HW 3, Math 320, Spring 2017

Nasser M. Abbasi (Discussion section 383, 8:50 AM - 9:40 AM Monday)

December 30, 2019

Contents

0.1 Section 2.4 problem 8 (page 122)

Problem Apply Euler method twice to approximate solution on interval $\left[0, \frac{1}{2}\right]$ first with step size $h = 0.25$ then with step size $h = 0.1$. Compare to three decimal places values of the two approximation at $x = \frac{1}{2}$ with the value $y\left(\frac{1}{2}\right)$ of the exact solution. $y' = e^{-y}; y(0) = 0$. Exact solution is $y(x) = \ln(x+1)$

Solution

Using forward Euler method, we write

$$y_{n+1} = y_n + hf\left(x_n, y_n\right)$$

Here $f\left(x, y\right) = e^{-y}$.

$h = 0.25$

$$\begin{aligned} y(0) &= 0 \\ y(h) = y(0.25) &= y(0) + he^{-y(0)} = 0 + 0.25e^{0} = 0.25 \\ y(2h) = y(0.5) &= y(0.25) + he^{-y(0.25)} = 0.25 + 0.25e^{-0.25} = 0.445 \end{aligned}$$

$h = 0.1$

$$\begin{aligned} y(0) &= 0 \\ y(h) = y(0.1) &= y(0) + he^{-y(0)} = 0 + 0.1e^{-0} = 0.1 \\ y(2h) = y(0.2) &= y(0.1) + he^{-y(0.1)} = 0.1 + 0.1e^{-0.1} = 0.190 \\ y(3h) = y(0.3) &= y(0.2) + he^{-y(0.2)} = 0.190 + 0.1e^{-0.190} = 0.273 \\ y(4h) = y(0.4) &= y(0.3) + he^{-y(0.3)} = 0.273 + 0.1e^{-0.273} = 0.349 \\ y(5h) = y(0.5) &= y(0.4) + he^{-y(0.4)} = 0.349 + 0.1e^{-0.349} = 0.420 \end{aligned}$$

Exact solution is $y(0.5) = \ln(0.5+1) = 0.405$

h size	$y\left(\frac{1}{2}\right)$
0.25	0.445
0.1	0.420
exact	0.405

0.2 Section 2.4 problem 13 (page 122)

Problem Find the exact solution, then apply Euler method twice to approximate to 4 decimal places values the solution on the given interval. First with step $h = 0.01$ then with step $h = 0.005$. Make table showing the approximate values and the actual values, together with percentage error in the more accurate approximation for x an integral multiple of 0.2. $yy' = 2x^3; y(1) = 3; 1 \le x \le 2$.

Solution

$$y' = \frac{2x^3}{y} = f\left(x, y\right)$$

Looking at $f(x)$ we see that solution is not defined at $y = 0$. Otherwise, $f\left(x, y\right)$ is continuous everywhere. Hence solution exist for $y \neq 0$. Also $\frac{\partial f}{\partial y} = -\frac{2x^3}{y^2}$, hence we see solution is unique, on some interval that does not include $y = 0$. Now we will solve the ODE

$$y\frac{dy}{dx} = 2x^3$$
$$ydy = 2x^3 dx$$

Integrating

$$\frac{1}{2}y^2 = \frac{1}{2}x^4 + c$$

Applying initial conditions

$$\frac{1}{2}(9) = \frac{1}{2} + c$$
$$\frac{9}{2} - \frac{1}{2} = c$$
$$c = 4$$

Hence exact solution is

$$\frac{1}{2}y^2 = \frac{1}{2}x^4 + 4$$
$$y^2 = x^4 + 8$$
$$y = \pm\sqrt{x^4 + 8}$$

Since $1 \le x \le 3$ and $y(1) = 3$, then y can not become negative (else it will have to cross $y = 0$). Therefore solution is just the positive branch

$$y_{exact} = \sqrt{x^4 + 8}$$

Using Euler, we write

$$y_{n+1} = y_n + hf\left(x_n, y_n\right)$$

But $f\left(x_n, y_n\right) = \frac{2x_n^3}{y_n}$ and $x_n = 1 + nh$ where h is the step size. The above becomes

$$y_{n+1} = y_n + hf\left(x_n, y_n\right)$$

Using initial conditions, where $n = 0$, the given values $y_0 = 3$ at $x_0 = 1$. A small function was written to implement Euler method and print table. Source code is given below. Here is the final table generated

x	1.	1.2	1.4	1.6	1.8	2.
h=0.01	3	3.1718	3.4368	3.8084	4.2924	4.889
h=0.005	3	3.1729	3.439	3.8117	4.2967	4.894
exact	3.	3.1739	3.4412	3.8149	4.3009	4.899
% error	0.	0.032303	0.062773	0.085478	0.098183	0.10218

Source code listing:

In[69]:=
```
SetOptions[$FrontEndSession, PrintPrecision → 5]
(*HW 3, Math 320. By Nasser M. Abbasi. Problem 2.4 13*)
f[x_, y_] := 2 x^3 / y;
makeTable[h_, from_, to_, y0_] := Module[{nSteps = (to - from) / h, data, y, x, skip},
  Array[y, Rationalize@nSteps, 0];
  Array[x, Rationalize@nSteps, 0];
  y[0] = y0; x[0] = from;

  Do[(*Euler loop*)
   y[n + 1] = y[n] + h f[x[n], y[n]];
   x[n + 1] = x[n] + h,
   {n, 0, nSteps}
  ];

  skip = Round[0.2 / h];
  Table[{x[n], y[n]}, {n, 0, nSteps, skip}]
 ]
```

In[76]:=
```
data1 = makeTable[0.01, 1, 2, 3];
data2 = makeTable[0.005, 1, 2, 3];
exact = Sqrt[#^4 + 8] & /@ data2[[All, 1]];
p = Grid[{
    {"x", Sequence @@ data1[[All, 1]]},
    {"h=0.01", Sequence @@ data1[[All, 2]]},
    {"h=0.005", Sequence @@ data2[[All, 2]]},
    {"exact", Sequence @@ exact},
    {"% error", Sequence @@ (Abs[data2[[All, 2]] - exact] / Abs[exact] * 100)}
  }, Frame → All]
```

0.3 Section 2.4 problem 25

Problem Apply Euler method for $\frac{dv}{dt} = 32 - 1.6v$ with $v(0) = 0$. For $0 \le t \le 2$, using step size $h = 0.01, h = 0.005$, round v to one decimal point. What percentage of limiting velocity 20 ft/sec has been attained after 1 second? After 2 seconds?

Solution

The exact solution is

$$\frac{dv}{dt} + 1.6v = 32$$

$\mu = e^{\int 1.6dt} = e^{1.6t}$, hence

$$\frac{d}{dt}\left(e^{1.6t}v\right) = 32e^{1.6t}$$

Integrating

$$\begin{aligned} e^{1.6t}v &= 32\int e^{1.6t}dt \\ &= \frac{32}{1.6}e^{1.6t} + c \end{aligned}$$

Hence

$$v(t) = \frac{32}{1.6} + ce^{-1.6t}$$

Applying initial conditions

$$\begin{aligned} 0 &= \frac{32}{1.6} + c \\ c &= -\frac{32}{1.6} \end{aligned}$$

Therefore, exact solution is

$$\begin{aligned} v(t) &= \frac{32}{1.6} - \frac{32}{1.6}e^{-1.6t} \\ &= 20\left(1 - e^{-1.6t}\right) \end{aligned}$$

The Euler method is

$$y_{n+1} = y_n + hf\left(x_n, y_n\right)$$

Where here

$$f\left(x_n, y_n\right) = 32 - 1.6y_n$$

Small function was written to find $v(t)$ at $t = 1, 2$ seconds using Euler, with the different step sizes. It prints the value of v when iteration reaches 1 and 2 seconds. Here is the screen output

```
data1 = makeTable[0.01, 0, 2, 0];
At one second, using h=0.01 speed is 16.078 at step n = 100
At 2 seconds, using h=0.01 speed is 19.206 at step n = 200
data2 = makeTable[0.005, 0, 2, 0];
At one second, using h=0.005 speed is 16.02 at step n = 200
At 2 seconds, using h=0.005 speed is 19.195 at step n = 400
```

Therefore (where percentage below, is percentage of limiting speed of 20 ft/sec)

h	speed at 1 second	speed at 2 seconds
0.01	16.078	19.206
0.005	16.02 (80.1%)	19.195 (95.98%)

The source code written for this problem is given below

In[80]:=
```
SetOptions[$FrontEndSession, PrintPrecision → 5]
(*HW 3, Math 320. By Nasser M. Abbasi. Problem 2.4 25*)
f[t_, y_] := 32 - 1.6 y;
makeTable[h_, from_, to_, y0_] := Module[{nSteps = Rationalize[(to - from) / h], data, t, skip, y},
  Array[y, nSteps, 0];
  Array[t, nSteps, 0];
  y[0] = y0; t[0] = from;

  Do[(*Euler loop*)
   y[n + 1] = y[n] + h f[t[n], y[n]];
   If[t[n] == 1, Print["At one second, using h=", h, " speed is ", y[n + 1], " at step n = ", n]];
   t[n + 1] = t[n] + h,
   {n, 0, nSteps}
  ];
  Print["At 2 seconds, using h=", h, " speed is ", y[nSteps], " at step n = ", nSteps];
  (*skip=Round[0.2/h];*)
  skip = 1;
  Table[{t[n], y[n]}, {n, 0, nSteps, skip}]
 ]
```

In[83]:=
```
data1 = makeTable[0.01, 0, 2, 0];
```
```
At one second, using h=0.01 speed is 16.078 at step n = 100
At 2 seconds, using h=0.01 speed is 19.206 at step n = 200
```

In[84]:=
```
data2 = makeTable[0.005, 0, 2, 0];
```
```
At one second, using h=0.005 speed is 16.02 at step n = 200
At 2 seconds, using h=0.005 speed is 19.195 at step n = 400
```

0.4 Section 2.4 problem 30

Problem Apply Euler method with successively smaller step sizes on the interval $[0,2]$ to verify empirically that the solution of $y' = x^2 + y^2; y(0) = 0$ has vertical asymptote near $x = 2.003147$. Contrast this with example 2, in which $y(0) = 1$.

Solution

Small function was written to implement Forward Euler for this problem.

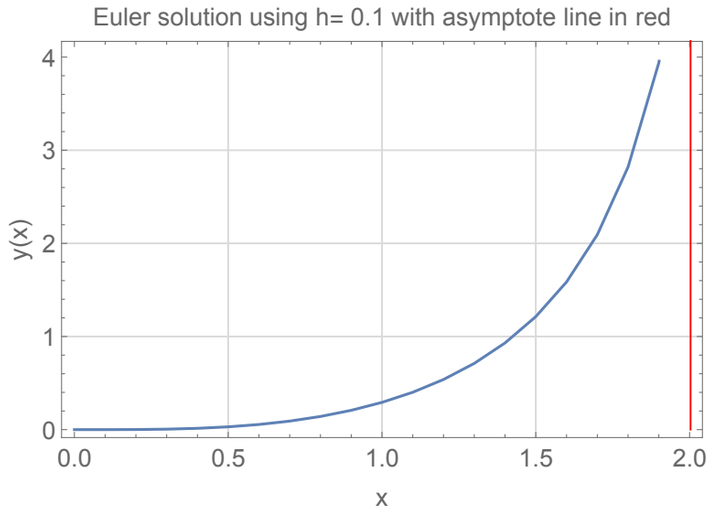
In[85]:=
```
SetOptions[$FrontEndSession, PrintPrecision → 5]
(*HW 3, Math 320. By Nasser M. Abbasi. Problem 2.4 30*)
f[x_, y_] := x^2 + y^2;
makeTable[h_, from_, to_, y0_] := Module[{nSteps = Rationalize[(to - from) / h], data, x, y, skip},
  Array[y, nSteps, 0];
  Array[x, nSteps, 0];
  (*Print["number of steps is ",nSteps];*)
  y[0] = y0; x[0] = from;

  Do[(*Euler loop*)
   y[n + 1] = y[n] + h f[x[n], y[n]];
   x[n + 1] = x[n] + h,
   {n, 0, nSteps}
  ];
  skip = 1;
  Table[{x[n], y[n]}, {n, 0, nSteps, skip}]
 ]
```

The above function was called for $h = 0.1, 0.01, 0.001$ which showed that better and better approximation, the numerical solution approached asymptote near $x = 2.003147$. For $h = 0.1$, here is the output

In[88]:=
```
h = 0.1;
data1 = makeTable[h, 0, 2, 0];
p1 = ListLinePlot[data1,
  Frame → True,
  FrameLabel →
   {{"y(x)", None},
    {"x", Row[{"Euler solution using h= ", h, " with asymptote line in red"}]}},
  BaseStyle → 14, GridLines → Automatic, GridLinesStyle → LightGray,
  Epilog → {Red, Line[{{2.003147, 0}, {2.003147, 7}}]}, ImageSize → 400]
```

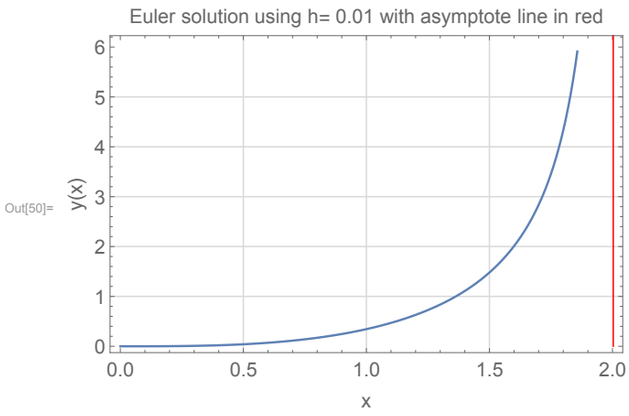
Out[90]=



For $h = 0.01$, here is the output

In[48]:=
```
h = 0.01;
data1 = makeTable[h, 0, 2, 0];
p1 = ListLinePlot[data1,
  Frame → True,
  FrameLabel →
   {{"y(x)", None},
    {"x", Row[{"Euler solution using h= ", h, " with asymptote line in red"}]}},
  BaseStyle → 14, GridLines → Automatic, GridLinesStyle → LightGray,
  Epilog → {Red, Line[{{2.003147, 0}, {2.003147, 7}}]}, ImageSize → 400]
```

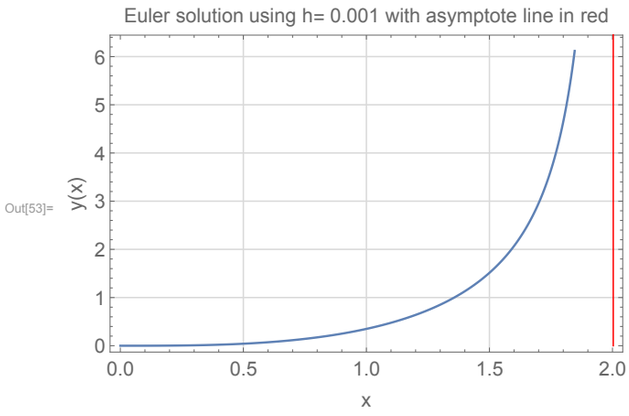
Out[50]=



For $h = 0.001$, here is the output

In[51]:=
```
h = 0.001;
data1 = makeTable[h, 0, 2, 0];
p1 = ListLinePlot[data1,
  Frame → True,
  FrameLabel →
   {{"y(x)", None},
    {"x", Row[{"Euler solution using h= ", h, " with asymptote line in red"}]}},
  BaseStyle → 14, GridLines → Automatic, GridLinesStyle → LightGray,
  Epilog → {Red, Line[{{2.003147, 0}, {2.003147, 7}}]}, ImageSize → 400]
```

Out[53]=



0.5 Section 2.5 problem 8

Apply improved Euler method to approximate solution on interval $\left[0, \frac{1}{2}\right]$ with step size $h = 0.1$ Construct table showing 4 decimal places values of approximation solution and exact solution at points $0.1, 0.2, 0.3, 0.4, 0.5$.

$$y' = e^{-y}; y(0) = 0$$

Exact solution is $y(x) = \ln(x+1)$

Solution

Improved Euler method uses

$$\begin{aligned} k_1 &= f\left(x_n, y_n\right) \\ u_{n+1} &= y_n + hk_1 \\ k_2 &= f\left(x_{n+1}, u_{n+1}\right) \\ y_{n+1} &= y_n + h\frac{k_1 + k_2}{2} \end{aligned}$$

A small function was written to implement the above improved Euler method. The following is source code

```
(*HW 3, Math 320. By Nasser M. Abbasi. Problem 2.5 8, improved Euler*)
f[x_, y_] := Exp[-y];
makeTableImproved[h_, from_, to_, y0_] :=
 Module[{nSteps = Rationalize[(to - from) / h], data, x, y, skip, k1, k2, predictor},
  Array[y, nSteps, 0];
  Array[x, nSteps, 0];
  y[0] = y0; x[0] = from;

  Do[(*Euler loop*)
   k1 = f[x[n], y[n]];
   predictor = y[n] + h k1;
   x[n + 1] = x[n] + h;
   k2 = f[x[n + 1], predictor];
   y[n + 1] = y[n] + h (1 / 2 * (k1 + k2)),
   {n, 0, nSteps}
  ];
  skip = Round[0.1 / h];
  Table[{x[n], y[n]}, {n, 0, nSteps, skip}]
 ]
```

This function was called to generate the table and format it. Here is the result

In[270]:=
```
h = 0.1;
data1 = makeTable[h, 0, .5, 0];
exact = Log[# + 1] & /@ data1[[All, 1]];
p = Grid[{
   {"x", Sequence @@ data1[[All, 1]]},
   {"h=0.01", Sequence @@ data1[[All, 2]]},
   {"exact", Sequence @@ exact},
   {"% error",
    Sequence @@ ((exact - data1[[All, 2]]) / (If[exact == 0, 1, exact, 1]) * 100)}
  }, Frame → All]
```

Out[273]=

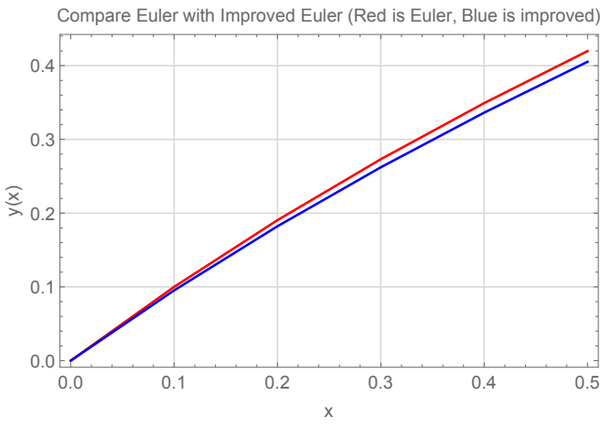
x	0	0.1	0.2	0.3	0.4	0.5
h=0.01	0	0.09524187	0.1822067	0.2622174	0.3363033	0.405281
exact	0	0.09531018	0.1823216	0.2623643	0.3364722	0.4054651
% error	0	0.00683089	0.01148799	0.01469129	0.01689781	0.01840683

Then Euler method was compared to Improved Euler for the same step size $h = 0.1$, by plotting them on the same figure. Here is the result. The red line is the Euler method, and

the blue line is the improved Euler method. We see the difference between them increases as more steps are taken.

In[216]:=
```
h = 0.1;
dataEuler = makeTableEuler[h, 0, .5, 0];
dataEulerImproved = makeTableImproved[h, 0, .5, 0];
p1 = ListLinePlot[{dataEuler, dataEulerImproved},
  Frame → True, PlotStyle → {Red, Blue},
  FrameLabel →
   {{"y(x)", None}, {"x", Row[{"Compare Euler with Improved Euler (Red is Euler, Blue is improved)"}]}},
  BaseStyle → 12, GridLines → Automatic, GridLinesStyle → LightGray, ImageSize → 400]
```

Out[219]=



0.6 Section 2.5 problem 13

Problem Find the exact solution, then apply improved Euler method twice to approximate to 5 decimal places values the solution on the given interval. First with step $h = 0.01$ then with step $h = 0.005$. Make table showing the approximate values and the actual values, together with percentage error in the more accurate approximation for x an integral multiple of 0.2. $yy' = 2x^3; y(1) = 3; 1 \le x \le 2$.

Solution

The analytical solution is the same as in problem 13, section 2.5 and hence will not be repeated again. The improved Euler function, which was written for problem 8 above, was now used for $h = 0.01$ and $h = 0.005$. Source code is given above in problem 8. Here is the final table generated

In[285]:=
```
h = 0.01;
data1 = makeTableImproved[h, 1, 2, 3];
h = 0.005;
data2 = makeTableImproved[h, 1, 2, 3];
exact = Sqrt[#^4 + 8] & /@ data2[[All, 1]];
p = Grid[{
   {"x", Sequence @@ data1[[All, 1]]},
   {"h=0.01", Sequence @@ data1[[All, 2]]},
   {"h=0.005", Sequence @@ data2[[All, 2]]},
   {"exact", Sequence @@ exact},
   {"% error", Sequence @@ ((exact - data2[[All, 2]]) / exact * 100)}
  }, Frame → All]
```

Out[290]=

x	1	1.2	1.4	1.6	1.8	2.
h=0.01	3	3.1739	3.44118	3.81494	4.30091	4.89901
h=0.005	3	3.1739	3.44117	3.81492	4.30089	4.89899
exact	3	3.17389	3.44116	3.81492	4.30088	4.89898
% error	0	-0.0000547372	-0.000101625	-0.000134386	-0.000151819	-0.000156696

To better compare the improved Euler method, with the Euler method, a new table was generated. This gives result only for $h = 0.01$. Here is the result. This used the Euler function which was written for section 2.4 and listed above. The table also includes the difference at each x between the two methods. We see from this table, that as more steps are made (at $x = 2$) that the difference between the improved Euler and Euler method has increased.

In[311]:=
```
h = 0.01;
dataEuler = makeTableEuler[h, 1, 2, 3];
dataImproved = makeTableImproved[h, 1, 2, 3];
p = Grid[{
   {"x", Sequence @@ dataEuler[[All, 1]]},
   {"h=0.01, Euler", Sequence @@ dataEuler[[All, 2]]},
   {"h=0.01, Improved Euler", Sequence @@ dataImproved[[All, 2]]},
   {"Absolute Difference", Sequence @@ (dataEuler[[All, 2]] - dataImproved[[All, 2]])}
  }, Frame → All, Alignment → Left]
```

Out[314]=

x	1	1.2	1.4	1.6	1.8	2.
h=0.01, Euler	3	3.171843	3.436841	3.808392	4.292431	4.88896
h=0.01, Improved Euler	3	3.1739	3.441177	3.814939	4.30091	4.89901
Absolute Difference	0	-0.002057166	-0.004335663	-0.006546691	-0.008478646	-0.01005077

0.7 Section 2.5 problem 25

Problem Apply improved Euler method for $\frac{dv}{dt} = 32 - 1.6v$ with $v(0) = 0$. For $0 \le t \le 2$, using step size $h = 0.01, h = 0.005$, round v to one decimal point. What percentage of limiting velocity 20 ft/sec has been attained after 1 second? After 2 seconds?

Solution The exact solution we derived in section 2.4 above. The improved Euler method, implemented in the function shown above, was used in this problem to generate similar table to section 2.4, problem 25. But now using the improved Euler. Here is the resulting table.

```
data1 = makeTableImproved[0.01, 0, 2, 0];
At one second, using h=0.01 speed is 15.96179 at step n = 100
At 2 seconds, using h=0.01 speed is 19.18464 at step n = 200
data2 = makeTableImproved[0.005, 0, 2, 0];
At one second, using h=0.005 speed is 15.962 at step n = 200
At 2 seconds, using h=0.005 speed is 19.18473 at step n = 400
```

Therefore, improved Euler method result is

h	speed at 1 second	speed at 2 seconds
0.01	15.96179	19.18464
0.005	15.962 (79.81%)	19.18473 (95.923%)

This can be compared with Euler method in problem 2.4.25. We see small difference in speeds at 1 and 2 seconds. The improved Euler result should be taken as the more accurate. Here is the Euler method result, copied from 2.4.25 to make it easier to compare with

h	speed at 1 second	speed at 2 seconds
0.01	16.078	19.206
0.005	16.02 (80.1%)	19.195 (95.98%)

0.8 Section 2.5 problem 26

Problem Deer population $P(t)$ in small forest initially numbered 25 and satisfies logistic equation $\frac{dP}{dt} = 0.0225P(t) - 0.0003P^2$. With t in months. Use improved Euler method to approximate solution for 10 years. First with step $h = 1$ and then with $h = 0.5$ rounding off P to 3 decimal points. What percentage of the limiting population of 75 deer has been attained after 5,10 years?

Solution The improved Euler method

$$\begin{aligned} k_1 &= f\left(x_n, y_n\right) \\ u_{n+1} &= y_n + hk_1 \\ k_2 &= f\left(x_{n+1}, u_{n+1}\right) \\ y_{n+1} &= y_n + h\frac{k_1 + k_2}{2} \end{aligned}$$

With initial conditions $y_0 = 25$ was used to solve this ODE with $f\left(x, y\right) = 0.0225y - 0.0003y^2$. The same improved Euler method function listed earlier was used. The following table summarizes the results

h (moths)	$p(t)$ at 5 years	$p(t)$ at 10 years
1	49.3909 (65.85%)	66.1129 (88.15%)
0.5	49.39135 (65.85%)	66.11343 (88.15%)

In[347]:=

```
(*HW 3, Math 320. By Nasser M. Abbasi. Problem 2.5 26, improved Euler*)
f[t_, y_] := 0.0225 y - 0.0003 y^2;
makeTableImproved[h_, from_, to_, y0_] :=
 Module[{nSteps = Rationalize[(to - from) / h], data, t, y, skip, k1, k2, predictor},
  Array[y, nSteps, 0];
  Array[t, nSteps, 0];
  y[0] = y0; t[0] = from;

  Do[(*Euler loop*)
   k1 = f[t[n], y[n]];
   predictor = y[n] + h k1;
   t[n + 1] = t[n] + h;
   k2 = f[t[n + 1], predictor];
   y[n + 1] = y[n] + h (1 / 2 * (k1 + k2)),
   {n, 0, nSteps}
  ];
  skip = 1; (*Round[0.2/h];*)
  Table[{t[n], y[n]}, {n, 0, nSteps, skip}]
 ]
```