

# NUMERICAL SOLUTIONS FOR FIRST ORDER DIFFERENTIAL EQUATIONS

We have already had some experience setting up and solving discrete equations. Thus far, we have focused on various types of motion problems, and have not yet considered first order differential equations more generally.

In this classnote, we introduce a simple technique for solving many first order differential equations. This technique, known as the Euler method, can be used in many cases to find numerical solutions to first order differential equations. Those of you who took AP Calculus in high school may have already encountered the Euler method in those courses.

## *The Euler Method*

We begin our discussion of the Euler method by recalling the definition of the derivative of a function  $f(x)$  :

$$f'(x) = \frac{f(x+h) - f(x)}{h} \quad (1)$$

in the limit that  $h$  is a small number.

With simple rewriting, we recognize that we can express the value of the function at the new position,  $f(x+h)$  as :

$$f(x+h) = f(x) + hf'(x) \quad (2)$$

This equation tells us that the value of the function at the next position  $f(x+h)$ , equals the value of the function at the previous position ( $f(x)$ ) plus the change in value of the function between the two points,  $x$  and  $x+h$ . If  $h$  is small enough, meaning that  $x+h$  and  $x$  are close enough together, we can approximate this distance by a straight line whose slope is the derivative of the function. Therefore, the total change in the value of the function (or in high school geometry terms the total "rise") is just the slope of the function multiplied by the "run". In equation (2) above, the slope is the derivative,  $f'(x)$  and the run ( $\Delta x$  in pre - calculus language) is represented by  $h$ .

Let's see how we can write a simple Mathematica code to use this technique to solve simple differential equations. As we are learning, the best strategy to use at the start is to find an equation whose solution you can easily find from analytic methods, and then compare the accuracy of your numerical results to the analytical results.

We'll choose :

$$\frac{dy}{dx} = x^2 \text{ with the initial condition that } y(0) = 1$$

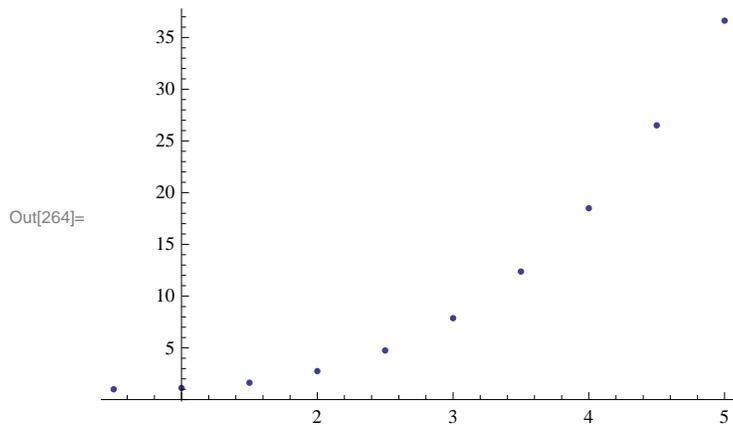
Let's arbitrarily choose a step - size of  $h = 0.5$ .

```
In[260]:= Clear[x, y, h]
          x[0] = 0; y[0] = 1; h = 0.5;
          x[n_] := x[n] = x[n - 1] + h
          y[n_] := y[n] = y[n - 1] + h x[n - 1]^2
```

Let's look at these four lines of code carefully. The first line simply clears all previous values for the variables we wish to use. The second line sets constants and establishes initial conditions. We will begin our numerical simulation at  $x = 0$ , so we obviously set  $x[0] = 0$ . We choose the initial condition that  $y[0] = 1$  and so we establish that condition.

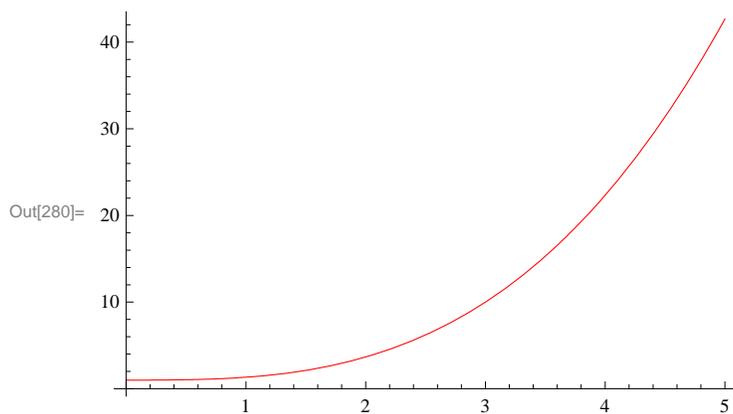
The third line of code calculates the value of  $x$  for the  $n$ th iteration; we need this value for the fourth line of code. Let's examine the fourth line in terms of Equation (2). The "new" value of  $y$  to be computed is represented by  $y[n]$ ; this is equated to the previous value of  $y$  (namely,  $y[n - 1]$ ) plus the product of  $h f'(x)$ . Remember that we are solving the equation  $f'(x) = x^2$ , so the term  $x[n - 1]^2$  takes the place of  $f'(x)$  in eq. (2). Now, let's plot the results of the numerical simulation for the interval  $[0,5]$ :

```
g1 = ListPlot[Table[{x[n], y[n]}, {n, 10}]]
```



Elementary techniques of solving differential equations allow us to easily determine that the analytical solution is  $y = (x^3/3)+1$ . Plotting this over the same interval:

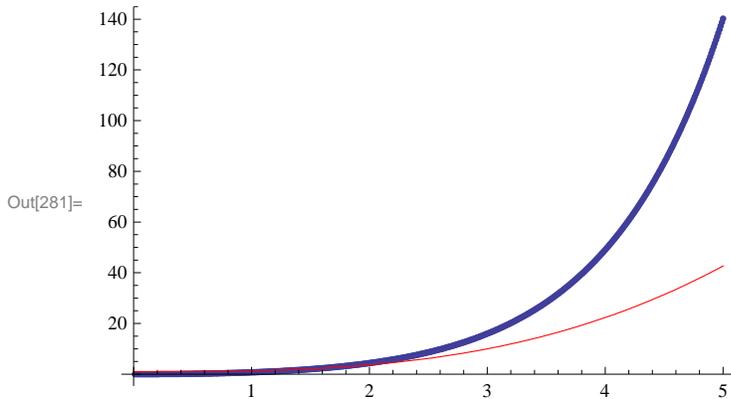
```
In[280]:= g2 = Plot[1 + (x^3 / 3), {x, 0, 5}, PlotStyle -> Red]
```



Where I have plotted the results of the analytic solution in red to ease in distinguishing the two curves.

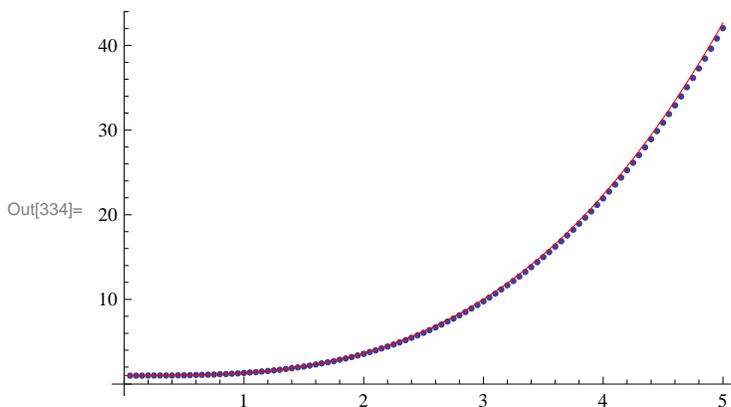
Notice that I named the graphs as if they were variables; this allows me to display them on the same set of axes for a more meaningful comparison :

```
In[281]:= Show[g1, g2]
```



And we see that the curves approximate each other pretty well for  $x < 2$ , but diverge for larger values of  $x$ . This is not too surprising given the large step - size we choose. We can do this again with a smaller step size; decreasing the step - size by a factor of 10 and increasing the number of evaluation points by a factor of 10, we have :

```
In[329]:= Clear[x, y, h]
x[0] = 0; y[0] = 1; h = 0.05;
x[n_] := x[n] = x[n - 1] + h
y[n_] := y[n] = y[n - 1] + h x[n - 1]^2
g3 = ListPlot[Table[{x[n], y[n]}, {n, 100}]];
Show[g3, g2]
```



Clearly a much better fit.

Let's try using this technique on the equation :

$$\frac{dy}{dx} = x y + x^2 \quad \text{with } y(0) = 1$$

Solving this using Mathematica's differential equation solve, DSolve :

```
In[670]:= Clear[y, x]
          DSolve[{y'[x] == y[x] x + x^2, y[0] == 1}, y[x], x]
```

```
Out[671]= {{y[x] -> 1/2 (2 e^{x^2/2} - 2 x + e^{x^2/2} \sqrt{2 \pi} Erf[\frac{x}{\sqrt{2}}])}}
```

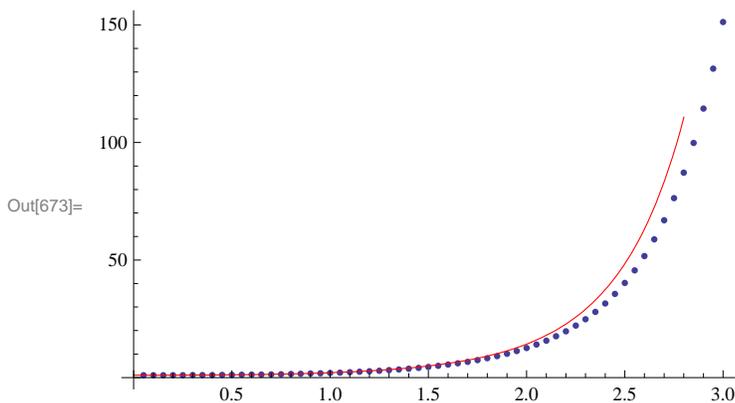
And the solution involves the ever popular error function Erf. We will plot this analytic solution against the numerical solution :

Using the Euler method :

```
In[665]:= Clear[x, y, h]
          x[0] = 0; y[0] = 1; h = 0.05;
          x[n_] := x[n] = x[n - 1] + h
          y[n_] := y[n] = y[n - 1] + h (x[n - 1] y[n - 1] + x[n - 1]^2)
          g5 = ListPlot[Table[{x[n], y[n]}, {n, 60}]];
```

```
In[672]:= g6 = Plot[1/2 (2 e^{x^2/2} - 2 x + e^{x^2/2} \sqrt{2 \pi} Erf[\frac{x}{\sqrt{2}}]), {x, 0, 3}, PlotStyle -> Red];
```

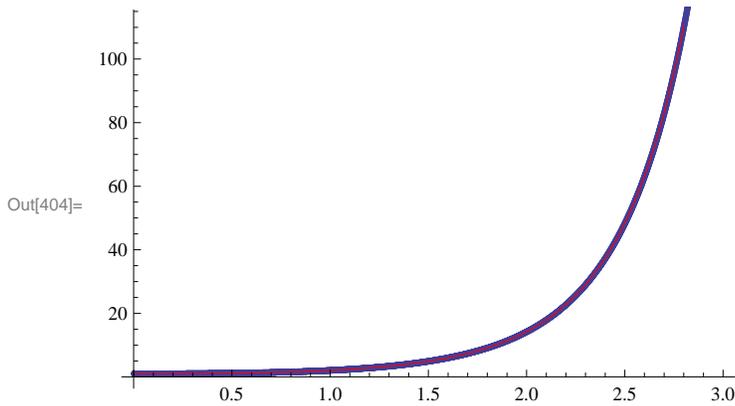
```
Show[g5, g6]
```



Not terrible, but you can see deviation between the two solutions; this should not be surprising since the derivative is a fairly complicated function and is most definitely not a linear function. Will decreasing the step - size help?

```
In[399]:= Clear[x, y, h]
          x[0] = 0; y[0] = 1; h = 0.001;
          x[n_] := x[n] = x[n - 1] + h
          y[n_] := y[n] = y[n - 1] + h (x[n - 1] y[n - 1] + x[n - 1]^2)
          g5 = ListPlot[Table[{x[n], y[n]}, {n, 3000}]];
```

```
In[404]:= Show[g5, g6]
```



Notice, though, that the function increases rapidly as  $x$  increases. After all, the rate of change,  $dy/dx$ , is equal to  $x y + x^2$ , so this is completely expected behavior.

### ***Accuracy of the Euler Method***

The principal advantage of the Euler method is that it is clear, simple to understand, and relatively easy to execute either by hand or via computer program. However, despite our successful comparisons above, the Euler method is limited by its linear nature, so that very small step sizes are needed to keep significant errors from propagating. For the simple differential equations we have used here, we can set  $h = 0.01$  or  $0.001$  and obtain fairly accurate results, however, for more complicated equations, or if we have to find solutions over a larger range of  $x$ , the inherent errors in the Euler method, coupled with the increasingly small step size requires significant computer time and power.

There are many other numerical methods to solve ODEs; in Math 264 you may have (or may in the future) study the Runge - Kutta technique, which is usually the next technique studied after the Euler method.