

9 *A glimpse of numerical methods

What to do if an analytical solution is not possible, and qualitative methods, that we studied in the last lecture, are not enough? In this case it is (almost) always possible to apply some *numerical methods* to solve the problem. By numerical methods it is usually meant a procedure that replaces the original problem with another one, which 1) in some sense produces a very close (approximate) solution, 2) easy to solve, at least in principle, 3) usually requires a significant amount of calculations, performed by a computer.

There exists a variety of free or commercial software of various levels of sophistication, which allow obtaining numerical solutions of ODE. For example, NDSU has a licence for using **Wolfram Mathematica**, which is an extremely powerful mathematical software.

Recall that the ODE $y' = y^2 + x$ has no solution that can be expressed using the basic functions. We can request **Mathematica** to find the solution and draw its graph using the following sequence of commands (I use here the initial condition $y(0) = 0.1$):

```
s = NDSolve[{y'[x] == y[x]^2 + x, y[0] == 0.1}, y, {x, 0, 1}];  
Plot[Evaluate[y[x] /. s], {x, 0, 30}, PlotRange -> All]
```

The result will be as follows.

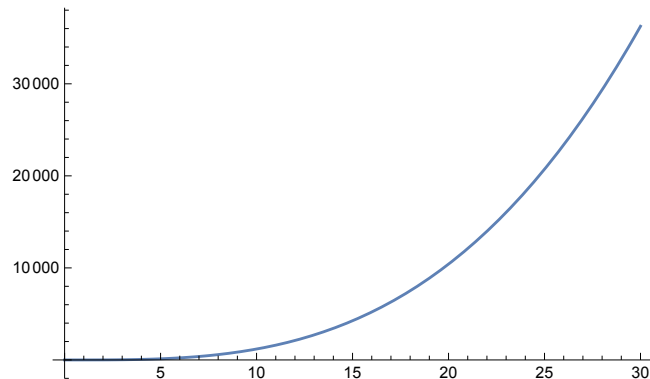


Figure 1: The graph of the numerical solution to $y' = y^2 + x$, $y(0) = 0.1$ produced by **Mathematica**.

But the key question remains: How does the software work and what exactly is being computed inside it? Here I will only touch on this topic, much more can be learnt in Math 488: Numerical Analysis. I also encourage you to download **Mathematica** and experiment with it. You can learn more by using the built-in help system. You can start with the command `?NDSolve`.

9.1 Euler's method

The earliest and the simplest method to numerically approximate the solution to the IVP

$$y' = f(x, y), \quad y(x_0) = y_0 \tag{1}$$

is the so-called Euler's method (after Leonhard Euler, 1707–1783, a great Swiss mathematician). This method is based on the observation that by having (1) we actually know point (x_0, y_0) through which

our solution passes, and the slope of the solution at this point, which is given by $y'(x_0) = f(x_0, y_0)$. Hence we can approximate our solution by a straight line on the interval $(x_0, x_0 + h)$:

$$x_1 = x_0 + h, \quad y_1 = y_0 + hf(x_0, y_0).$$

Now we have an approximation of the (unknown) solution $y(x)$ at the point $x_1 = x_0 + h$ with y_1 . Hence we can make the same step, assuming now that the initial condition (x_1, y_1) :

$$x_2 = x_1 + h = x_0 + 2h, \quad y_2 = y_1 + hf(x_1, y_1).$$

Or, generalizing, one has

$$y_{k+1} = y_k + hf(x_k, y_k), \quad x_k = x_0 + kh, \quad k = 0, 1, 2, \dots$$

In general, if one needs to approximate the solution to (1) at the point x , the number of steps n is chosen, such that now the step $h = \frac{x-x_0}{n}$, and calculations stop when k reaches the final value n . The biggest question of course is what kind of error we are making by using this approximation, and how many steps (or, equivalently, what is the step's length) we should take to guarantee the required accuracy.

Example 1. Consider the following IVP

$$y' = \frac{2(y^2 + 1)}{x^2 + 4}, \quad y(0) = 1.$$

This is actually separable equation that can be solved analytically, the solution is (fill in the missed steps)

$$y(x) = \frac{x + 2}{2 - x}.$$

Now let us see how Euler's method works for this example (see Fig. 2). I choose $n = 2$ ($h = 0.5$), $n = 4$ ($h = 0.25$), and $n = 10$ ($h = 0.1$) points. As expected, the more steps I take, the better approximation I obtain.

Here are the results for $n = 10$ and $n = 20$:

x_i	$y(x)$	$h = 0.1$	ϵ_1	$h = 0.05$	ϵ_2
0.0000	1.0000	1.0000	0.0000	1.0000	0.0000
0.1000	1.1100	1.1000	0.0053	1.1000	0.0027
0.2000	1.2200	1.2100	0.0120	1.2200	0.0063
0.3000	1.3500	1.3300	0.0207	1.3400	0.0108
0.4000	1.5000	1.4700	0.0321	1.4800	0.0169
0.5000	1.6700	1.6200	0.0471	1.6400	0.0249
0.6000	1.8600	1.7900	0.0670	1.8200	0.0358
0.7000	2.0800	1.9800	0.0940	2.0300	0.0505
0.8000	2.3300	2.2000	0.1310	2.2600	0.0709
0.9000	2.6400	2.4500	0.1810	2.5400	0.0995
1.0000	3.0000	2.7500	0.2530	2.8600	0.1400

In this table x_i are coordinates, $y(x)$ are the exact values of the function at x_i , $h = 0.1$ means these are the values of the approximation obtained by Euler's method using the step size $h = 0.1$, ϵ_1 is the

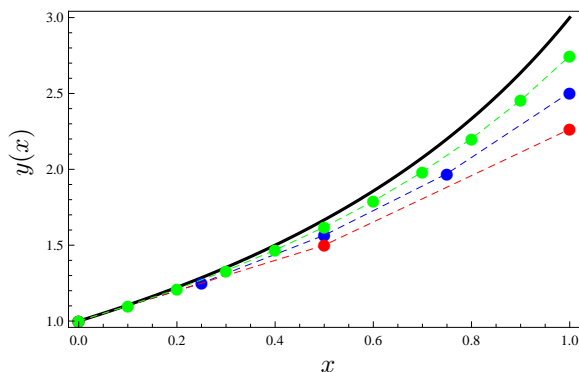


Figure 2: Numerical solutions of the IVP $y' = \frac{2(y^2+1)}{x^4+4}$, $y(0) = 1$. The bold black curve is the exact solution, the dashed lines are the numerical approximations obtained by Euler's method with the number of steps $n = 2$ (red), $n = 4$ (blue), $n = 10$ (green).

absolute error $|y(x_i) - y_i|$ and similar things for the last two columns that refer to the approximation with $h = 0.05$ (note that I do not show all the points here). You can note that the absolute error is about 2 times smaller for the approximation with two times shorter step h . This means that Euler's method gives us accuracy of the order h , the usual notation is $\mathcal{O}(h)$. And this is why this method is not exactly suitable for accurate calculations. Assume we would need the accuracy 10^{-10} at $x = 1$. To reach this accuracy we would need (assuming that the error is proportional to the step size)

$$h = \frac{0.05}{0.14 \cdot 10^{10}} \approx 3.6 \times 10^{-11},$$

which would require 2.8×10^{10} steps!

9.2 Improved Euler's method

How can we improve Euler's method? One thing is to note that we can adjust the slope of the approximation by using the slope at the new point. Again, $f(x_0, y_0)$ gives the slope at point x_0 , and $f(x_1, \tilde{y}_1)$ gives the approximation of the slope at the point x_1 , where $\tilde{y}_1 = y_0 + hf(x_0, y_0)$. The idea now is to take the average of these slopes such that the new approximate value of y_1 can be found as

$$y_1 = y_0 + \frac{h}{2} (f(x_0, y_0) + f(x_1, \tilde{y}_1)).$$

Or, in general,

$$\begin{aligned} x_{k+1} &= x_k + h, \\ \tilde{y}_{k+1} &= y_k + hf(x_k, y_k), \\ y_{k+1} &= y_k + \frac{h}{2} (f(x_k, y_k) + f(x_{k+1}, \tilde{y}_{k+1})), \quad k = 0, \dots, n. \end{aligned}$$

Here is the same information in the table form for our problem

x_i	$y(x)$	$h = 0.1$	ϵ_1	$h = 0.05$	ϵ_2
0.0000	1.0000	1.0000	0.0000	1.0000	0.0000
0.1000	1.1100	1.1100	0.0002	1.1100	0.0000
0.2000	1.2200	1.2200	0.0004	1.2200	0.0001
0.3000	1.3500	1.3500	0.0007	1.3500	0.0002
0.4000	1.5000	1.5000	0.0011	1.5000	0.0003
0.5000	1.6700	1.6600	0.0018	1.6700	0.0005
0.6000	1.8600	1.8500	0.0027	1.8600	0.0007
0.7000	2.0800	2.0700	0.0041	2.0800	0.0011
0.8000	2.3300	2.3300	0.0061	2.3300	0.0016
0.9000	2.6400	2.6300	0.0091	2.6300	0.0024
1.0000	3.0000	2.9900	0.0137	3.0000	0.0036

Here if the reduce the step size by half, the accuracy improves approximately 4 times. This can be rephrased that the improved Euler's method has the accuracy of the order h^2 , or, simply, $\mathcal{O}(h^2)$.

Going back to the same example and required accuracy 10^{-10} , one can find that now

$$h = \left(\frac{10^{-10}}{0.036} \right)^{1/2} \cdot 0.5 \approx 8.3 \times 10^{-6},$$

and the number of steps required $n = 1/h \approx 120123$.

Note that there are much better numerical methods, e.g., the fourth order Runge–Kutta method. For this method

$$\begin{aligned} k_1 &= f(x_j, y_j) \\ k_2 &= f(x_j + h/2, y_j + hk_1/2) \\ k_3 &= f(x_j + h/2, y_j + hk_2/2) \\ k_4 &= f(x_j + h, y_j + hk_3) \\ y_{j+1} &= y_j + h(k_1 + 2k_2 + 2k_3 + k_4)/6. \end{aligned}$$

This method has accuracy $\mathcal{O}(h^4)$.

All these (and many other) topics are covered in greater detail in Math 488: Numerical Analysis, which I strongly recommend for all the students to take.