Euler's Method

An initial value problem (IVP) is a differential equation (DE) together with an initial condition (IC):

$$\textbf{IVP} \quad \begin{matrix} y'(t) = f(t, y(t)) & \textbf{DE} \\ y(t_0) = y_0 & \textbf{IC} \end{matrix}$$

Here t_0 is a particular value of "time" at which we have information about the solution y(t); namely the point (t_0, y_0) is on the solution curve.

The objective when solving an IVP is to determine a function y(t) so that

1. **its derivative is the specified expression f(t,y(t))**
2. **when y(t) is evaluated at t = t_0, we get specified value y_0.**

<u>Example</u>

$$\textbf{IVP} \quad \begin{matrix} y'(t) = -3\,y(t) & \textbf{DE} \\ y(0) = 5 & \textbf{IC} \end{matrix}$$

Here we can solve the IVP analytically and get a "closed form" solution (that is, a formula).

$$\frac{dy}{dt} = -3y \Rightarrow \frac{dy}{y} = -3dt \Rightarrow \ln|y| = -3t + C \Rightarrow y = e^{-3t+C} = e^{-3t}e^{C} \Rightarrow y = Ke^{-3t}$$

Use the initial condition: set t = 0 and y = 5; we get K = 5.

Thus the solution to the IVP is $y(t) = 5e^{-3t}$.

++

Unfortunately many DEs cannot be solved analytically hence we need to develop procedures to approximate the solution y(t). **Approximation procedures estimate the value of the solution at a set of discrete points.** **A simple procedure uses a Taylor expansion the over an interval t to t + h to replace the derivative y'(t) as follows; y(t) = p_1(t) + remainder, then solve for y'(t).**

$$y'(t) = \frac{y(t+h) - y(t)}{h} - \frac{1}{2}hy''(\alpha)$$

Then the DE becomes

$$y'(t) = \frac{y(t+h) - y(t)}{h} - \frac{1}{2}hy''(\alpha) = f(t, y(t))$$

Rearranging we get

$$y(t+h) = y(t) + hf(t, y(t)) + \frac{1}{2}h^2y''(\alpha) \qquad (1)$$

This suggests the following "approximation" method.

1. **Omit the term** $\frac{1}{2}h^2y''(\alpha)$.
2. **Start at $t = t_0$ and define a set of "grid" points $t_n = t_0 + nh$, where h is called the stepsize (or mesh spacing or grid size). (Note: n takes on integer values. If $n > 0$ then we forward approximate the solution & for $n < 0$ we backward approximate the solution.)**
3. **Replace Equation 1 by the "difference equation"**

$$y_{n+1} = y_n + hf(t_n, y_n) \qquad (2)$$

Here y_n is an approximation to $y(t_n)$ the true value of the solution of the IVP at $t = t_n$.

Note that to use (2) we start with the initial condition and proceed to develop a set of approximations at the grid points as follows:

$y_0 = y_0 \Rightarrow$ **point** (t_0, y_0) **is on the solution curve**

$y_1 = y_0 + hf(t_0, y_0)$

$\Rightarrow$ **point** $(t_1, y_1) = (t_0 + h, y_1)$ **approximates a point on the solution curve at** t_1

$y_2 = y_1 + hf(t_1, y_1) = y1 + hf(t_0 + h, y_1)$

$\Rightarrow$ **point** $(t_2, y_2) = (t_0 + 2h, y_2)$ **approximates a point on the solution curve at** t_2

ETC.

Thus we have the discrete set of approximations

$$\{(t_j, y_j) \mid j = 0, 1, 2, \ldots, n\}$$

This procedure is known as Euler's Method.

EXAMPLE

IVP $\begin{array}{ll} y'(t) = -y + \sin(t) & \textbf{DE} \\ y(0) = 1 & \textbf{IC} \end{array}$ Note $t_0 = 0$ & $y_0 = 1$.

Use Euler's method over the interval [0, 1] with stepsize h = ¼.
Hence we have grid points $t_0 = 0$, $t_1 = 0.25$, $t_2 = 0.5$, $t_3 = 0.75$, $t_4 = 1$.

$y_0 = 1$

$y_1 = y_0 + h\, f(t_0, y_0) = 1 + 0.25\,(-1 + \sin(0)) = 0.75$

$y_2 = y_1 + h\, f(t_1, y_1) = 0.75 + 0.25\,(-0.75 + \sin(0.25)) \approx 0.624350$

$y_3 = y_2 + h\, f(t_2, y_2)$
$= 0.6243505 + 0.25\,(-0.624350 + \sin(0.5)) \approx 0.588119$

$y_4 = y_3 + h\, f(t_3, y_3)$
$= 0.588119 + 0.25\,(-0.588119 + \sin(0.75)) \approx 0.611499$

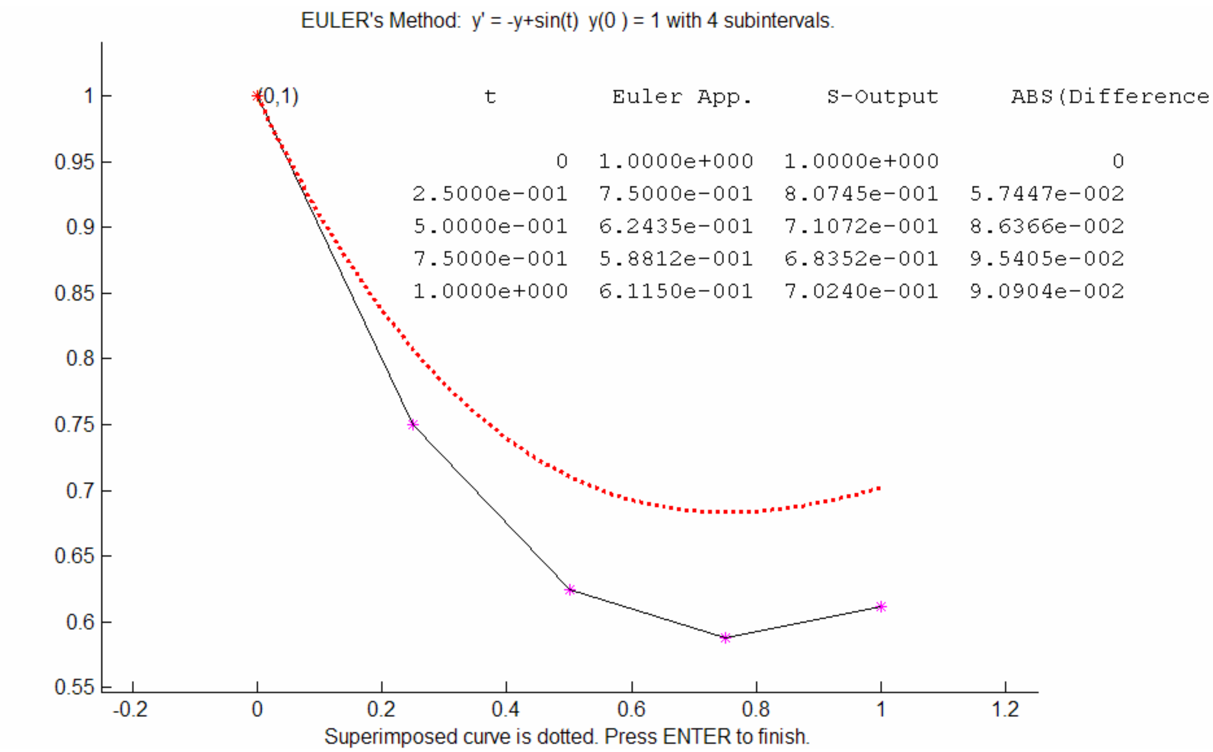
What errors were encountered in this process? Explain.

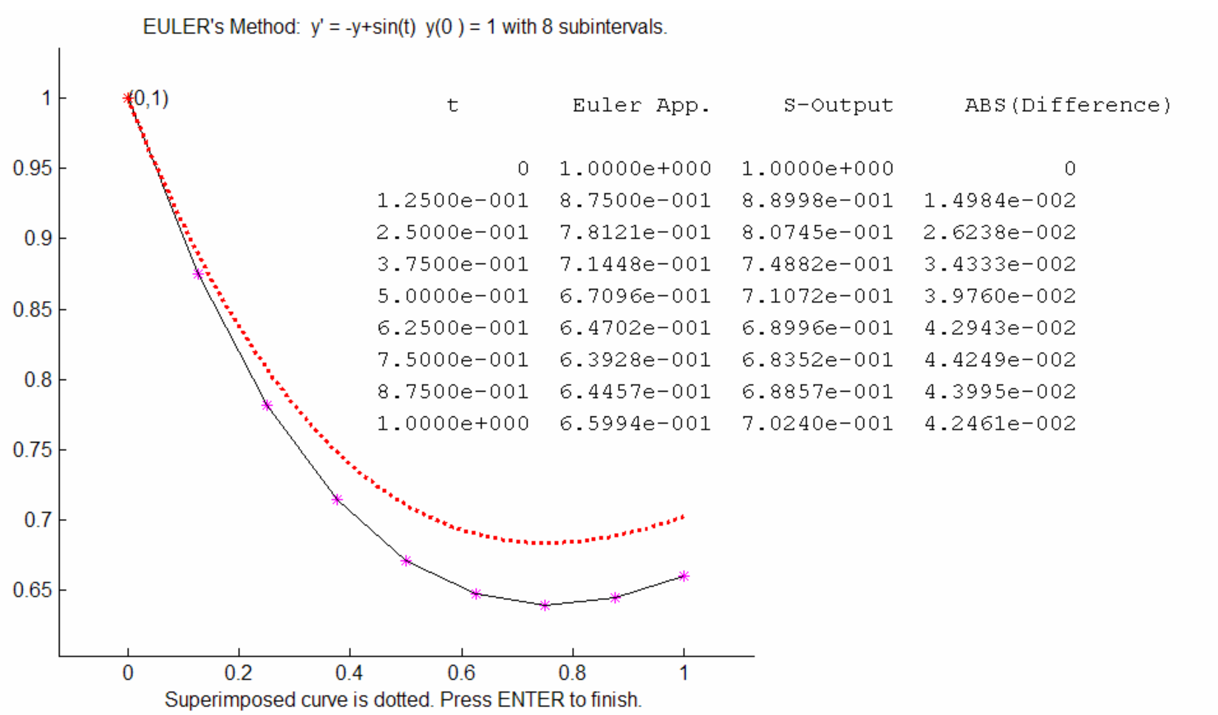
Note: True solution is $y(t) = 1.5\, e^{-t} + 0.5\,(\sin(t) - \cos(t))$

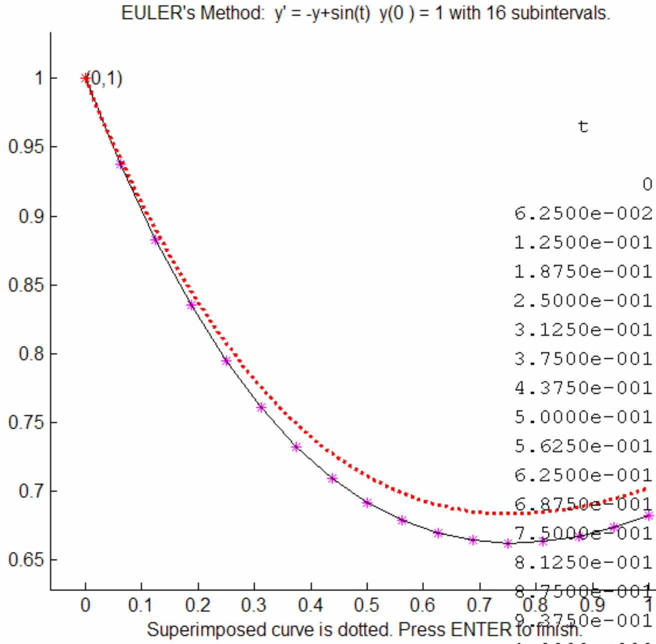
Summary:

T	Euler Approx.	True Solution	Absolute Error
0	1	1	0
0.25	0.75	0.80745	0.057447
0.5	0.62435	0.71072	0.086366
0.75	0.58812	0.68352	0.095405
1	0.6115	0.7024	0.090904

Compare the absolute errors in the figures below where we have decreased the step size h.


EULER's Method: y' = -y+sin(t) y(0) = 1 with 4 subintervals.
(0,1)
t Euler App. S-Output ABS(Difference)
0 1.0000e+000 1.0000e+000 0
2.5000e-001 7.5000e-001 8.0745e-001 5.7447e-002
5.0000e-001 6.2435e-001 7.1072e-001 8.6366e-002
7.5000e-001 5.8812e-001 6.8352e-001 9.5405e-002
1.0000e+000 6.1150e-001 7.0240e-001 9.0904e-002
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
Superimposed curve is dotted. Press ENTER to finish.


EULER's Method: y' = -y+sin(t) y(0) = 1 with 8 subintervals.
(0,1)
t Euler App. S-Output ABS(Difference)
0 1.0000e+000 1.0000e+000 0
1.2500e-001 8.7500e-001 8.8998e-001 1.4984e-002
2.5000e-001 7.8121e-001 8.0745e-001 2.6238e-002
3.7500e-001 7.1448e-001 7.4882e-001 3.4333e-002
5.0000e-001 6.7096e-001 7.1072e-001 3.9760e-002
6.2500e-001 6.4702e-001 6.8996e-001 4.2943e-002
7.5000e-001 6.3928e-001 6.8352e-001 4.4249e-002
8.7500e-001 6.4457e-001 6.8857e-001 4.3995e-002
1.0000e+000 6.5994e-001 7.0240e-001 4.2461e-002
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0
0.2
0.4
0.6
0.8
1
Superimposed curve is dotted. Press ENTER to finish.



t	Euler App.	S-Output	ABS(Difference
0	1.0000e+000	1.0000e+000	0
6.2500e-002	9.3750e-001	9.4133e-001	3.8255e-003
1.2500e-001	8.8281e-001	8.8998e-001	7.1739e-003
1.8750e-001	8.3543e-001	8.4551e-001	1.0082e-002
2.5000e-001	7.9486e-001	8.0745e-001	1.2584e-002
3.1250e-001	7.6065e-001	7.7536e-001	1.4712e-002
3.7500e-001	7.3232e-001	7.4882e-001	1.6495e-002
4.3750e-001	7.0944e-001	7.2740e-001	1.7961e-002
5.0000e-001	6.9158e-001	7.1072e-001	1.9135e-002
5.6250e-001	6.7832e-001	6.9836e-001	2.0041e-002
6.2500e-001	6.6926e-001	6.8996e-001	2.0700e-002
6.8750e-001	6.6400e-001	6.8513e-001	2.1135e-002
7.5000e-001	6.6216e-001	6.8352e-001	2.1363e-002
8.1250e-001	6.6338e-001	6.8478e-001	2.1403e-002
8.7500e-001	6.6729e-001	6.8857e-001	2.1273e-002
9.3750e-001	6.7356e-001	6.9455e-001	2.0987e-002
1.0000e+000	6.8184e-001	7.0240e-001	2.0562e-002

Inspection of the errors suggests (does not prove) that Euler's Method is $O(h)$.

Important discussion of the error in Euler's Method

There can be a slow but steady growth in the global error as t increases. Since each step introduces new error into the computed approximate solution, we might expect this type of behavior in every problem; however, the actual accumulation of global error is very problem dependent. Essentially, the error introduced by each step of the time marching process moves us from one solution of the differential equation onto a different solution. If, as in the previous example, nearby solutions separate from one another as t increases, we can expect to see a steady increase in the global error. On the other hand, if nearby solutions move closer together as t increases, we could expect to observe a steady decline in the global error.

In MATLAB

eulerh

>> help eulerh

EULERH Euler's method for the approximation of an initial
value problem y' = f(t,y) on [a,b]
y(a) = alpha
using n subintervals of equal length.
Function f must be defined as a string with
independent variable t and dependent variable y.
A graph of the piece-wise linear approximation is shown.

There is tabular output of t versus z(t), where z(t)
approximates y(t), the true solution.

f is a string containing the right side of the ode
in terms of t and y
a is left end of interval of solution
b is right end of interval of solution
alpha is value of y at t = a
n is the number of subintervals of [a,b] to use

Use in the following forms:
==> eulerh(f,a,b,alpha,n) <== Displays the Euler graph all at once.

==> eulerh(f,a,b,alpha,n,'S') <== Displays the Euler graph step-by-step.

EXAMPLE: eulerh('y-t^2+1',0,2,0.5,10)

The IVP is y'(t) = y - t*t+ 1, y(0) = 1/2 over [0,2]

++