# Project 2: Numerical Methods

**Due Date: 20 Oct 2015 (Tuesday)**
**50 Points**

As we've discussed in class, some differential equations cannot be solved analytically, and others are particularly tricky to solve (and thus may not be worth the time it takes to figure them out). Consequentially, we must turn to approximations of the solutions generated with techniques known as numerical methods. In general, a numerical method is some procedure for generating a sequence of points that approximate the solution.

Suppose we have the following initial value problem:

$$y' = f(x, y) \quad \text{where } y(x_0) = y_0$$

Let's call the analytic solution to this IVP $\phi(x)$ (which we may or may not be able to find). Now, we know the initial value $y_0 = \phi(x_0)$. Then a numerical method specifies some procedure/formula to obtain an approximate value slightly away from the initial value $y_1 \approx \phi(x + \Delta x)$. We can then iterate our numerical method to obtain an approximation a bit further away $y_2 \approx \phi(x + 2\Delta x)$. Repeating this over and over again allows us to obtain an approximate solution.

## Euler's Method

Perhaps the most natural and well-known numerical method is Euler's Method. Leonhard Euler was a Swiss mathematician from the 18$^{\text{th}}$ century who made contributions to many math topics including graph theory, number theory, and analysis. (Perhaps you remember our discussion of Euler's Formula in class? Same guy!)

Euler's Method takes advantage of our knowledge of the derivative from calculus. Recall the definition of the derivative:

$$g'(x) = \lim_{h \to 0} \frac{g(x+h) - g(x)}{h}$$

Then for very small values of $h > 0$, we can approximate the derivative and do a bit of algebra:

$$\begin{aligned} g'(x) &\approx \frac{g(x+h) - g(x)}{h} \\ hg'(x) &\approx g(x+h) - g(x) \\ hg'(x) + g(x) &\approx g(x+h) \end{aligned}$$

Thus if we know an expression for $g'(x)$, we can approximate the function slightly away from any value of $g(x)$ we happen to know. Another way to think about this is as a truncated Taylor series expansion of $g(x+h)$ at the point $a = x$:

$$\begin{aligned} g(x+h) &= g(x) + g'(x)\left(x + h - (x)\right) + \frac{g''(x)}{2!}\left(x + h - (x)\right)^2 + \cdots \\ &= g(x) + g'(x)h + \frac{g''(x)h^2}{2!} + \cdots \\ &\approx g(x) + hg'(x) \end{aligned}$$

Let's consider the initial value problem:

$$y' = f(x, y) \quad \text{where } y(x_0) = y_0$$

Then $f(x, y)$ is just an expression for the derivative of $y$. Since we know the value $y_0 = y(x_0)$, we can approximate the value a bit away by using the above logic:

$$y_1 = hf(x_0, y_0) + y_0 \approx y(x_0 + h)$$

So now we have an approximate value of the true solution. Continuing in this fashion, we have:

$$y_2 = hf(x_1, y_1) + y_1 \approx y(x_1 + h) = f(x_0 + 2h)$$

Continuing in this technique, we end up with a sequence of values $\{y_0, y_1, y_2, \cdots\}$ that approximate the analytic solution.

For example, consider the initial value problem:

$$y' = f(x, y) = \cos(x) \quad \text{where } y(0) = 0$$

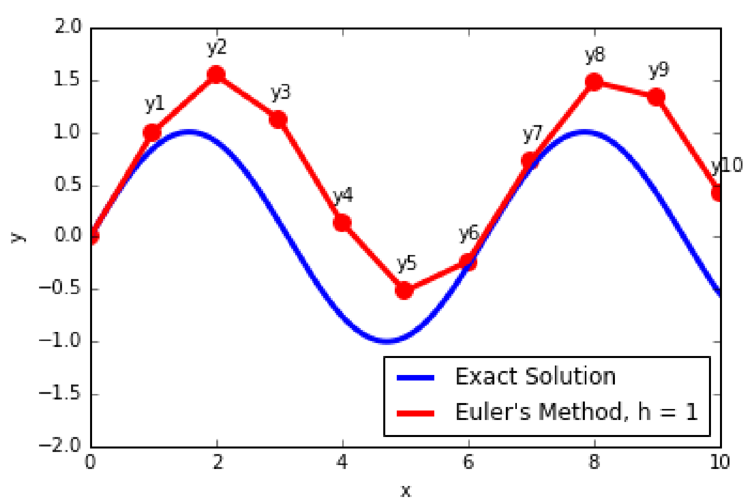
This is a separable differential equation; so analytically we know the true solution is given by:

$$\phi = \sin(x)$$

But using Euler's Method we can approximate this solution (with $h = 1$):

| $\mathbf{x}$ | $\mathbf{y_i}$ | $\mathbf{y' = f(x, y)}$ |
|---|---|---|
| 0 | $0$ | $\cos(0) = 1$ |
| 1 | $0 + 1(1) = 1$ | $\cos(1) \approx 0.540$ |
| 2 | $1 + 0.540(1) = 1.540$ | $\cos(2) \approx -0.416$ |
| 3 | $1.540 - 0.416(1) = 1.124$ | $\cos(3) \approx -0.990$ |
| 4 | $1.124 - 0.990(1) = 0.134$ | $\cos(4) \approx -0.654$ |
| 5 | $0.134 - 0.654(1) = -0.520$ | $\cos(5) \approx 0.284$ |
| 6 | $-0.520 + 0.284(1) = -0.236$ | $\cos(6) \approx 0.960$ |
| 7 | $-0.236 + 0.960(1) = 0.724$ | $\cos(7) \approx 0.754$ |
| 8 | $0.724 + 0.754(1) = 1.478$ | $\cos(8) \approx -0.146$ |
| 9 | $1.478 - 0.146(1) = 1.332$ | $\cos(9) \approx -0.911$ |
| 10 | $1.332 - 0.911(1) = 0.421$ | $-$ |

Which is best seen in the following picture:



In general, Euler's Method may be described with the following algorithm:

**Step 1.** Define `f(t,y)` (the derivative function).

**Step 2.** Input initial conditions: Initial time `t0` and initial function value `y0`

**Step 3.** Input parameters for Euler's Method: Step size `h` and number of steps `n`

**Step 4.** Run a loop to implement Euler's Method:

$$\mathtt{y_{i+1}} \approx \mathtt{y_i} + \mathtt{h} * \mathtt{f}(\mathtt{t_i}, \mathtt{y_i})$$

## Python's Numerical ODE Solver

Euler's Method is just one numerical technique that is used to solve differential equations. If you're interested in learning more about this sort of thing, you should take MTH 438: Numerical Analysis I. (Essentially, numerical analysis is the branch of applied mathematics that intersects computer science.)

As part of the SciPy package, Python has a built-in command called `odeint()` that will choose and implement the "optimal" numerical method for a given differential equation. `Intro to Python 2.ipynb` on Blackboard (also available as a `.html` file) walks you through how to use this command.

## Goals

By the end of this project, you should...

- understand what `for` and `while` loops are and how Python handles them.
- know what a numerical method is and how they may be used to approximate the solution of a differential equation.
- be able to implement Euler's Method.
- be able to use the numerical ODE solver `odeint()` from SciPy.

## Instructions

Work through the `Intro to Python 2.ipynb` file on Blackboard (also available as an `.html` file). Then write Python code that implements Euler's Method.

Consider the following initial value problems:

**1.**

$$\frac{dy}{dt} - y = t \quad \text{on the interval } [0, 8] \text{ where } y(0) = 4$$

with analytic solution:

$$\phi(t) = -t + 5e^t - 1$$

**2.**

$$\frac{dy}{dt} = y - 0.5e^{\frac{t}{2}} \sin(3t) + 3 * e^{\frac{t}{2}} \cos(3t) \quad \text{on the interval } [0, 6] \text{ where } y(0) = 0$$

with analytic solution:

$$\phi(t) = e^{\frac{t}{2}} \sin(3t)$$

**3.**

$$\frac{dy}{dt} - y\sin(t) - e^{\cos(t)} = 0 \quad \text{on the interval } [0, 30] \text{ where } y(0) = e$$

which has no "nice" analytic solution.

For each of the above IVPs:

**(a)** Use your Python code to implement Euler's Method with $h = 1, 0.1, 0.01$. Plot these three numerical approximations together with the given analytic solution.

**(b)** Use Python's `odeint()` to find an approximate solution. Plot this together with the given exact solution.

In addition, write a brief summary that addresses the following questions:

- What happens as you decrease the value of $h$ in Euler's Method? Why does this make sense? What are some possible complications of decreasing $h$ too far?
- How accurate does Python's `odeint()` seem to be? What might be some of the "dangers" in using `odeint()`?

At the beginning of class on the due date, you need to turn in a write up that must include the code you've used to complete the above tasks and the output of that code (i.e., the graphs).