DIFFERENTIAL EQUATION NUMERICAL METHODS 1

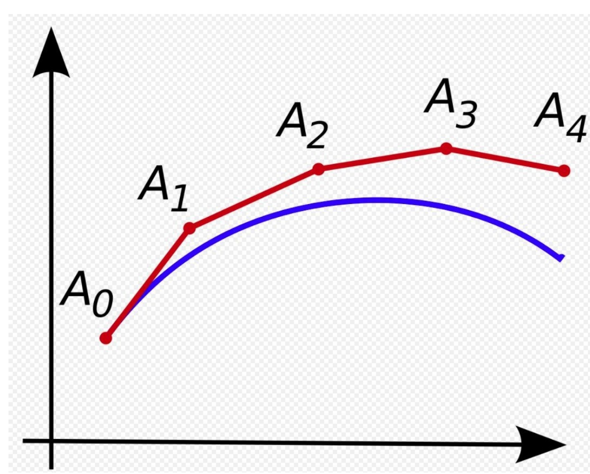
These are methods for approximating DE numerically. That is, when you cannot solve one for all variables, this will give an approximate numerical solution. The caveat for this is that you need an initial value. Most of the time, in the real world, this is not difficult.
*These equations are mostly used for physical problems, ie, astronomical predictions, biological systems, chemical reactions, etc. A researcher should know the initial conditions, location, population numbers, etc.
*The accepted symbol for the independent variable is "t." This is because for most uses of DE, time is the independent variable. Whereas, in mathematics, this variable is normally "x."

We will begin with the most basic algorithm, Euler. While this method is very simple, it presents the format that all other methods follow. Euler is a one step, or starter method. I will describe this method in detail, because the remaining methods were developed from Euler.
*As a reminder, we will check the accuracy of our calculations with "actual" calculations. Very simply, this is a solution using the initial variable values inserted into the actual DE solution (Important note, do not put these numbers into the given DE, you need to find the solution and use that). Therefore, we will need to know this real solution in order to determine the accuracy of our predictive algorithms. From these, we can decide which method will be appropriate for future problems. (See equation (04) p 5, below).

EULER

The Euler method is a first-order numerical procedure for solving ordinary differential equations with a given initial value. It is the most basic explicit method for numerical integration of ordinary differential equations and is the simplest Runge–Kutta method. The Euler method is a first-order method, which means that the local error (error per step) is proportional to the square of the step size, and the global error (error at a given time) is also proportional to the step size. The Euler method often serves as the basis to construct more complex methods.

Consider the problem of calculating the shape of an unknown curve which starts at a given point and satisfies a given differential equation. Here, a differential equation can be thought of as a formula by which the slope of the tangent line to the curve can be computed at any point on the curve, once the position of that point has been calculated.



The idea is that while the curve is initially unknown, its starting point, which we denote by A_0 is known. Then, from the differential equation, the slope to the curve at A_0 can be computed, and so, the tangent line.

Take a small step along that tangent line up to a point A_1. Along this small step, the slope does not change too much, so A_1 will be close to the curve. If we pretend that A_1 is still on the curve, the same reasoning as for the point A_0 above can be used. After several steps, a polygonal curve A_0 A_1 A_2 … is computed. In general, this curve does not diverge too far from the original unknown curve, and the error between the two curves can be made small if the step size is small enough and the interval of computation is finite. (Wikipedia, Euler Method).

Up to this point practically every differential equation that we've been presented with could be solved. The problem with this is that these are the exceptions rather than the rule. The vast majority of first order differential equations can't be solved. (Dawkins, 101).

In order to teach you something about solving first order differential equations we've had to restrict ourselves down to the fairly restrictive cases of linear, separable, or exact differential equations or differential equations that could be solved with a set of very specific substitutions. Most first order differential equations however fall into none of these categories. In fact, even those that are separable or exact cannot always be solved for an explicit solution. Without explicit solutions to these it would be hard to get any information about the solution.

So, what do we do when faced with a differential equation that we can't solve? The answer depends on what you are looking for. If you are only looking for long term behavior of a solution you can always sketch a direction field. This can be done without too much difficulty for some fairly complex differential equations that we can't solve to get exact solutions.

The problem with this approach is that it's only really good for getting general trends in solutions and for long term behavior of solutions. There are times when we will need something more. For instance, maybe we need to determine how a specific solution behaves, including some values that the solution will take. There are also a fairly large set of differential equations that are not easy to sketch good direction fields for.

In these cases, we resort to numerical methods that will allow us to approximate solutions to differential equations. There are many different methods that can be used to approximate solutions to a differential equation and in fact whole classes can be taught just dealing with the various methods. We are going to look at one of the oldest and easiest to use here. This method was originally devised by Euler and is called, oddly enough, Euler's Method.

Let's start with a general first order IVP

$$\frac{dy}{dx} = f(t,y) \qquad y(t_0) = f(t_0) \tag{01}$$

where f(t,y) is a known function and the values in the initial condition are also known numbers. From the second theorem in the Intervals of Validity section (Dawkins, 79), we know that if f and f_y are continuous functions then there is a unique solution to the IVP in some interval surrounding $t = t_0$. So, let's assume that everything is nice and continuous so that we know that a solution will in fact exist.

We want to approximate the solution to (01) near $t = t_0$. We'll start with the two pieces of information that we do know about the solution. First, we know the value of the solution at $t = t_0$ from the initial condition. Second, we also know the value of the derivative at $t = t_0$. We can get this by plugging the initial condition into f(t,y) into the differential equation itself. So, the derivative at this point is

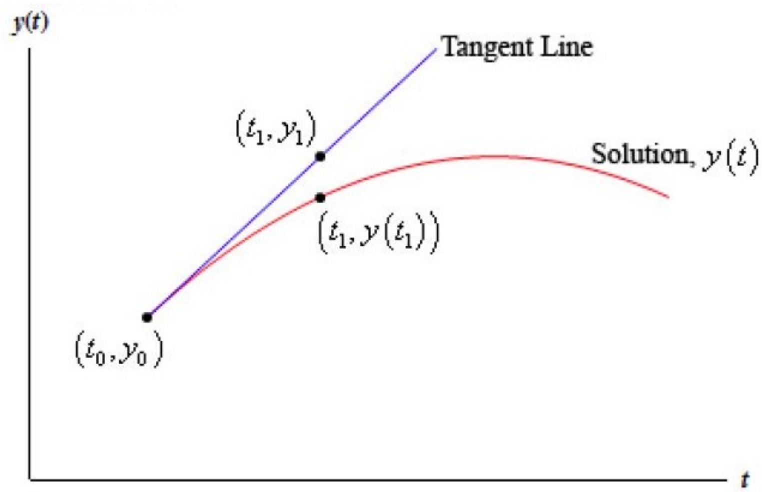
$$\left.\frac{dy}{dx}\right|_{t=t_0} = f(t_0,y_0)$$

Now, recall from Calculus I that these two pieces of information are enough for us to write down the equation of the tangent line to the solution at $t = t_0$. The tangent line is

$$y = y_0 + f(t_0,y_0)(t - t_0)$$

If t_1 is close enough to t_0 then the point y_1 on the tangent line should be fairly close to the actual value of the solution at t_1 or $y(t_1)$. Finding y_1 is easy enough. All we need to do is plug t_1 in the equation for the tangent line:

$$y = y_0 + f(t_0,y_0)(t - t_0)$$



Now, we would like to proceed in a similar manner, but we don't have the value of the solution at t_1 and so we won't know the slope of the tangent line to the solution at this point. This is a problem. We can partially solve it however, by recalling that y_1 is an approximation to the solution at t_1. If y_1 is a very good approximation to the actual value of the solution then we can use that to estimate the slope of the tangent line at t_1.
So, let's hope that y_1 is a good approximation to the solution and construct a line through the point (t_1,y_1) that has slope $f(t_1,y_1)$. This gives:

$$y = y_1 + f(t_1,y_1)(t - t_1)$$

Now, to get an approximation to the solution at $t = t_2$ we will hope that this new line will be fairly close to the actual solution at t_2 and use the value of the line at t_2 as an approximation to the actual solution.

$$y_2 = y_1 + f(t_1,y_1)(t_2 - t_1)$$

We continue in this fashion. Use the previously computed approximation to get the next.

$$y_3 = y_2 + f(t_2,y_2)(t_3 - t_2)$$
$$y_4 = y_3 + f(t_3,y_3)(t_4 - t_3)$$
etc.

In general, if we have t_n and the approximation to the solution at this point, y_n, and we want to find the approximation at t_{n+1} all we need to do is use the following.

$$y_{n+1} = y_n + f(t_n,y_n)(t_{n+1} - t_n)$$

If we define $f_n = + \ f(t_n,y_n)$ we can simplify the formula to

$$y_{n+1} = y_n + \ (f_n)(t_n,y_n) \qquad (02)$$

Often, we will assume that the step sizes between the points t_0, t_1, t_2, … are of a uniform size of h. In other words, we will often assume that,

$$t_{n+1} - t_n = h$$

This doesn't have to be done and there are times when it's best that we not do this. However, if we do the formula for the next approximation becomes:

$$y_{n+1} = y_n + (h)(f_n) \qquad (03)$$

So, how do we use Euler's Method? It's fairly simple. We start with (01) and then decide if we want to use a uniform step size or not. Then starting with (t_0,y_0) we repeatedly evaluate (02) or (03) depending on whether we chose to use a uniform set size or not. We continue until we've gone the desired number of steps or reached the desired time. This will give us a sequence of numbers y_0, y_1, y_2, … y_n that will approximate the value of the actual solution at t_0, t_1, t_2, … t_n.

What do we do if we want a value of the solution at some other point than those used here? One possibility is to go back and redefine our set of points to a new set that will include the points we are after and redo Euler's Method using this new set of points. However this is cumbersome and could take a lot of time especially if we had to make changes to the set of points more than once.

Another possibility is to remember how we arrived at the approximations in the first place. Recall that we used the tangent line

$$y = y_0 + \ f(t_0,y_0)(t - t_0)$$

to get the value of y_1. We could use this tangent line as an approximation for the solution on the interval $[t_0,t_1]$. Likewise, we used the tangent line

$$y = y_1 + \ f(t_1,y_1)(t - t_1)$$

to get the value of y_2. We could use this tangent line as an approximation for the solution on the interval $[t_1,t_2]$. Continuing in this manner we would get a set of lines that, when strung together, should be an approximation to the solution as a whole.

In practice you would need to write a computer program to do these computations for you. In most cases the function f(t,y) would be too large and/or complicated to use by hand and in most
serious uses of Euler's Method you would want to use hundreds of steps which would make doing this by hand prohibitive. So, here is a bit of pseudo-code that you can use to write a program for Euler's Method that uses a uniform step size, h.

```
1. define f (t, y) .
2. input t0 and y0.
3. input step size, h and the number of steps, n.
4. for j from 1 to n do
a. m = f (t0, y0)
b. y1 = y0 + h*m
c. t1 = t0 + h
d. Print t1 and y1
```

e. t0 = t1
f. y0 = y1
5. end

So, let's take a look at a couple of examples. We'll use Euler's Method to approximate solutions to a couple of first order differential equations. The differential equations that we'll be using are linear first order differential equations that can be easily solved for an exact solution. Of course, in practice we wouldn't use Euler's Method on these kinds of differential equations, but by using easily solvable differential equations we will be able to check the accuracy of the method. Knowing the accuracy of any approximation method is a good thing. It is important to know if the method is liable to give a good approximation or not.

EXAMPLE 01
$y' + 2y = 2 - e^{-4t}$ $y(0) = 1$
Use Euler's Method with a step size of h = 0.1 to find approximate values of the solution at t = 0.1, 0.2, 0.3, 0.4, and 0.5. Compare them to the exact values of the solution as these points.
This is a fairly simple linear differential equation so we'll leave it to you to check that the solution is
$$y(t) = 1+\frac{1}{2}e^{-4t}-\frac{1}{2}e^{-2t} \qquad (4)$$
*Note: this equation, (4) gives the exact solution to the DE. Substituting the 'X' values for 't' will give the 'Exact' answers in the spreadsheet below.
First rewrite the equation into the form in 1.
$y' = 2 - e^{-4t} - 2y$
We see that $f(t,y) = 2 - e^{-4t} - 2y$. Also note that $t_0 = 0$ and $y_0 = 1$. We can now start doing some computations.
$f(0) = f(0,1) = 2e^{-4(0)} - 2(1) = -1$
$y_1 = y_0 + (h)(f_0) = 1 + (0.1)(-1) = 0.9$
So, the approximation of the solution at $t_1 = 0.1$ is $y_1 = 0.9$.
At the next step we have,
$f_1 = f(0.1,0.9) = 2 - e^{-4(0.1)} - (2)(0.9) = -0.470320046$
$y_2 = y_1 + (h)(f_1) = 0.9 + (0.1)(-0.470320046) = 0.852967995$
Therefore, the approximation at t_2 is $y_2 = 0.852967995$
$f_2 = -0.155264954$ $y_3 = 0.837441500$
$f_3 = 0.023922788$ $y_4 = 0.839833779$
$f_4 = 0.1184359245$ $y_5 = 0.851677371$
Since we already solved this problem in the beginning (4), we can find the exact values for each t value. This we will use for a comparison to determine the accuracy of this method.
$\text{Error} = \frac{\text{Exact} - \text{Approx}}{\text{Exact}} \times 100$

Time (t_n)	Approx	Exact	Error
$t_0 = 0.0$	$y_0 = 1.000000000$	$y(0.0) = 1.000000000$	0.00%
$t_1 = 0.1$	$y_1 = 0.900000000$	$y(0.1) = 0.925794646$	2.79%
$t_2 = 0.2$	$y_2 = 0.852967995$	$y(0.2) = 0.889504459$	4.11%
$t_3 = 0.3$	$y_3 = 0.837441500$	$y(0.3) = 0.876191288$	4.42%
$t_4 = 0.4$	$y_4 = 0.839833779$	$y(0.4) = 0.876283777$	4.16%
$t_5 = 0.5$	$y_5 = 0.851677371$	$y(0.5) = 0.883727921$	3.63%

(Dawkins, 104 #1).

The maximum error in the approximations from the last example was 4.42%, which isn't too bad, but also isn't all that great of an approximation. So, provided we aren't after very accurate approximations this didn't do too badly. This kind of error is generally unacceptable in almost all real applications however. So, how can we get better approximations?

Recall that we are getting the approximations by using a tangent line to approximate the value of the solution and that we are moving forward in time by steps of h. So, if we want a more accurate approximation, then we should take smaller h's.

EXAMPLE 02
Repeat the previous example only this time give the approximations at t = 1, t = 2, t = 3, t = 4, and t = 5. Use h = 0.1, h = 0.05, h = 0.01, h = 0.005, and h = 0.001 for the approximations.
Do this by: for t = 1, t_i = 0, y_i = 1, t_f = 1, h = 0.1; for t = 2, t_i = 0, y_i = 1, t_f = 1, h = 0.1.
However! I don't recommend doing this problem. It is very time consuming. Some of the solution is here.

		APPROXIMATIONS				
TIME	**EXACT**	**h=0.1**	**h=0.05**	**h=0.01**	**h=0.005**	**h=0.001**
t=1	0.9414902	0.9313244	0.9364698	0.9404994	0.9409957	0.9413914
t=2	0.9910099	0.9913681	0.9911126	0.9910193	0.9910139	0.9910106
t=3	0.9987637	0.9990501	0.9988982	0.9987890	0.9987763	0.9987662
t=4	0.9998323	0.9998976	0.9998657	0.9998390	0.9998357	0.9998330
t=5	0.9999773	0.9999890	0.9999837	0.9999786	0.9999780	0.9999774

		PERCENTAGE ERRORS			
TIME	**h=0.1**	**h=0.05**	**h=0.01**	**h=0.005**	**h=0.001**
t=1	1.08%	0.53%	0.105%	0.053%	0.0105%
t=2	0.036%	0.010%	0.00094%	0.00041%	0.0000703%
t=3	0.029%	0.013%	0.0025%	0.0013%	0.00025%
t=4	0.0065%	0.0033%	0.00067%	0.00034%	0.000067%
t=5	0.0012%	0.00064%	0.00013%	0.000068%	0.000014%

(Dawkins, 105 #2).

It should be clear that this method is not very practical. If h is not very small, then the approximation errors grow to inaccurate results. If h is very small, then the errors will be smaller, but the number of computations will increase. So this method will become tedious. In general, we will find for a fixed value of h, the error becomes larger and larger the distance we move from the initial point. In fact, for Euler, reducing the step size of h by a factor of 2 generally reduces the error sizes by a factor of 2. But, this increases the number of calculations by 3. (Ross, 446).

Please understand that much of the data illustrated here is using rounded off numbers to save typing strain.
*In practice, use all of the calculated values without rounding. This will improve the accuracy of all of these methods.

BIBLIOGRAPHY

Dawkins, Paul. "Paul's Online Math Notes." http: //tutorial.math.lamar.edu/. Downloaded: 18 Feb 2017. This PDF is "DE Complete."

Ross, Shepley L. Introduction To Ordinary Differential Equations. 4th ed. New York: John Wiley & Sons, 1989.