*d - b*c*(1 + m)))))*Hypergeometric2F1[1, 1 + m, 2 + m, (d*(a
+ b*Tan[e + f*x]))/(-(b*c) + a*d)]*(a + b*Tan[e + f*x])^(1 + m))/((-(b*c) +
 a*d)*(c^2 + d^2)*f*(1 + m))) + (((I/2)*(d*(-(b*c) + a*d)*(-2*c*(b*c - a*d)
*(B*c - (A - C)*d) - b*d*(c^2*C - B*c*d + A*d^2)*(1 - m) + d*(A*(2*c*(b*c -
 a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m)))) + c*((2*d*(b*c
 - a*d)*(B*c - (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))*(-(a*d) +
b*c*(1 + m)) + (-(c*(-(b*c) + a*d)) - b*d^2*m)*(A*(2*c*(b*c - a*d) + b*d^2*
(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m))) + b*m*(-(c*(2*d*(b*c - a*d)*(
B*c - (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))) + d^2*(A*(2*c*(b*c
 - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m))))) + I*(c*(-(b
*c) + a*d)*(-2*c*(b*c - a*d)*(B*c - (A - C)*d) - b*d*(c^2*C - B*c*d + A*d^2
)*(1 - m) + d*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b
*c*(1 + m)))) - d*((2*d*(b*c - a*d)*(B*c - (A - C)*d) - b*c*(c^2*C - B*c*d
+ A*d^2)*(1 - m))*(-(a*d) + b*c*(1 + m)) + (-(c*(-(b*c) + a*d)) - b*d^2*m)*
(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m))) +
 b*m*(-(c*(2*d*(b*c - a*d)*(B*c - (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*
(1 - m))) + d^2*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d -
 b*c*(1 + m)))))))*Hypergeometric2F1[1, 1 + m, 2 + m, ((-I)*a - I*b*Tan[e +
 f*x])/((-I)*a + b)]*(a + b*Tan[e + f*x])^(1 + m))/((a + I*b)*f*(1 + m)) -
((I/2)*(d*(-(b*c) + a*d)*(-2*c*(b*c - a*d)*(B*c - (A - C)*d) - b*d*(c^2*C -
 B*c*d + A*d^2)*(1 - m) + d*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B
*d)*(2*a*d - b*c*(1 + m)))) + c*((2*d*(b*c - a*d)*(B*c - (A - C)*d) - b*c*(
c^2*C - B*c*d + A*d^2)*(1 - m))*(-(a*d) + b*c*(1 + m)) + (-(c*(-(b*c) + a*d
)) - b*d^2*m)*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b
```

```
*c*(1 + m))) + b*m*(-(c*(2*d*(b*c - a*d)*(B*c - (A - C)*d) - b*c*(c^2*C - B
*c*d + A*d^2)*(1 - m))) + d^2*(A*(2*c*(b*c - a*d) + b*d^2*(1 - m)) + (c*C -
 B*d)*(2*a*d - b*c*(1 + m))))) - I*(c*(-(b*c) + a*d)*(-2*c*(b*c - a*d)*(B*c
 - (A - C)*d) - b*d*(c^2*C - B*c*d + A*d^2)*(1 - m) + d*(A*(2*c*(b*c - a*d)
 + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m)))) - d*((2*d*(b*c - a*
d)*(B*c - (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))*(-(a*d) + b*c*(
1 + m)) + (-(c*(-(b*c) + a*d)) - b*d^2*m)*(A*(2*c*(b*c - a*d) + b*d^2*(1 -
m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m))) + b*m*(-(c*(2*d*(b*c - a*d)*(B*c -
 (A - C)*d) - b*c*(c^2*C - B*c*d + A*d^2)*(1 - m))) + d^2*(A*(2*c*(b*c - a*
d) + b*d^2*(1 - m)) + (c*C - B*d)*(2*a*d - b*c*(1 + m)))))))*Hypergeometric
2F1[1, 1 + m, 2 + m, -((I*a + I*b*Tan[e + f*x])/((-I)*a - b))]*(a + b*Tan[e
 + f*x])^(1 + m))/((a - I*b)*f*(1 + m)))/(c^2 + d^2))/((-(b*c) + a*d)*(c^2
+ d^2)))/(2*(-(b*c) + a*d)*(c^2 + d^2))
```

fricas [F] time = 1.74, size = 0, normalized size = 0.00

$$\text{integral}\left(\frac{\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}}{d^{3}\tan\left(fx+e\right)^{3}+3\,cd^{2}\tan\left(fx+e\right)^{2}+3\,c^{2}d\tan\left(fx+e\right)+c^{3}},x\right)$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="fricas")
```

```
[Out] integral((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d^
3*tan(f*x + e)^3 + 3*c*d^2*tan(f*x + e)^2 + 3*c^2*d*tan(f*x + e) + c^3), x)
```

giac [F] time = 0.00, size = 0, normalized size = 0.00

$$\int\frac{\left(C\tan\left(fx+e\right)^{2}+B\tan\left(fx+e\right)+A\right)\left(b\tan\left(fx+e\right)+a\right)^{m}}{\left(d\tan\left(fx+e\right)+c\right)^{3}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="giac")
```

```
[Out] integrate((C*tan(f*x + e)^2 + B*tan(f*x + e) + A)*(b*tan(f*x + e) + a)^m/(d
*tan(f*x + e) + c)^3, x)
```

maple [F] time = 5.04, size = 0, normalized size = 0.00

$$\int\frac{\left(a+b\tan\left(fx+e\right)\right)^{m}\left(A+B\tan\left(fx+e\right)+C\left(\tan^{2}\left(fx+e\right)\right)\right)}{\left(c+d\tan\left(fx+e\right)\right)^{3}}\,dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x
)
```

```
[Out] int((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e))^3,x
)
```

maxima [F(-1)] time = 0.00, size = 0, normalized size = 0.00

Timed out

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))^m*(A+B*tan(f*x+e)+C*tan(f*x+e)^2)/(c+d*tan(f*x+e
))^3,x, algorithm="maxima")
```

```
[Out] Timed out
```

mupad [F] time = 0.00, size = -1, normalized size = -0.00

$$\int \frac{\left(a+b\tan\left(e+f\,x\right)\right)^m\,\left(C\,{\tan\left(e+f\,x\right)}^2+B\,\tan\left(e+f\,x\right)+A\right)}{{\left(c+d\,\tan\left(e+f\,x\right)\right)}^3} \,d x$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^3,x)
```

```
[Out] int(((a + b*tan(e + f*x))^m*(A + B*tan(e + f*x) + C*tan(e + f*x)^2))/(c + d
*tan(e + f*x))^3, x)
```

sympy [F] time = 0.00, size = 0, normalized size = 0.00

$$\int \frac{\left(a + b \tan{\left(e + f x \right)}\right)^{m} \left(A + B \tan{\left(e + f x \right)} + C \tan^{2}{\left(e + f x \right)}\right)}{\left(c + d \tan{\left(e + f x \right)}\right)^{3}}\, dx$$

Verification of antiderivative is not currently implemented for this CAS.

```
[In] integrate((a+b*tan(f*x+e))**m*(A+B*tan(f*x+e)+C*tan(f*x+e)**2)/(c+d*tan(f*x
+e))**3,x)
```

```
[Out] Integral((a + b*tan(e + f*x))**m*(A + B*tan(e + f*x) + C*tan(e + f*x)**2)/(
c + d*tan(e + f*x))**3, x)
```

Chapter 4

Listing of Grading functions

The following are the current version of the grading functions used for grading the quality of the antiderivative with reference to the optimal antiderivative included in the test suite.

There is a version for Maple and for Mathematica/Rubi. There is a version for grading Sympy and version for use with Sagemath.

The following are links to the current source code.

The following are the listings of source code of the grading functions.

4.0.1 Mathematica and Rubi grading function

```
(* Original version thanks to Albert Rich emailed on 03/21/2017 *)
(* ::Package:: *)

(* ::Subsection:: *)
(*GradeAntiderivative[result,optimal]*)


(* ::Text:: *)
(*If result and optimal are mathematical expressions, *)
(*         GradeAntiderivative[result,optimal] returns*)
(*  "F" if the result fails to integrate an expression that*)
(*      is integrable*)
(*  "C" if result involves higher level functions than necessary*)
(*  "B" if result is more than twice the size of the optimal*)
(*      antiderivative*)
(*  "A" if result can be considered optimal*)


GradeAntiderivative[result_,optimal_] :=
  If[ExpnType[result]<=ExpnType[optimal],
```

```
    If[FreeQ[result,Complex] || Not[FreeQ[optimal,Complex]],
      If[LeafCount[result]<=2*LeafCount[optimal],
        "A",
      "B"],
    "C"],
  If[FreeQ[result,Integrate] && FreeQ[result,Int],
    "C",
  "F"]]


(* ::Text:: *)
(*The following summarizes the type number assigned an *)
(*expression based on the functions it involves*)
(*1 = rational function*)
(*2 = algebraic function*)
(*3 = elementary function*)
(*4 = special function*)
(*5 = hyperpergeometric function*)
(*6 = appell function*)
(*7 = rootsum function*)
(*8 = integrate function*)
(*9 = unknown function*)


ExpnType[expn_] :=
  If[AtomQ[expn],
    1,
  If[ListQ[expn],
    Max[Map[ExpnType,expn]],
  If[Head[expn]===Power,
    If[IntegerQ[expn[[2]]],
      ExpnType[expn[[1]]],
    If[Head[expn[[2]]]===Rational,
      If[IntegerQ[expn[[1]]] || Head[expn[[1]]]===Rational,
        1,
      Max[ExpnType[expn[[1]]],2]],
    Max[ExpnType[expn[[1]]],ExpnType[expn[[2]]],3]]],
  If[Head[expn]===Plus || Head[expn]===Times,
    Max[ExpnType[First[expn]],ExpnType[Rest[expn]]],
  If[ElementaryFunctionQ[Head[expn]],
    Max[3,ExpnType[expn[[1]]]],
  If[SpecialFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],4]],
  If[HypergeometricFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],5]],
  If[AppellFunctionQ[Head[expn]],
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],6]],
```

```
  If[Head[expn]===RootSum,
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],7]],
  If[Head[expn]===Integrate || Head[expn]===Int,
    Apply[Max,Append[Map[ExpnType,Apply[List,expn]],8]],
  9]]]]]]]]]]


ElementaryFunctionQ[func_] :=
  MemberQ[{
        Exp,Log,
        Sin,Cos,Tan,Cot,Sec,Csc,
        ArcSin,ArcCos,ArcTan,ArcCot,ArcSec,ArcCsc,
        Sinh,Cosh,Tanh,Coth,Sech,Csch,
        ArcSinh,ArcCosh,ArcTanh,ArcCoth,ArcSech,ArcCsch
},func]


SpecialFunctionQ[func_] :=
  MemberQ[{
        Erf, Erfc, Erfi,
        FresnelS, FresnelC,
        ExpIntegralE, ExpIntegralEi, LogIntegral,
        SinIntegral, CosIntegral, SinhIntegral, CoshIntegral,
        Gamma, LogGamma, PolyGamma,
        Zeta, PolyLog, ProductLog,
        EllipticF, EllipticE, EllipticPi
},func]


HypergeometricFunctionQ[func_] :=
  MemberQ[{Hypergeometric1F1,Hypergeometric2F1,HypergeometricPFQ},func]


AppellFunctionQ[func_] :=
  MemberQ[{AppellF1},func]
```

4.0.2 Maple grading function

```
# File: GradeAntiderivative.mpl
# Original version thanks to Albert Rich emailed on 03/21/2017

#Nasser 03/22/2017  Use Maple leaf count instead since buildin
#Nasser 03/23/2017  missing 'ln' for ElementaryFunctionQ added
#Nasser 03/24/2017  corrected the check for complex result
#Nasser 10/27/2017  check for leafsize and do not call ExpnType()
#                   if leaf size is "too large". Set at 500,000
```

```
#Nasser 12/22/2019  Added debug flag, added 'dilog' to special functions
#                   see problem 156, file Apostol_Problems

GradeAntiderivative := proc(result,optimal)
local leaf_count_result, leaf_count_optimal,ExpnType_result,ExpnType_optimal,
   debug:=false;

      leaf_count_result:=leafcount(result);
      #do NOT call ExpnType() if leaf size is too large. Recursion problem
      if leaf_count_result > 500000 then
          return "B";
      fi;

      leaf_count_optimal:=leafcount(optimal);

      ExpnType_result:=ExpnType(result);
      ExpnType_optimal:=ExpnType(optimal);

      if debug then
         print("ExpnType_result",ExpnType_result," ExpnType_optimal=",
   ExpnType_optimal);
      fi;

# If result and optimal are mathematical expressions,
#  GradeAntiderivative[result,optimal] returns
#   "F" if the result fails to integrate an expression that
#       is integrable
#   "C" if result involves higher level functions than necessary
#   "B" if result is more than twice the size of the optimal
#       antiderivative
#   "A" if result can be considered optimal

  #This check below actually is not needed, since I only
  #call this grading only for passed integrals. i.e. I check
  #for "F" before calling this. But no harm of keeping it here.
  #just in case.


  if not type(result,freeof('int')) then
     return "F";
  end if;


  if ExpnType_result<=ExpnType_optimal then
      if debug then
          print("ExpnType_result<=ExpnType_optimal");
      fi;
```

```
      if is_contains_complex(result) then
         if is_contains_complex(optimal) then
            if debug then
               print("both result and optimal complex");
            fi;
            #both result and optimal complex
            if leaf_count_result<=2*leaf_count_optimal then
               return "A";
            else
               return "B";
            end if
         else #result contains complex but optimal is not
            if debug then
               print("result contains complex but optimal is not");
            fi;
            return "C";
         end if
      else # result do not contain complex
           # this assumes optimal do not as well
           if debug then
               print("result do not contain complex, this assumes optimal do not
    as well");
           fi;
           if leaf_count_result<=2*leaf_count_optimal then
              if debug then
                 print("leaf_count_result<=2*leaf_count_optimal");
              fi;
              return "A";
           else
              if debug then
                 print("leaf_count_result>2*leaf_count_optimal");
              fi;
              return "B";
           end if
      end if
  else #ExpnType(result) > ExpnType(optimal)
      if debug then
         print("ExpnType(result) > ExpnType(optimal)");
      fi;
      return "C";
  end if

end proc:

#
# is_contains_complex(result)
# takes expressions and returns true if it contains "I" else false
```

```
#
#Nasser 032417
is_contains_complex:= proc(expression)
  return (has(expression,I));
end proc:

# The following summarizes the type number assigned an expression
# based on the functions it involves
# 1 = rational function
# 2 = algebraic function
# 3 = elementary function
# 4 = special function
# 5 = hyperpergeometric function
# 6 = appell function
# 7 = rootsum function
# 8 = integrate function
# 9 = unknown function

ExpnType := proc(expn)
  if type(expn,'atomic') then
    1
  elif type(expn,'list') then
    apply(max,map(ExpnType,expn))
  elif type(expn,'sqrt') then
    if type(op(1,expn),'rational') then
       1
    else
       max(2,ExpnType(op(1,expn)))
    end if
  elif type(expn,'`^`') then
    if type(op(2,expn),'integer') then
      ExpnType(op(1,expn))
    elif type(op(2,expn),'rational') then
      if type(op(1,expn),'rational') then
         1
      else
         max(2,ExpnType(op(1,expn)))
      end if
    else
         max(3,ExpnType(op(1,expn)),ExpnType(op(2,expn)))
    end if
  elif type(expn,'`+`') or type(expn,'`*`') then
    max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
  elif ElementaryFunctionQ(op(0,expn)) then
    max(3,ExpnType(op(1,expn)))
  elif SpecialFunctionQ(op(0,expn)) then
    max(4,apply(max,map(ExpnType,[op(expn)])))
```

```
  elif HypergeometricFunctionQ(op(0,expn)) then
    max(5,apply(max,map(ExpnType,[op(expn)])))
  elif AppellFunctionQ(op(0,expn)) then
    max(6,apply(max,map(ExpnType,[op(expn)])))
  elif op(0,expn)='int' then
    max(8,apply(max,map(ExpnType,[op(expn)]))) else
  9
  end if
end proc:


ElementaryFunctionQ := proc(func)
  member(func,[
        exp,log,ln,
        sin,cos,tan,cot,sec,csc,
        arcsin,arccos,arctan,arccot,arcsec,arccsc,
        sinh,cosh,tanh,coth,sech,csch,
        arcsinh,arccosh,arctanh,arccoth,arcsech,arccsch])
end proc:

SpecialFunctionQ := proc(func)
  member(func,[
        erf,erfc,erfi,
        FresnelS,FresnelC,
        Ei,Ei,Li,Si,Ci,Shi,Chi,
        GAMMA,lnGAMMA,Psi,Zeta,polylog,dilog,LambertW,
        EllipticF,EllipticE,EllipticPi])
end proc:

HypergeometricFunctionQ := proc(func)
  member(func,[Hypergeometric1F1,hypergeom,HypergeometricPFQ])
end proc:

AppellFunctionQ := proc(func)
  member(func,[AppellF1])
end proc:

# u is a sum or product.  rest(u) returns all but the
# first term or factor of u.
rest := proc(u) local v;
  if nops(u)=2 then
     op(2,u)
  else
     apply(op(0,u),op(2..nops(u),u))
  end if
end proc:
```

```
#leafcount(u) returns the number of nodes in u.
#Nasser 3/23/17  Replaced by build-in leafCount from package in Maple
leafcount := proc(u)
   MmaTranslator[Mma][LeafCount](u);
end proc:
```

4.0.3 Sympy grading function

```
#Dec 24, 2019. Nasser M. Abbasi:
#              Port of original Maple grading function by
#              Albert Rich to use with Sympy/Python
#Dec 27, 2019  Nasser. Added `RootSum`. See problem 177, Timofeev file
#              added 'exp_polar'
from sympy import *

def leaf_count(expr):
    #sympy do not have leaf count function. This is approximation
    return round(1.7*count_ops(expr))

def is_sqrt(expr):
    if isinstance(expr,Pow):
        if expr.args[1] == Rational(1,2):
            return True
        else:
            return False
    else:
        return False

def is_elementary_function(func):
    return func in [exp,log,ln,sin,cos,tan,cot,sec,csc,
            asin,acos,atan,acot,asec,acsc,sinh,cosh,tanh,coth,sech,csch,
            asinh,acosh,atanh,acoth,asech,acsch
        ]

def is_special_function(func):
    return func in [ erf,erfc,erfi,
             fresnels,fresnelc,Ei,Ei,Li,Si,Ci,Shi,Chi,
             gamma,loggamma,digamma,zeta,polylog,LambertW,
             elliptic_f,elliptic_e,elliptic_pi,exp_polar
         ]

def is_hypergeometric_function(func):
    return func in [hyper]

def is_appell_function(func):
    return func in [appellf1]
```

```
def is_atom(expn):
    try:
        if expn.isAtom or isinstance(expn,int) or isinstance(expn,float):
           return True
        else:
           return False

    except AttributeError as error:
        return False

def expnType(expn):
    debug=False
    if debug:
        print("expn=",expn,"type(expn)=",type(expn))

    if is_atom(expn):
       return 1
    elif isinstance(expn,list):
        return max(map(expnType, expn))   #apply(max,map(ExpnType,expn))
    elif  is_sqrt(expn):
        if isinstance(expn.args[0],Rational): #type(op(1,expn),'rational')
            return 1
        else:
            return max(2,expnType(expn.args[0]))  #max(2,ExpnType(op(1,expn)))
    elif isinstance(expn,Pow):   #type(expn,'`^`')
        if isinstance(expn.args[1],Integer):  #type(op(2,expn),'integer')
            return expnType(expn.args[0])   #ExpnType(op(1,expn))
        elif isinstance(expn.args[1],Rational):  #type(op(2,expn),'rational')
            if isinstance(expn.args[0],Rational): #type(op(1,expn),'rational')
                return 1
            else:
                return max(2,expnType(expn.args[0]))  #max(2,ExpnType(op(1,expn)
   ))
        else:
            return max(3,expnType(expn.args[0]),expnType(expn.args[1])) #max(3,
   ExpnType(op(1,expn)),ExpnType(op(2,expn)))
    elif isinstance(expn,Add) or isinstance(expn,Mul): #type(expn,'`+`') or type
   (expn,'`*`')
        m1 = expnType(expn.args[0])
        m2 = expnType(list(expn.args[1:]))
        return max(m1,m2)  #max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
    elif is_elementary_function(expn.func):  #ElementaryFunctionQ(op(0,expn))
        return max(3,expnType(expn.args[0]))  #max(3,ExpnType(op(1,expn)))
    elif is_special_function(expn.func): #SpecialFunctionQ(op(0,expn))
        m1 = max(map(expnType, list(expn.args)))
        return max(4,m1)   #max(4,apply(max,map(ExpnType,[op(expn)])))
```

```
    elif is_hypergeometric_function(expn.func): #HypergeometricFunctionQ(op(0,
   expn))
        m1 = max(map(expnType, list(expn.args)))
        return max(5,m1)   #max(5,apply(max,map(ExpnType,[op(expn)])))
    elif is_appell_function(expn.func):
        m1 = max(map(expnType, list(expn.args)))
        return max(6,m1)   #max(5,apply(max,map(ExpnType,[op(expn)])))
    elif isinstance(expn,RootSum):
        m1 = max(map(expnType, list(expn.args))) #Apply[Max,Append[Map[ExpnType,
   Apply[List,expn]],7]],
        return max(7,m1)
    elif str(expn).find("Integral") != -1:
        m1 = max(map(expnType, list(expn.args)))
        return max(8,m1)   #max(5,apply(max,map(ExpnType,[op(expn)])))
    else:
        return 9

#main function
def grade_antiderivative(result,optimal):

    leaf_count_result  = leaf_count(result)
    leaf_count_optimal = leaf_count(optimal)

    expnType_result  = expnType(result)
    expnType_optimal = expnType(optimal)

    if str(result).find("Integral") != -1:
        return "F"

    if expnType_result <= expnType_optimal:
        if result.has(I):
            if optimal.has(I): #both result and optimal complex
                if leaf_count_result <= 2*leaf_count_optimal:
                    return "A"
                else:
                    return "B"
            else: #result contains complex but optimal is not
                return "C"
        else: # result do not contain complex, this assumes optimal do not as
   well
            if leaf_count_result <= 2*leaf_count_optimal:
                return "A"
            else:
                return "B"
    else:
        return "C"
```

4.0.4 SageMath grading function

```
#Dec 24, 2019. Nasser: Ported original Maple grading function by
#              Albert Rich to use with Sagemath. This is used to
#              grade Fricas, Giac and Maxima results.
#Dec 24, 2019. Nasser: Added 'exp_integral_e' and 'sng', 'sin_integral'
#              'arctan2','floor','abs','log_integral'

from sage.all import *
from sage.symbolic.operators import add_vararg, mul_vararg

debug=False;

def tree_size(expr):
    r"""
    Return the tree size of this expression.
    """
    if expr not in SR:
        # deal with lists, tuples, vectors
        return 1 + sum(tree_size(a) for a in expr)
    expr = SR(expr)
    x, aa = expr.operator(), expr.operands()
    if x is None:
        return 1
    else:
        return 1 + sum(tree_size(a) for a in aa)

def is_sqrt(expr):
    if expr.operator() == operator.pow:   #isinstance(expr,Pow):
        if expr.operands()[1]==1/2: #expr.args[1] == Rational(1,2):
            if debug: print ("expr is sqrt")
            return True
        else:
            return False
    else:
        return False

def is_elementary_function(func):
    debug=False
    m = func.name() in ['exp','log','ln',
            'sin','cos','tan','cot','sec','csc',
            'arcsin','arccos','arctan','arccot','arcsec','arccsc',
            'sinh','cosh','tanh','coth','sech','csch',
            'arcsinh','arccosh','arctanh','arccoth','arcsech','arccsch','sgn',
        'arctan2','floor','abs'
        ]
    if debug:
        if m:
```

```
            print ("func ", func , " is elementary_function")
        else:
            print ("func ", func , " is NOT elementary_function")


    return m

def is_special_function(func):
    debug=False
    if debug: print ("type(func)=", type(func))

    m= func.name() in ['erf','erfc','erfi','fresnel_sin','fresnel_cos','Ei',
           'Ei','Li','Si','sin_integral','Ci','cos_integral','Shi','
    sinh_integral'
           'Chi','cosh_integral','gamma','log_gamma','psi,zeta',
           'polylog','lambert_w','elliptic_f','elliptic_e',
           'elliptic_pi','exp_integral_e','log_integral']

    if debug:
        print ("m=",m)
        if m:
            print ("func ", func ," is special_function")
        else:
            print ("func ", func ," is NOT special_function")


    return m


def is_hypergeometric_function(func):
    return func.name() in ['hypergeometric','hypergeometric_M','hypergeometric_U
    ']

def is_appell_function(func):
    return func.name() in ['hypergeometric']   #[appellf1] can't find this in
    sagemath

def is_atom(expn):

    debug=False
    if debug: print ("Enter is_atom")


    #thanks to answer at https://ask.sagemath.org/question/49179/what-is-
    sagemath-equivalent-to-atomic-type-in-maple/
    try:
        if expn.parent() is SR:
```

```
            return expn.operator() is None
        if expn.parent() in (ZZ, QQ, AA, QQbar):
            return expn in expn.parent() # Should always return True
        if hasattr(expn.parent(),"base_ring") and hasattr(expn.parent(),"gens"):
            return expn in expn.parent().base_ring() or expn in expn.parent().
    gens()
        return False

    except AttributeError as error:
        return False


def expnType(expn):

    if debug:
        print (">>>>>Enter expnType, expn=", expn)
        print (">>>>>is_atom(expn)=", is_atom(expn))

    if is_atom(expn):
        return 1
    elif type(expn)==list:   #isinstance(expn,list):
        return max(map(expnType, expn))   #apply(max,map(ExpnType,expn))
    elif  is_sqrt(expn):
        if  type(expn.operands()[0])==Rational: #type(isinstance(expn.args[0],
    Rational):
            return 1
        else:
            return max(2,expnType(expn.operands()[0]))  #max(2,expnType(expn.
    args[0]))
    elif expn.operator() == operator.pow:   #isinstance(expn,Pow)
        if type(expn.operands()[1])==Integer:  #isinstance(expn.args[1],Integer)
            return expnType(expn.operands()[0])   #expnType(expn.args[0])
        elif type(expn.operands()[1])==Rational:  #isinstance(expn.args[1],
    Rational)
            if type(expn.operands()[0])==Rational: #isinstance(expn.args[0],
    Rational)
                return 1
            else:
                return max(2,expnType(expn.operands()[0]))  #max(2,expnType(expn.
    args[0]))
        else:
            return max(3,expnType(expn.operands()[0]),expnType(expn.operands()
    [1])) #max(3,expnType(expn.operands()[0]),expnType(expn.operands()[1]))
    elif expn.operator() == add_vararg or expn.operator() == mul_vararg: #
    isinstance(expn,Add) or isinstance(expn,Mul)
        m1 = expnType(expn.operands()[0]) #expnType(expn.args[0])
        m2 = expnType(expn.operands()[1:]) #expnType(list(expn.args[1:]))
```

```
        return max(m1,m2)  #max(ExpnType(op(1,expn)),max(ExpnType(rest(expn))))
    elif is_elementary_function(expn.operator()):  #is_elementary_function(expn.
    func)
        return max(3,expnType(expn.operands()[0]))
    elif is_special_function(expn.operator()): #is_special_function(expn.func)
        m1 = max(map(expnType, expn.operands()))      #max(map(expnType, list(
    expn.args)))
        return max(4,m1)   #max(4,m1)
    elif is_hypergeometric_function(expn.operator()): #
    is_hypergeometric_function(expn.func)
        m1 = max(map(expnType, expn.operands()))       #max(map(expnType, list(
    expn.args)))
        return max(5,m1)   #max(5,m1)
    elif is_appell_function(expn.operator()):
        m1 = max(map(expnType, expn.operands()))       #max(map(expnType, list(
    expn.args)))
        return max(6,m1)   #max(6,m1)
    elif str(expn).find("Integral") != -1: #this will never happen, since it
                #is checked before calling the grading function that is passed.
                #but kept it here.
        m1 = max(map(expnType, expn.operands()))       #max(map(expnType, list(
    expn.args)))
        return max(8,m1)   #max(5,apply(max,map(ExpnType,[op(expn)])))
    else:
        return 9

#main function
def grade_antiderivative(result,optimal):

    if debug: print ("Enter grade_antiderivative for sagemath")

    leaf_count_result  = tree_size(result) #leaf_count(result)
    leaf_count_optimal = tree_size(optimal) #leaf_count(optimal)

    if debug: print ("leaf_count_result=", leaf_count_result, "
    leaf_count_optimal=",leaf_count_optimal)


    expnType_result  = expnType(result)
    expnType_optimal = expnType(optimal)

    if debug: print ("expnType_result=", expnType_result, "expnType_optimal=",
    expnType_optimal)

    if expnType_result <= expnType_optimal:
        if result.has(I):
            if optimal.has(I): #both result and optimal complex
```

```
            if leaf_count_result <= 2*leaf_count_optimal:
                return "A"
            else:
                return "B"
        else: #result contains complex but optimal is not
            return "C"
    else: # result do not contain complex, this assumes optimal do not as
well
        if leaf_count_result <= 2*leaf_count_optimal:
            return "A"
        else:
            return "B"
else:
    return "C"
```