

Math 2280: Computer Project 1
Due Feb. 11

In this project you will compute some numerical approximations of solutions to some simple ODEs using Euler's method, and then compare the numerical solution to the actual solution. I will provide some starter code (in MAPLE) at

<http://www.math.utah.edu/~ratzkin/m2280/projects/project1.mws>

Of course, you may use another computer tool (like Mathematica) if you prefer.

First recall how Euler's method works. We consider the initial value problem for a first order ODE of the following form:

$$y' = f(x, y), \quad y(0) = y_0. \quad (1)$$

Here y_0 is some real number, and we chose the starting point of $x_0 = 0$ for convenience.

Suppose we want to compute a numerical solution for $0 \leq x \leq 1$ (again, we're choosing an endpoint of 1 for convenience). The first thing we have to decide is our step size. In other words, how many data points do we want, or how finely can we approximate the real solution? Let's call the number of data points $n + 1$ (including $x_0 = 0$); then the stepsize will be

$$h = \frac{1}{n}.$$

This means the horizontal distance between two consecutive data points is $h = 1/n$. Next we need to generate the numerical solution at each data point. We will do this according to the rule

$$(x_0, y_0) = (0, y_0), \quad (x_{k+1}, y_{k+1}) = (x_k + h, y_k + hf(x_k, y_k)). \quad (2)$$

These data points (x_k, y_k) form the numerical solution.

What's the big idea? The big idea is that $f(x, y)$ gives the slope of the tangent line to the solution curve $(x, y(x))$, so near (x_k, y_k) , the solution is near it's tangent line

$$y = y_k + (x - x_k)f(x_k, y_k).$$

Euler's method replaces the real solution curve with the tangent line in the interval $x_k \leq x \leq x_{k+1}$. As the step size goes to zero, the numerical solution will converge to the real solution. Of course, as the step size goes to zero the number of data points also goes to infinity! Thus you have to sacrifice quick computing for a better numerical solution.

The Actual Project: We will consider the differential equation

$$y' = 2y, \quad (3)$$

with a varying initial condition.

1. Log on to the computer and familiarize yourself with MAPLE, or whatever computer tool you're using. You might want to play around for a little while before you start. You don't have to turn anything in for this part.
2. Verify that the solution to equation (3) with $y(0) = 1$ is

$$y(x) = e^{2x}.$$

3. Run Euler's method with a step size $h = 1/100$ and plot the data points you get.
4. Plot the data points from the numerical solution and the actual solution curve on the same graph and compare.
5. The data points for the numerical solution should all lie below the real solution curve in your previous plot. Can you explain why?
6. Do the same thing with a starting point $y(0) = -1$. This time the real solution is

$$y(x) = -e^{2x}$$

and the numerical solution lies *above* the real solution curve. Why does that happen?

7. Explain what happens when $y(0) = 0$.
8. Notice also that for the first two examples, the numerical solutions got worse as x increased. Why is that? (Hint: it's related to the fact that the numerical solution lies on one side of the actual solution.) This phenomenon of the numerical solution getting worse is called *creep*.
9. Can you think of an example where creep doesn't really occur?