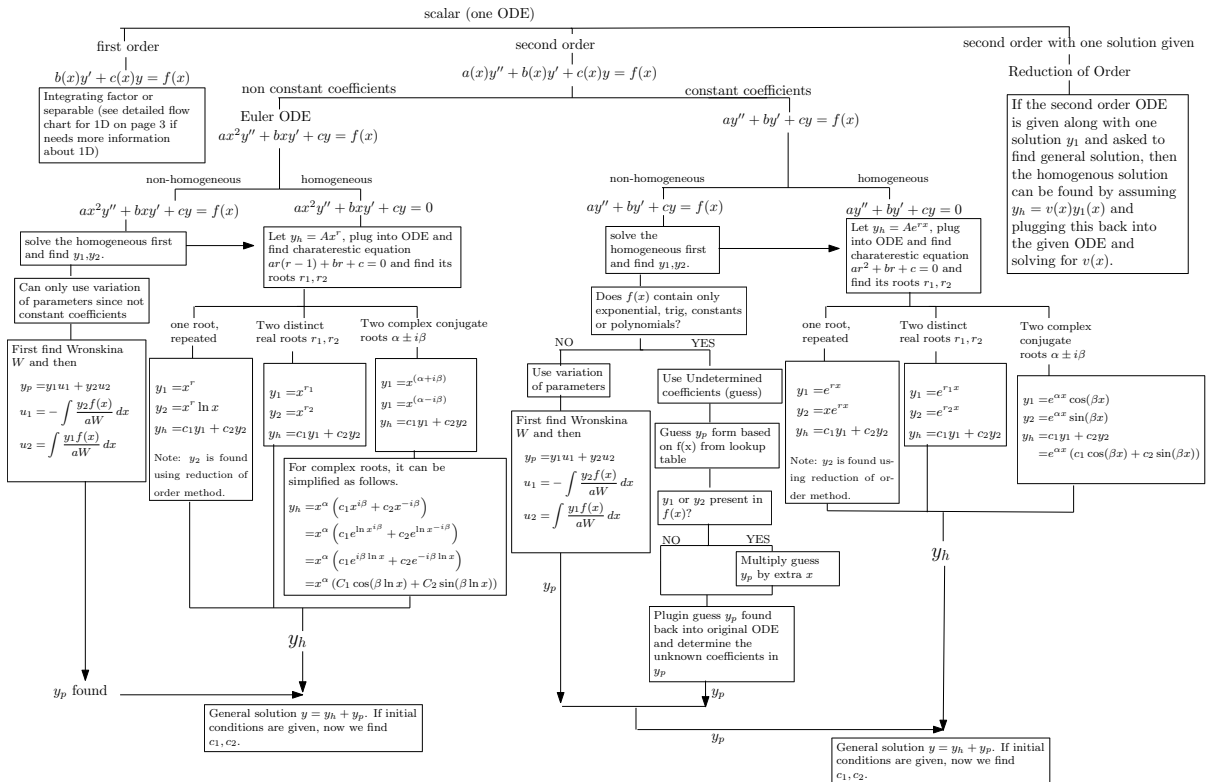


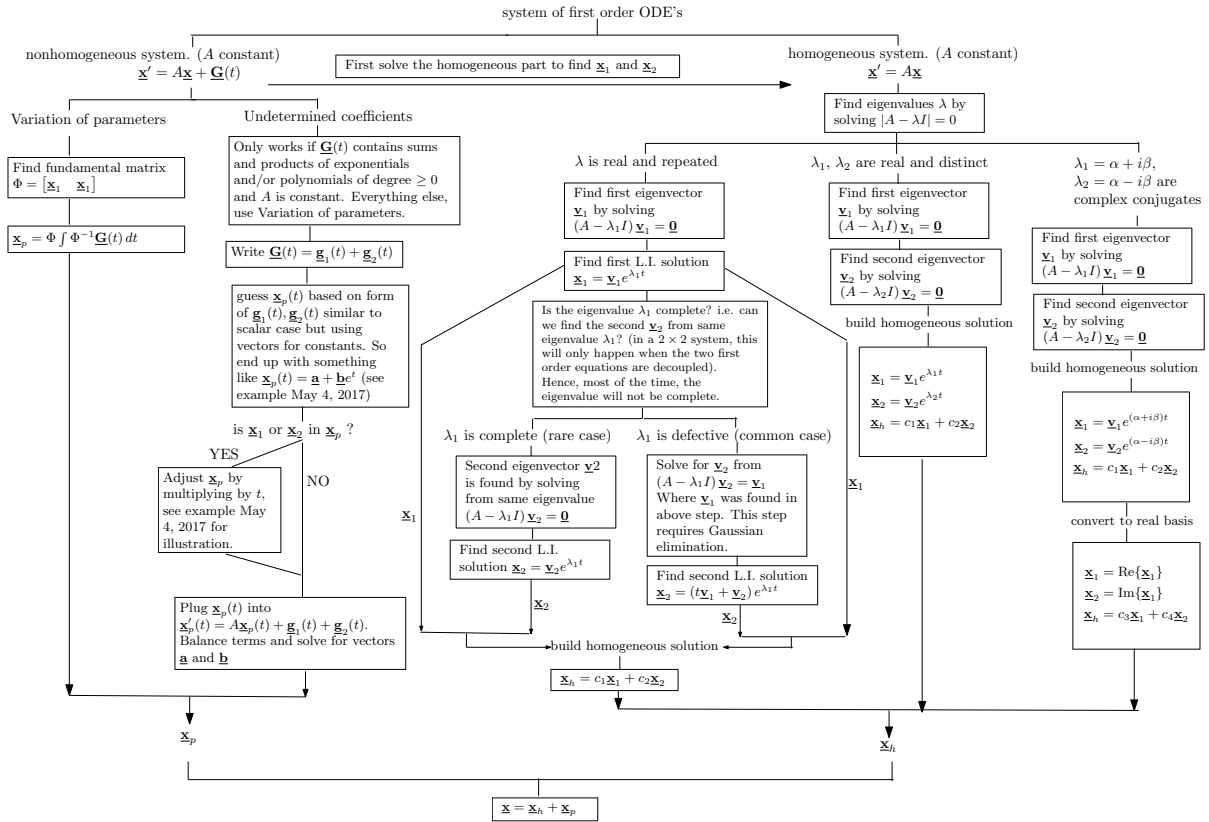
ODE classification

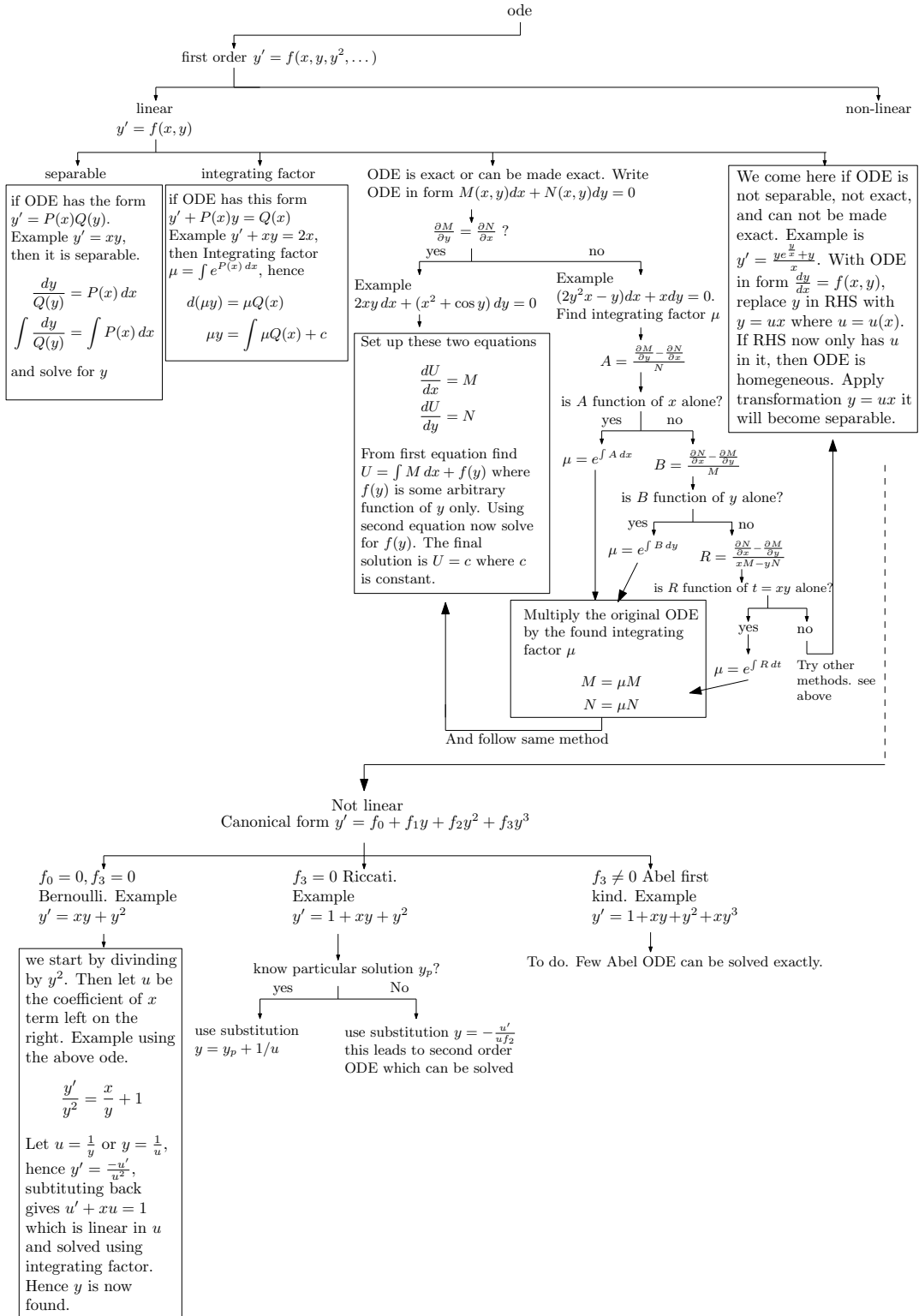
Nasser M. Abbasi

December 29, 2020

Compiled on December 29, 2020 at 12:05am







1 Detailed algorithm

Algorithm 1 solve_first_order_ode

```
procedure SOLVE_FIRST_ORDER_ODE(ode,y,x,ic)
  if ode missing  $y'(x)$  then
    if ode missing  $y(x)$  then
      return (error)
    else
      this is algebraic equation and not ode. solve for  $y(x)$ 
    end if
  else
    if ode missing  $y(x)$  then
      solve for  $y'(x)$  and integrate each solution directly to find  $y(x)$ 
    else
      general case, where both  $y'(x)$  and  $y(x)$  are present.
      if ode is linear in  $y'(x)$  then
        solve for  $y'(x)$  and write ode as  $y'(x) = f(x, y)$ 
        if  $f(x, y)$  is linear in  $y(x)$  then
          write  $f(x, y)$  in the form  $f(x, y) = A + By(x)$ 
          solve  $y' = A + By$  using integrating factor
        else
          ode is linear in  $y'(x)$  but not linear in  $y(x)$ . Most common case.
          if is_class_A_ode(ode) then
            return solution
          else
            unable to solve
          end if
        end if
      else
        the hard case. ode not linear in  $y'(x)$ 
        if is_class_B_ode(ode) then
          return solution
        else
          unable to solve
        end if
      end if
    end if
  end if
end procedure
```

Algorithm 2 is_class_A_ode

procedure IS_CLASS_A_ODE(ode,y,x)

ode is 1st order, linear in $y'(x)$ but not linear in $y(x)$

let ode be $y'(x) = f(x,y)$

if is_Bernoulli_ode(ode) **then**

 solve as Bernoulli. Example is $y' = xy + y^3$

return true

else if is_Riccati_ode(ode) **then**

 solve as Riccati.

return true

else if is_Abel_first_kind(ode) **then**

 solve as Abel first kind

return true

else if is_Abel_second_kind(ode) **then**

 solve as Abel second kind

return true

else if is_Chini_(ode) **then**

 solve as Chini ode

return true

else if is_separable_(ode) **then**

 solve as separable ode

return true

else if is_homogeneous_first_order(ode) **then**

 solve as homogeneous first order ode

return true

else if is_polynomial_form_ode(ode) **then**

 solve as special polynomial form ode

return true

else if is_exact_ode(ode) **then**

 solve as exact ode

return true

else if is_dAlembert_ode(ode) **then**

 Special check. Check for dAlembert ODE even if the input ODE was linear in $y'(x)$ this is because of cases like $y'(x) = \sqrt{1+x+y(x)}$, which can be written as $y'(x)^2 = 1+x+y(x)$ or $y(x) = -1-x+y'(x)^2$ which is dAlembert ODE.

 Call dAlembert ODE solver to solve it.

return true

else if is_Clairaut_ode(ode) **then**

 Special check. Check for Clairaut ODE even if the input ODE was linear in $y'(x)$ this is because of cases like $y'(x) + 1 + \frac{x}{2} = \sqrt{x^2 + 4x + 4y}$, which can be written as $y(x) = xy'(x) + ((y')^2 + 1 + 2y')$ which is Clairaut ODE ODE.

 Call Clairaut ODE solver to solve it.

return true

else

return false

end if

end procedure

Algorithm 3 is_class_B_ode

```
procedure IS_CLASS_B_ODE(ode,y,x,ic)
  ode is first order ODE, not linear in  $y'(x)$ 
  if is_Clairaut_ode(ode) then
    solve as clairaut ode
    return true
  else if is_dAlembert_ode(ode) then
    solve as dAlembert_ode
    return true
  else if ode has form  $(y')^n = f(x, y)$  then
    if  $|n| > 1$  then
      sols = solve for  $y'$  by finding the roots. This will generate  $n$  new ode's
      for all  $sol_i$  in sols do
        call solve_first_order_ode( $sol_i, y, x, ic$ )
      end for
      return true
    else
      since  $|n| < 1$ , then ode has form  $(y')^{1/m} = f(x, y)$  for some integer  $m > 1$ . raise power of
      both side by  $m$  to obtain  $y' = f(x, y)^m$  and call solve_first_order_ode() on this ode.

      return true
    end if
  else
    if ode has form  $(A(y')^n + B(y')^m + \dots) = f(x, y)$  then
      solve for  $y'$ . For each root solution call solve_first_order_ode()
      return true
    else
      return false
    end if
  end if
end procedure
```

Algorithm 4 check if ODE is separable type

```
procedure IS_SEPARABLE_ODE(ode,y,x,ic)
  ode is first order ODE linear in  $y'(x)$ 
  if ode has form  $y' = A(x)B(y)$  where  $A(x)$  is function of  $x$  only and  $B(y)$ 
  is function of  $y$  only, both  $A(x), B(y)$  can be linear or nonlinear functions then
    return true
  else
    return false
  end if
end procedure
```

Algorithm 5 check if ODE is homogeneous first order type

```
procedure IS_HOMOGENEOUS_FIRST_ORDER(ode,y,x,ic)
  ode is first order ODE linear in  $y'(x)$ 
  if ode has form  $y' = F(x,y)$  where  $F(x,y)$  is homogeneous function then
    A homogeneous is one which  $F(tx, ty) = F(x,y)$  for all  $t$ 
    return true
  else
    return false
  end if
end procedure
```

Algorithm 6 check if ODE is ratio of two polynomials type

```
procedure IS_POLYNOMIAL_FORM_ODE(ode,y,x,ic)
  ode is first order ODE linear in  $y'(x)$ 
  if ode has form  $y' = \frac{Ax+By+C}{Dx+Ey+F}$  for non-zero  $A, B, C, D, E, F$  then
    return true
  else
    return false
  end if
end procedure
```

Algorithm 7 check if ODE is exact or can be made exact using special integrating factor

```
procedure IS_EXACT_ODE(ode,y,x,ic)
  ode is first order ODE linear in  $y'(x)$ 
  write the ode as  $M(x,y) + N(x,y)y' = 0$  or  $M(x,y) dx + N(x,y) dy = 0$ 
  if  $\frac{\partial M}{\partial y} = \frac{\partial N}{\partial x}$  then
    example is  $2xy + (x^2 + \cos y)y' = 0$ 
    return true
  else
    example is  $(2y^2x - y) dx + x dy = 0$ 
    Let  $A = \frac{1}{N} \left( \frac{\partial M}{\partial y} - \frac{\partial N}{\partial x} \right)$ 
    if A function of  $x$  only then
      return true
    else
      Let  $B = \frac{-1}{M} \left( \frac{\partial M}{\partial y} - \frac{\partial N}{\partial x} \right)$ 
      if B function of  $y$  only then
        return true
      else
        Let  $R = \frac{-1}{xM-yN} \left( \frac{\partial M}{\partial y} - \frac{\partial N}{\partial x} \right)$ 
        if R function of  $t$  only where  $t = xy$  then
          return true
        else
          return false
        end if
      end if
    end if
  end if
end procedure
```

Algorithm 8 check if ODE is Bernoulli type

```
procedure IS_BERNOULLI_ODE(ode,y,x,ic)  
  ode is first order ODE linear in  $y'(x)$   
  if ode has form  $y' = Ay + By^n$  with  $n \neq 0, n \neq 1$  then  
    if at least one of  $A, B$  is function of  $x$  then  
      return true  
    else  
      return false  
    end if  
  else  
    return false  
  end if  
end procedure
```

Algorithm 9 check if ODE is Riccati type

```
procedure IS_RICCATI_ODE(ode,y,x,ic)  
  ode is first order ODE linear in  $y'(x)$   
  if ode has the form  $y' = A + By + Cy^2$  or  $y' = A + Cy^2$  then  
    if at least one of  $A, B, C$  is function of  $x$  then  
      return true  
    else  
      return false  
    end if  
  else  
    return false  
  end if  
end procedure
```

Algorithm 10 check if ODE is Abel first kind type

```
procedure IS_ABEL_FIRST_KIND_ODE(ode,y,x,ic)
  ode is first order ODE linear in  $y'(x)$ 
  if ode has one of the following forms
     $y' = A + By + Cy^2 + Dy^3$ 
     $y' = By + Cy^2 + Dy^3$ 
     $y' = A + Cy^2 + Dy^3$ 
     $y' = A + By + Dy^3$ 
     $y' = A + Dy^3$ 
     $y' = Cy^2 + Dy^3$ 
    then
      if at least one of  $A, B, C, D$  is function of  $x$  then
        return true
      else
        return false
      end if
    else
      return false
    end if
  end procedure
```

Algorithm 11 check if ODE is Abel first kind type

```
procedure IS_ABEL_FIRST_KIND_ODE(ode,y,x,ic)
  ode is first order ODE linear in  $y'(x)$ 
  if ode has the form  $(E + Fy)y' = A + By + Cy^2 + Dy^3$  then
    if is_Abel_first_kind_ode( $y' = A + By + Cy^2 + Dy^3$ ) and  $F \neq 0$  then
      return true
    else
      return false
    end if
  else
    return false
  end if
end procedure
```

Algorithm 12 check if ODE is Chini type

```
procedure IS_CHINI_ODE(ode,y,x,ic)  
  ode is first order ODE linear in  $y'(x)$   
  if ode has one of these forms  
     $y' = A + By^n$   
     $y' = A + By + Cy^n$   
    where  $n \neq 0, n \neq 1, n \neq 2, n \neq 3$  then  
      if at least one of  $A, B, C$  is function of  $x$  then  
        return true  
      else  
        return false  
      end if  
    else  
      return false  
    end if  
end procedure
```

2 examples

2.1 second order, constant coeff.

2.1.1 second order, constant coeff. homogeneous

second order, constant coeff. homogeneous, one root repeated

$$y'' - 2y' + 1 = 0$$

Let $y = Ae^{rx}$ and plug into the above and simplify, we obtain the characteristic equation

$$r^2 - 2r + 1 = 0$$

$$(r - 1)^2 = 0$$

$$r = 1$$

Repeated root. Hence the two L.I. basis solutions are

$$y_1 = e^x$$

$$y_2 = xe^x$$

And the homogeneous solution is

$$y_h = c_1y_1 + c_2y_2$$

$$= c_1e^x + c_2xe^x$$

second order, constant coeff. homogeneous, two real distinct roots

$$y'' + y' - 2y = 0$$

Let $y = Ae^{rx}$ and plug into the above and simplify, we obtain the characteristic equation

$$\begin{aligned}r^2 + r - 2 &= 0 \\(r - 1)(r + 2) &= 0 \\r_1 &= 1 \\r_2 &= -2\end{aligned}$$

Hence the L.I. basis solutions are

$$\begin{aligned}y_1 &= e^x \\y_2 &= e^{-2x}\end{aligned}$$

And the homogeneous solution is

$$\begin{aligned}y_h &= c_1y_1 + c_2y_2 \\&= c_1e^x + c_2e^{-2x}\end{aligned}$$

second order, constant coeff. homogeneous, two complex conjugate roots

$$y'' - 6y' + 13y = 0$$

Let $y = Ae^{rx}$ and plug into the above and simplify, we obtain the characteristic equation

$$r^2 - 6r + 13 = 0$$

Whose roots are

$$\begin{aligned}r_1 &= 3 + 2i \\r_2 &= 3 - 2i\end{aligned}$$

Hence the L.I. basis solutions are

$$\begin{aligned}y_1 &= e^{(3+2i)x} \\y_2 &= e^{(3-2i)x}\end{aligned}$$

the homogeneous solution is

$$\begin{aligned}y_h &= c_1y_1 + c_2y_2 \\&= c_1e^{(3+2i)x} + c_2e^{(3-2i)x}\end{aligned}$$

This can be converted to real basis using Euler relation which results in

$$\begin{aligned}y_h &= C_1e^{3x} \cos 2x + C_2e^3 \sin 2x \\&= e^{3x}(C_1 \cos 2x + C_2 \sin 2x)\end{aligned}$$