Integrating Differential Equations

Phy 379

Dave Pawlowski

We are going to spend some time investigating numerical techniques used to integrate systems of ordinary differential equations (ODEs).

An ODE is a differential equation in which all dependent variables are functions of a single independent variable.

We would like to be able to solve an n^{th} order ODE, which means when the ODE is reduced to its simplest form, the highest order derivative it contains is n^{th}-order.

One such equation that we might find useful is:

$$\frac{d^2 x_j}{dt^2} = \frac{F_j(x_1, \ldots x_n, t)}{m_j}$$

Where x_j is the position of the j^{th} object, m_j is its mass, etc.

As it turns out, a set of n second-order ODEs can be rewritten as a set of 2n first-order ODEs.

$$\frac{dx_j}{dt} = v_j,$$

$$\frac{dv_j}{dt} = \frac{F_j(x_1, \ldots x_n, t)}{n_j}$$

For this reason, it's useful to study some techniques that apply 1^{st} order ODEs.

We are after numerical solutions to the equation

$$y' = f(x, y)$$

where the prime denotes taking the derivative with respect to x and the solution is subject to the initial-value boundary condition:

$$y(x_0) = y_0$$

Generalized → a solution can be applied to a system of N 1st order ODEs.

In order to solve this problem, like all numerical problems, we have to evaluate approximations to the function, y(x), at a series of discrete points, x_n, where:

$$x_n = x_0 + nh$$

and h, as you know by now, is the step-height.

We want to use the information given to us by the original ODE to find a solution.

Let y_n be the approximation to the function y(x) at the grid point x_n. We are given the gradient of y(x) at this point:

$$y' = f(x, y)$$

The most basic method to integrate an ODE is to employ Euler's method, which finds the solution of the function at a neighboring grid point, y_{n+1}, by assuming that the derivative is constant between the two points, y_n and y_{n+1}.

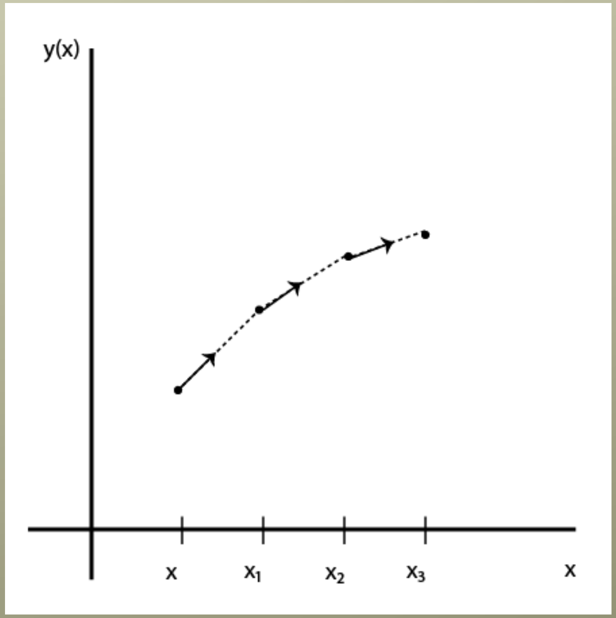
This is written as:

$$y_{n+1} = y_n + y'_n h$$

Since we are given information about y' everywhere, we can determine y_{n+1} if we know y_n, thus we require a boundary condition:

$$y(x_0) = y_0$$

Graphically:



Euler’s method is straightforward, however, there are issues: 1) error and 2) stability.

The methods that are typically used by scientists are called Runge-Kutta schemes (after a couple German mathematicians)

Since Euler’s method is asymmetric- it only depends on derivatives taken at the beginning of the interval- the errors are relatively high.

$$y' = f(x, y)$$

Runge-Kutta methods attempt to perform a more symmetric step

Take a Euler step to the midpoint of the interval

$$y_{n+1/2} = y_n + (\frac{h}{2}) f(x_n, y_n)$$

Then, then use x and y values there to calculate the solution at the real interval. This results in the 2nd order Runge-Kutta method.

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + h/2, y_n + k_1/2)$$

$$y_{n+1} = y_n + k_2 + O(h^3)$$

The result of this is the first order errors are cancelled out and we get a method that is 2nd order accurate. We gain accuracy by using the derivative evaluated at an extra point, and by being smart about it.

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + h/2, y_n + k_1/2)$$

$$y_{n+1} = y_n + k_2 + O(h^3)$$

Taking the derivative at more and more points allows us to develop higher order schemes. The 4th order Runge-Kutta method is one of the most popular methods of integrating ODEs currently in use.

4th Order Runge-Kutta integration

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf(x_n + h/2, y_n + k_1/2)$$

$$k_3 = hf(x_n + h/2, y_n + k_2/2)$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

$$y_{n+1} = y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5)$$

The 4th order RK method isn't necessarily limited by round off error as you increase n, but rather, but the computation effort. For every step, the ODE must be evaluated 4 times (in general, n times for a nth order RK method). As n gets large, this means many calculations are performed.

This is why higher order methods are not used very frequently. Higher order problems do not tend to give enough benefit in the reduction of error for the decrease in performance.