www.simiode.org
SIMIODE
A SYSTEMIC INITIATIVE FOR MODELING INVESTIGATIONS & OPPORTUNITIES WITH DIFFERENTIAL EQUATIONS

STUDENT VERSION
Introduction to Numerical Methods

Brian Winkel, SIMIODE
Chardon OH 44024 USA
BrianWinkel@simiode.org

Abstract: We develop elementary approaches to numerically solving first order differential equations with Euler's Method and Improved Euler's Method. We ask students to develop these geometrically and to compute numeric solutions and compare them to the analytic solutions for a number of different step sizes.

INTRODUCTION

Introduction

We try to solve differential equations analytically (i.e. obtain a closed form solution) for the most part by attempting to maneuver the equation into a position where we can "see" an antiderivative emerge and then jump on the opportunity by integrating both sides of the differential equation. This solution strategy, while difficult at times, offers us a nice formula with structure we can examine and tells us where our parameters are involved. For example the analytic solution for the exponential growth equation (with $k > 0$),

$$y'(t) = ky(t)\,, \quad y(0) = y_0\,, \tag{1}$$

is simply $y(t) = y_0 e^{k\,t}$. We see that k is the exponent and if we make k positive and bigger then $y(t)$ will grow faster. It is all right in front of us when we can analytically solve. Our emphasis is on modeling and we will delegate analytic solutions, for the most part, to *Mathematica*'s `DSolve` command or other software commands. Of course, other comparable software packages have their own differential equation solving commands. However, we will find out that even a powerful computer algebra system will not always be able to pull a solution for $y(t)$ out of the proverbial hat. So, it may not be possible to solve the differential equation analytically (or in closed form) or it might be just beyond the software's integration skills or our own by-hand strategies and methods. Indeed, there exist differential equations on which mathematicians have spent considerable effort proving that there exist no analytic solutions. For example consider

$$y'(t) = e^{-t^2}\,, \qquad y(0) = y_0\ . \tag{2}$$

Perhaps in your calculus class a wise instructor challenged students to integrate the function e^{-t^2} which is necessary to solve (2). Integrating this function is something statisticians often need to do, for if you plot the function $x(t) = e^{-t^2}$ you will see the familiar bell-shaped curve which describes many phenomena in the world and is a simplified form of the *normal distribution* used in probability and statistics courses. However, there are proofs that no anlaytic or closed form solution exists for (2).

What if there is no analytic solution?

So what do we do in case we cannot find an analytic solution to a differential equation,

$$y'(t) = f(t, y(t))\,, \qquad y(0) = y_0\ , \tag{3}$$

where $f(t, y(t))$ is perhaps some "wild" function of t and $y(t)$? We revert to how we got the differential equation in the first place, i.e. the derivative $y'(t)$ represents some rate of change,

$$y'(t) = \lim_{h\to 0} \frac{y(t+h) - y(t)}{h}\,. \tag{4}$$

Recall that for very small values of h the ratio $\dfrac{y(t+h) - y(t)}{h}$ is approximately the derivative, $y'(t)$. So perhaps if we used this approximation and kept h small we could render an approximation, maybe even a good approximation to the solution for (5). Let us do just that by attempting to solve the *difference* equation (5) approximation to the actual *differential* equation (3).

$$\frac{y(t+h) - y(t)}{h} = f(t, y(t))\,, \quad y(0) = y_0\,. \tag{5}$$

We will attempt to develop a scheme and apply it to a differential equation whose solution we actually know to see just how good our scheme really is and how changing the size of h, often referred to as the *step size*, alters our accuracy. We consider this simple differential equation:

$$y'(t) = .05 * y(t)\,, \qquad y(0) = 100\,. \tag{6}$$

One interpretation for this model is growth on a continuously compounding bank account paying annual interest of 5% with a \$100 initial deposit. We know the solution to this differential equation, by separating the variables and integrating, is $y(t) = 100\, e^{0.05t}$.

Developing an approximation scheme for solving numerically

We now turn to developing an approximation scheme for (5). Scientists will often develop algorithms or test their theories in known situations, minimizing the effects of extraneous variables. This is known as solving a *toy problem* in mathematics. Figure 1 might give you some ideas.

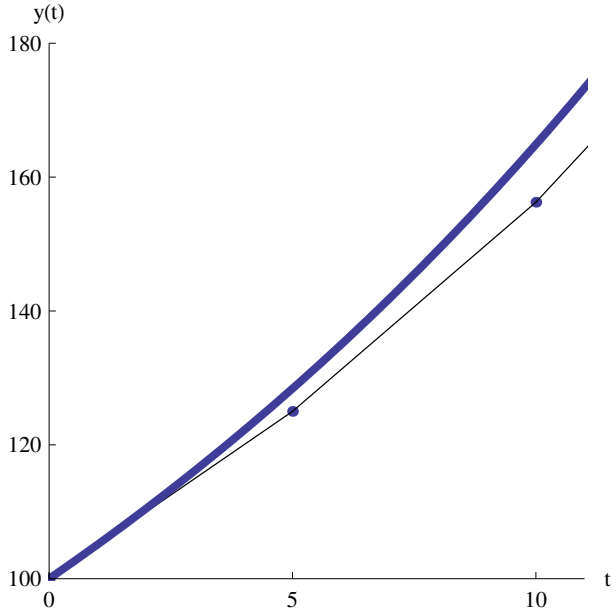

Figure 1. Possible estimate plot for solution of differential equation (1). The dashed curve is the analytic solution to (1) while the linear piecewise curve is the graphical summary of Euler's Method, treated below, with estimated points indicated and connections drawn.

a) Starting at point $(0, 100)$ for \$100 at time $t = 0$ begin numerically estimating the solution to the difference equation (7). h is known as the *step size* of our iterations.

$$\frac{y(t+h) - y(t)}{h} = .05 * y(t)\,, \qquad y(0) = 100\ . \tag{7}$$

Perhaps the formulation in (8) where we solve for $y(t+h)$ will help. If we know a value or an approximation for the function $y(t)$, in this case at $t = 0$, can we use (8) to estimate $y(t)$ at $t = 5$? Here $h = 5$. We solve (7) for $y(t+h)$ to find:

$$y(t+h) = y(t) + h * 0.05 * y(t)\,. \tag{8}$$

Can we use our knowledge of slope and line to estimate $y(5)$? Indeed, this first iteration is akin to moving along the tangent line at $(0, 0)$ with slope $y'(0)$ until the estimated value of $y(x)$ at $x = 5$ is obtained. We shall call this estimate $\hat{y}(5)$ (pronounced "y hat at .5") to denote an estimate. Write out the exact arithmetic necessary to obtain $\hat{y}(5)$. Compare this to the known solution in our case, i.e. $y(5) = 100e^{.05*5} = \$128.40$. Identify the relevant points on the plot in Figure 1 for this new found estimate and for the actual value of our solution to the differential equation (1). Write out your thoughts about the comparison, including your observations of what is in Figure 1.

b) Now we have an estimate, $\hat{y}(5)$, of the solution of (7) at $t = 5$. Enter your results in Table 1. Then use this new found value to estimate the solution to (7) at $t = 10$. Write out explicitly

the arithmetic statement that will get you from $\hat{y}(5)$ to $\hat{y}(10)$ using $h = 5$, being sure to identify each term in the situation in terms of y, t and h. Also identify where the estimate is on the plot in Figure 1.

c) Use your estimate, $\hat{y}(10)$, to render an estimate to the solution of the differential equation (7) at $t = 15$, i.e. compute $\hat{y}(15)$. Compare your estimate, using $h = 5$, to the actual value of the solution of the differential equation.

d) Now do several more computations to fill out Table 1. Also make your own plot of the actual solution and the estimates you have computed. For the former use *Mathematica*'s `Plot` command and label the plot `SolutionPlot` and for the latter use `ListPlot` of your data set and label the data plot `DataPlot`. Finally, `Show[SolutionPlot, DataPlot]` to see your estimates in comparison with the analytic solution. Comments are in order. What can you say about what is happening and why it is happening?

t	0	5	10	15	20	25
$\hat{y}(t)$	\$100.00					
$y(t)$	\$100.00					

Table 1. Your estimates for the solution of differential equation (8) for several values of t.

We can summarize what we have done with (5) in one general formula as shown in (9)

$$\underbrace{\hat{y}(t+h)}_{\text{new}} = \underbrace{\hat{y}(t)}_{\text{old}} + \underbrace{h * f(t, \hat{y}(t))}_{\text{change}}. \tag{9}$$

We simply find a new estimate, $\hat{y}(t+h)$ by adding the change $f(t, \hat{y}(t)) * h$ to the old estimate $\hat{y}(t)$. This method is an old one and is known as *Euler's Method*, after Leonhard Euler [1], the great (no make that awesome!) Swiss mathematician of the 18^{th} century.

d) Here is some *Mathematica* code that will help you play with this situation. We use $h = 0.1$ to gain more accuracy and we iterate for 300 times to solve our differential equation (8) over the interval $[0, 300 * 0.1 = 30]$:

```
h = 0.1;
data = {{0, 100}};
Do[AppendTo[data,
      {data[[i - 1, 1]] + h, data[[i - 1, 2]] + .05 data[[i - 1, 2]]*h}],
{i, 2, 300}]
```

An explanation of this sequence of commands in *Mathematica* is in order.

- `h = 0.1` is our step size in our estimating process.
- `data` is where we store our successive estimates - two coordinates for each data point, $(t_i, \hat{y}(t_i))$ for the i^{th} estimate.

- We `Append` our new estimate to our data set.
- `data[[i - 1, 1]]` is the first coordinate of the $(i-1)^{\text{th}}$ data point (we start at $i = 2$) while `data[[i - 1, 2]]` is the second coordinate of the $(i-1)^{\text{th}}$ data point and `.05 data[[i - 1, 2]]*h` is the change at t_i.
- We do this for i from 2 to 300 with step size $h = 0.1$, which will get us to $t = 0.1 * 300 = 30$ years.

Of course we can change h, but then we need to change the number of iterations to be sure we go as far along as we really want.

Use this code and your plotting expertise to estimate the value $\hat{y}(30)$. Try values as big as $h = 30$ and as small as $h = 0.001$. Hint: Change {`i, 2, 30`} to {`i, 2, 30/h`} to automatically account for just how many estimates are necessary to compute to get to $t = 30$ with this given value of h. Render a plot of these estimates for a reasonable set of values of h and then for each h selected plot the analytic solution and the estimate data on the same axes. Comment on what these plots show, say for $h = 0.001$ and $h = 10$.

e) Summarize what we have just learned in part (d). Try to think beyond this one example to how this method might apply to other equations. Visualize the geometry of the estimating process of the straight line segments used in the estimating and of the error when we arrive at the time we want, in this case $t = 30$, in terms of the step size h we elect.

f) We take some time to get a handle on just how good this process is at estimating solutions to differential equations. Consider the following differential equations with initial conditions:

i) $y'(t) = 0.03y(t)\,, \quad y(0) = 10.$

ii) $y'(t) = 0.003y(t)^2\,, \quad y(0) = 2.$

iii) $y'(t) = 0.05y(t) + 3\,, \quad y(0) = 10.$

iv) $y'(t) = 5 - t + .02y(t)\,, \quad y(0) = 1.$

In each case solve the differential equation analytically (use *Mathematica*'s `DSolve` command) and compute specifically the value of $y(10)$. Then, using $h = 0.5$, 0.25, 0.2, 0.1, 0.05, and 0.01 as step sizes, compute estimates, $\hat{y}(10)$ using Euler's Method and compare the answer you got with the analytic solution; a plot of the analytic solution along with the data points for each increment h would be in order. Just how good an approximation Euler's Method is can be assessed by examining how good it is at predicting a specific value of our function, say, $\hat{y}(10)$ for $y(10)$. Fill in Table 2 and come to some conclusions as to the accuracy in Euler's Method.

For each differential equation, (i), (ii), (iii), and (iv), plot the error between the estimate and the actual value of the solution at $t = 10$ vs. step size h. What can we infer from these plots? State this as a mathematical conjecture.

g) Now use this new method to approximate the solution of the differential equation (10) over the time interval $t \in [0, 4]$:

h	$y(10)$	$\hat{y}(10)$	$y(10) - \hat{y}(10)$
0.50			
0.25			
0.20			
0.10			
0.05			
0.01			

Table 2. A summary of errors for the numerical solution using Euler's Method for one of the four ((i), (ii), (iii), or (iv)) differential equations for various step sizes h in computing $\hat{y}(10)$.

$$y'(t) = 0.2\, y(t) \frac{(10 + 5\sin(25t) - y(t))}{100 + 5\sin(25t)}\,, \quad y(0) = 50\,. \tag{10}$$

and plot your solution. Of course, you might try to `DSolve`, but we do not hold out much hope that an analytic solution for (10) can be found. We might then consider a numerical solution with *Mathematica*'s `NDSolve`, a very powerful, advance numerical solution technique which gives a comparable function (only as a list of points and interpolations). Compare your results from Euler's Method with *Mathematica*'s `NDSolve` solution. Also, consider what this differential equation might model.

h) Show that a concise formula for Euler's Method with step size h for approximating the solution of the differential equation $y'(t) = f(t, y(t))$ with initial condition $y(t_0) = y_0$ is given by (11):

$$y(t_{k+1}) = y(t_k) + h * f(t_k, y(t_{k+1}))\,, \tag{11}$$

where $t_k = t_0 + k * h$, $k = 0, 1, 2, \ldots,$.

One way to do this is to convince yourself, with pictures perhaps, that

$$y(t_1) = y(t_0) + h * f(t_0, y(t_0))\,,$$

and then that

$$y(t_2) = y(t_1) + h * f(t_1, y(t_1))\,,$$

as starters.

What we are doing here is repeatedly taking tangent line predictions at each iteration from $t = t_k$ to $t = t_{k+1}$ and, apparently, getting worse and worse results at each step.

Improving on Euler's Method

Let us return to Figure 1 and see if we can improve on our approximation strategy. In Figure 1 we see the actual solution to the differential equation (12) (curved line)

$$y'(t) = 0.05 * y(t)\,, \qquad y(0) = 100\ , \tag{12}$$

and the stepwise approximations from $t = 0$ to $t = 5$ and thence to $t = 10$.

Notice that when we go from the point $(5, \hat{y}(5))$ to $(10, \hat{y}(10))$ our accuracy in our estimation gets worse. However, if we were to connect the initial point $(0, 100)$ to the estimate point $(10, \hat{y}(10))$ to form a line segment and then take the estimate at $t = 5$ to be the value on this line segment at $t = 5$ we can see that we would have a better estimate of $y(5)$. Call this new estimate $\overline{y}(5)$. In effect, we are using the estimate $\hat{y}(10)$ to tell our estimation process to get ready to correct for the sharper turn between $(5, \hat{y}(5))$ and $(10, \hat{y}(10))$.

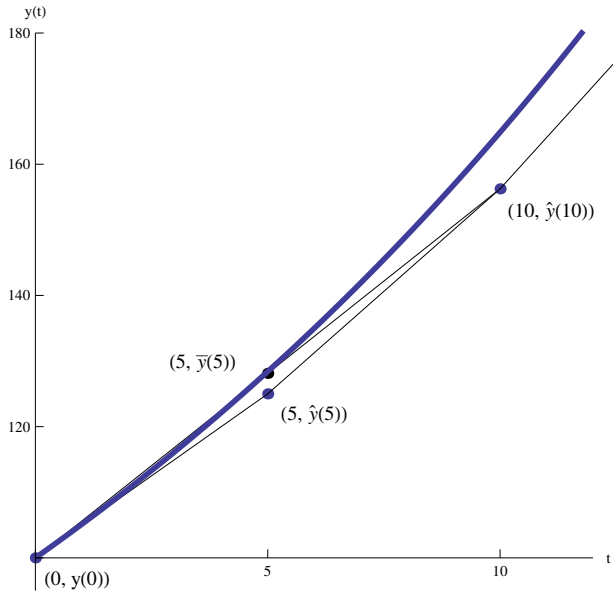

Figure 2. Improved Euler's Method estimate at $t = 5$ for the solution of differential equation (12). The dashed curve is the analytic solution to (12); the linear piecewise curve is the graphical summary of Euler's Method showing estimated points, $(5, \hat{y}(5))$ and $(10, \hat{y}(10))$, with connections drawn and solution $(5, \hat{y}(5))$; and the dotted line is the intermediate step in computing the Improved Euler's Method and solution $(5, \overline{y}(5))$.

i) Use the point slope method for determining the equation of a linear function $y = g(t)$ connecting $(0, 100)$ to the estimate point $(10, \hat{y}(10))$ and then evaluate $g(5)$ as an improved estimate for our differential equation (12) at $t = 5$. Call this improved estimate, $\overline{y}(t)$. Compare $y(5)$, $\hat{y}(5)$, and $\overline{y}(5)$. Is it really improved?

Here is another, less complicated way to view the situation and arrive at a formula for the Improved Euler's Method. Let us average the two slopes (1) m_1 the slope of the line segment from $(0, y(0))$ to our point at which we use Euler's Method for a first estimate, $(5, \hat{y}(5))$, and (2) m_2, the slope of the line segment from $(5, \hat{y}(5))$ to the second Euler's Method estimate $(10, \hat{y}(10))$ (since the horizontal step size in each segment is the same, namely $h = 5$). Call this average slope M_1. If we then go from our initial point $(0, y(0)$ toward the point $(10, \hat{y}(10))$ along

a line of this new slope, M_1, we will arrive at our Improved Euler's Method point of $(5, \overline{y}(5))$. Here is the formula for this Improved Euler's Method and we ask you to explain each step of the expressions:

$$\overline{y}(5) = y(0) + h\frac{1}{2}(m_1 + m_2)\,, \text{ here } h = 5\,. \tag{13}$$

Hint: We suggest you compute m_1 and m_2 first.

j) Build a concise formula for this Improved Euler's Method with step size h for approximating the solution of the differential equation $y'(t) = f(t, y(t))$ with initial condition $y(t_0) = y_0$ and use your results to fill in (14):

$$y(t_{k+1}) = y(t_k) + \underline{\qquad\qquad\qquad\qquad}\,, \tag{14}$$

where $t_k = t_0 + k * h$, $k = 0, 1, 2, \ldots,$. Figure 3 might help, as it depicts the general step from estimate $\overline{y}(t_k)$ to the next estimate $\overline{y}(t_{k+1})$.

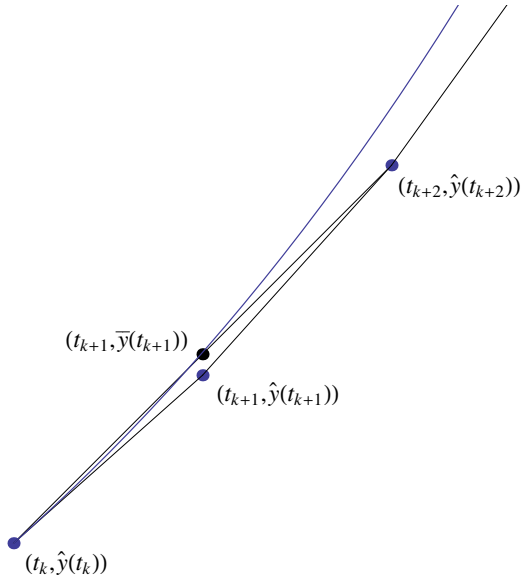

Figure 3. General diagram to help develop iteration formula for Improved Euler's Method estimate of solution at $(t_{k+1}, \overline{y}(t_{k+1})$ from that at t_k for solution of differential equation (12). Also shown is the Euler's Method estimate for solution at $(t_{k+1}, \hat{y}(t_{k+1}))$.

k) Summarize what you have just learned in part (i) - (j). Try to think beyond this one example to how this method might apply to other differential equations. Visualize the geometry of the estimating process via the straight line segments and the corrections used in the estimating process, as well as the error when we change the selected step size h.

Let us now take some time to get a handle on just how good this process is at estimating solutions to differential equations. Consider the following differential equations with initial conditions:

i) $y'(t) = 0.03y(t)\,, \quad y(0) = 10.$

ii) $y'(t) = 0.003y(t)^2\,, \quad y(0) = 2.$

iii) $y'(t) = 0.05y(t) + 3\,, \quad y(0) = 10.$

iv) $y'(t) = 5 - t + 0.02y(t)\,, \quad y(0) = 1.$

In each case solve the differential equation analytically (use *Mathematica*'s `DSolve` command) and compute specifically the value of $y(10)$. Then using $h = 0.5$, 0.25, 0.2, 0.1, 0.05, and 0.01 as step sizes, compute estimates, $\overline{y}(t)$ using Improved Euler's Method. Then compare the answer you got with the analytic solution; a plot of the analytic solution along with the data points for each increment h would be in order. How well the Improved Euler's Method performs can be assessed by examining how well it predicts a specific value of our function, say, $y(10)$. Fill in Table 3 and come to some conclusions as to the accuracy in te Improved Euler's Method.

h	$y(10)$	$\hat{y}(10)$	$y(10) - \hat{y}(10)$
0.50			
0.25			
0.20			
0.10			
0.05			
0.01			

Table 3. A summary of errors for the numerical solution using the Improved Euler's Method for one of the four ((i), (ii), (iii), or (iv)) differential equations for various step sizes h in computing $\hat{y}(10)$.

For your differential equations (i), (ii), (iii), or (iv) plot the error between the estimate and the actual value of the solution at $t = 10$ vs. step size h. What can you infer from these plots? State this as a mathematical conjecture.

l) Compare your results and your conclusions about the Euler's Method (Table 2) and Improved Euler's Method (Table 3). Offer a general mathematical summary and compare Euler's Method and the Improved Euler's Method.

m) Now use the Improved Euler's Method to solve the following differential equation (15):

$$y'(t) = .2y(t)\frac{(10 + 5\sin(25t) - y(t))}{100 + 5\sin(25t)}\,, \quad y(0) = 50\,. \tag{15}$$

and plot your solution, comparing it with the solution you got using Euler's Method in (g).

Advanced Numerical Methods

There exist some very sophisticated algorithms for estimating solutions for differential equations. Indeed, some corporations and individuals make very good profits from producing computer code

to approximate solutions to these differential equations, often with hundreds, sometimes thousands, of differential equations in the systems they seek to analyze.

We could pursue additional methods here. For example the Runge-Kutta approach is quite good. There also exist varying step size algorithms, which essentially take large steps when the derivative, $y'(t)$, is small and smaller step sizes when the derivative, $y'(t)$, is large. Why would one want to vary step sizes in this way?

Mathematica has a very powerful and accurate numerical solution technique for solving differential equations numerically, which is invoked with the command `NDSolve` as opposed to `DSolve`, which gives the analytic solution if one exists – well, if it exists in the strategies *Mathematica* employs to integrate!

`NDSolve` has lots of variations in itself, but its basis is a sophisticated algorithm which renders an *Interpolating Function*, i.e. a set of points with interpolation between these points. In general, `NDSolve`'s capabilities are so good that once we have picked off the numerical solution to a differential equation from `NDSolve` output, we can do just about anything we want with this interpolating function, e.g., differentiate, integrate, plot, use in other equations, etc. What we cannot see from the output of `NDSolve` is a nice closed form solution with parameters visible (which `DSolve` does offer); `NDSolve` renders the solution as a set of numbers and rules for interpolating, but it is a solution that we can use with confidence.

n) This activity is to show that *Mathematica*'s `NDSolve` command is very good in returning an interpolation function. In fact, it is so good that you can differentiate and integrate the resulting interpolating function very successfully.

Compare plots of the two solution strategies, `DSolve` and `NDSolve`, for the differential equation (16).

$$y'(t) = 12 - 5\sin(3t) + y(t)\,, \quad y(0) = 3. \tag{16}$$

Here is the *Mathematica* code to solve and pick out the solution of (16) using `DSolve`

```
yd[t_ ] = y[t] /. DSolve[{y'[t] == 12 - 5 Sin[3 t] + y[t],
                 y[0] == 3}, y[t], t][[1]]
```

Here is the *Mathematica* code to solve and pick out the solution of (16) using `NDSolve`

```
ynd[t_ ] = y[t] /. NDSolve[{y'[t] == 12 - 5 Sin[3 t]
                  + y[t], y[0] == 3}, y[t], {t, 0, 3}][[1]]
```

Conclusion

We now have more differential equation solving capability with an understanding of the difference between `DSolve` and `NDSolve` and some idea when to use which command. More importantly, we have a better understanding of how computers numerically solve differential equations which they cannot solve analytically.

REFERENCES

[1] Wikipedia. 2010. *Leonhard Euler*. `http://en.wikipedia.org/wiki/Leonhard_Euler`. Accessed 7 January 2019.