Numerical Solution Methods for Differential Equations

A first order ordinary differential equation is an equation of the form

$$\frac{\mathrm{d}\,x(t)}{\mathrm{d}t} = f(t, x(t))$$

along with initial conditions

$$x(0) = x_0$$

Solving the equation requires that we find an expression for $x(t)$ that satisfies both the equation and the initial condition. Not all differential equations can be solved exactly; however, there are a number of relatively simple and straightforward methods for finding numerical approximations to solutions. In these lecture notes we will look at three such methods.

The Mean Value Theorem

The mean value theorem says that a differentiable function $x(t)$ satisfies the equation

$$\frac{\Delta\, x}{\Delta\, t} = \frac{x(t_{n+1}) - x(t_n)}{t_{n+1} - t_n} = x'(t_c)$$

where t_c is some value of t in the interval $(t_n\,,\, t_{n+1})$. This theorem can be used as the basis for a method for solving differential equations, because it allows us to compute the change in the value of the function $x(t)$ as a function of the time step Δt and a derivative:

$$x(t_{n+1}) = x(t_n) + x'(t_c)\ (t_{n+1} - t_n) \tag{1}$$

There are two problems with this formula. The first is that it does not tell us what t_c is. The second is that even if we could find out what the correct value of t_c is, we would still not be able to compute

$$x'(t_c) = f(t_c\,,\, x(t_c))$$

because we would not know how to compute $x(t_c)$. All of the methods we are going to look at below get around this problem by using intelligent approximations for $x'(t_c)$.

Euler's Method

The first method is based on the approximation

$$x'(t_c) \approx x'(t_n) = f(t_n\,,\, x(t_n))$$

Dropping this estimate into (1) above gives

$$x(t_{n+1}) = x(t_n) + f(t_n\,,\, x(t_n))\ (t_{n+1} - t_n)$$

Here is a concrete example. Consider the differential equation

$$\frac{\mathrm{d}\, x(t)}{\mathrm{d}t} = t^2 + (x(t))^2$$

with initial condition

$$x(0) = 1$$

The initial condition tells us that at time $t_0 = 0$ the function $x(t_0)$ has value 1. The differential equation tells us that at time $t = 0$ the function has derivative

$$x'(0) = 0^2 + (1)^2 = 1$$

If we allow a short span of time $\Delta\ t$ to elapse between t_0 and t_1, the function will have a value approximately equal to

$$x(t_1) = x(t_0) + f(t_0\ , x(t_0))\ (t_1 - t_0) = \ 1 + (0^2 + 1^2)\Delta t = \ 1 + \Delta\ t$$

We can then repeat this calculation process again from the new point:

$$x(t_2) \approx x(t_1) + f(t_1\ , x(t_1))\ (t_2 - t_1) = (1 + \Delta t) + ((\Delta t)^2 + (1 + \Delta t)^2)\ \Delta t$$

More generally, what this method does is to compute a sequence of pairs $(t_n\ , x_n)$ where successive t_n and x_n values are related by equations

$$t_{n+1} = t_n + \Delta\ t$$

$$x_{n\,+\,1} = x_n + \Delta\ x \approx x_n + x'(t_n)\ \Delta t = x_n + f(t_n\ , x_n)\ \Delta\ t$$

The Modified Euler Method

The Modified Euler Method attempts to improve on the Euler method by using an average of a couple of derivatives as a better approximation for $x'(t_c)$. The Modified Euler method starts out by using the Euler method to compute an approximation for x_{n+1} called p. Plugging t_{n+1} and the approximate x_{n+1} into the differential equation provides an approximation for the derivative $x'(t_{n+1})$. The Modified Euler method then takes the average of this derivative and $x'(t_n)$, uses that as an approximation for $x'(t_c)$, and then computes a better estimate for x_{n+1}.

Here are the relevant formulas.

$$t_{n+1} = t_n + \Delta\ t$$

$$p = x_n + x'(t_n)\ \Delta t = x_n + f(t_n\ , x_n)\ \Delta\ t$$

$$x_{n\,+\,1} = x_n + \frac{f(t_n\ , x_n) + f(t_{n+1}\ , p)}{2}\ \Delta\ t$$

The Runge-Kutta method

Although the Euler and the Modified Euler methods are both very easy to understand, neither one of these methods does a very good job at approximating the exact solution to a problem. To get better results, we typically have to apply more sophisticated methods. One such method which typically produces pretty good results is the Runge-Kutta method. Like both the Euler and the Modified Euler methods, this method uses previously computed points x_n in combination with information from the $f(t,x(t))$ function to compute subsequent points x_{n+1}. To do this, the Runge-Kutta uses a more sophisticated set of formulas, as shown here:

$$k_1 = f(t_n\ ,\ x_n)$$

$$k_2 = f(t_n + (\Delta\ t)/2, x_n + k_1\ (\Delta\ t)/2\ )$$

$$k_3 = f(t_n + (\Delta\ t)/2\ ,\ x_n + k_2\ (\Delta\ t)/2\ )$$

$$k_4 = f(t_n + \Delta\ t\ ,\ x_n + k_3\ (\Delta\ t))$$

$$x_{n+1} = x_n + \frac{\Delta\ t}{6}(k_1 + 2\ k_2 + 2\ k_3 + k_4)$$

Deriving the Runge-Kutta method

The Runge-Kutta formulas above give us a method for approximating $x_{n+1} = x(t_{n+1})$ from $x_n = x(t_n)$ and the function $f(t,x)$. Another way to solve for x_{n+1} is to compute this integral

$$\int_{t_n}^{t_{n+1}} x'(t)\, dt = x(t_{n+1}) - x(t_n) = x_{n+1} - x_n$$

We can imagine beginning to compute the integral by noting that $x'(t) = f(t,x(t))$

$$\int_{t_n}^{t_{n+1}} x'(t)\, dt = \int_{t_n}^{t_{n+1}} f(t,x(t))\, dt$$

Unfortunately, we can not do the integral on the right hand side exactly, because we don't know what $x(t)$ is. That is, after all, the unknown we are trying to solve for. Even though we can't compute the integral on the right exactly, we can estimate it.

One technique that is commonly used to estimate integrals is Simpson's rule:

$$\int_a^b g(t)\, dt \approx \frac{b - a}{6}\left(g(a) + 4\ g\left(\frac{a+b}{2}\right) + g(b)\right)$$

For example, applying Simpson's rule to the integral above with $a = t_n$ and $b = t_{n+1}$ produces the estimate

$$\int_{t_n}^{t_{n+1}} f(t,x(t))\, dt \approx \frac{\Delta\ t}{6}\left(f(t_n,x(t_n)) + 4\ f\left(\frac{t_n+t_{n+1}}{2},x\left(\frac{t_n+t_{n+1}}{2}\right)\right) + f(t_{n+1},x(t_{n+1}))\right)$$

The Runge-Kutta method takes this estimate as a starting point. The thing we need to do to make this estimate work is to find a way to estimate the unknown terms $x((t_n + t_{n+1})/2)$ and $x(t_{n+1})$.

The first step is to rewrite the estimate as

$$\frac{\Delta t}{6}\left(f(t_n, x(t_n)) + 2 f\left(\frac{t_n + t_{n+1}}{2}, x\left(\frac{t_n + t_{n+1}}{2}\right)\right) + 2 f\left(\frac{t_n + t_{n+1}}{2}, x\left(\frac{t_n + t_{n+1}}{2}\right)\right) + f(t_{n+1}, x(t_{n+1}))\right)$$

We write the middle term twice because we are going to develop two different estimates for $x((t_n + t_{n+1})/2)$. The thinking is that the mistakes we make in developing those two interior estimates may partly cancel each other out.

Here is how we will develop our estimates.

1. $x(t_n)$ is just x_n.

2. We estimate the first $x((t_n + t_{n+1})/2)$ by driving the original Euler slope $k_1 = f(t_n, x_n)$ half-way across the interval:

$$k_1 = f(t_n, x(t_n))$$

$$x\left(\frac{t_n + t_{n+1}}{2}\right) \approx x_n + k_1 (\Delta t)/2$$

3. Next we compute a second slope at that midpoint we just estimated. We then rewind to the start and drive that slope half-way across the interval again.

$$k_2 = f(t_n + (\Delta t)/2 , x_n + k_1(\Delta t)/2)$$

$$x\left(\frac{t_n + t_{n+1}}{2}\right) \approx x_n + k_2 (\Delta t)/2$$

4. We use the second estimated midpoint to compute another slope and then drive that slope all the way across the interval.

$$k_3 = f(t_n + (\Delta t)/2 , x_n + k_2 (\Delta t)/2)$$

$$x(t_{n+1}) = x_n + k_3(\Delta t)$$

$$k_4 = f(t_n + (\Delta t), x_n + k_3 (\Delta t))$$

Substituting all of these estimates into the Simpson's rule formula above gives

$$x_{n+1} - x_n = \int_{t_n}^{t_{n+1}} f(t, x(t))\, dt \approx$$
$$\frac{\Delta t}{6}\left(f(t_n, x(t_n)) + 2 f\left(\frac{t_n + t_{n+1}}{2}, x\left(\frac{t_n + t_{n+1}}{2}\right)\right) + 2 f\left(\frac{t_n + t_{n+1}}{2}, x\left(\frac{t_n + t_{n+1}}{2}\right)\right) + f(t_{n+1}, x(t_{n+1}))\right)$$

or

$$x_{n+1} = x_n + \frac{\Delta\ t}{6}(k_1 + 2\ k_2 + 2\ k_3 + k_4)$$

Solving Systems of Equations

The methods developed above to solve ODEs generalize in a natural and straightforward way to systems of ODEs.

Here is an example. Consider the formulas for the Modified Euler method:

$$t_{n+1} = t_n + \Delta\ t$$

$$p = x_n + x'(t_n)\ \Delta t = x_n + f(t_n\ ,\ x_n)\ \Delta\ t$$

$$x_{n\,+\,1} = x_n + \frac{f(t_n\ ,\ x_n) + f(t_{n+1}\ ,\ p)}{2}\ \Delta\ t$$

Suppose that instead of solving an ODE

$$x'(t) = f(t,x(t))$$

$$x(t_0) = x_0$$

we have to solve a system of equations

$$\alpha'(t) = M^{1}\ K\,\alpha(t) + M^{1}\ \mathbf{f}(t)$$

$$\alpha(t_0) = \alpha_0$$

Writing out the formulas for the Modified Euler method with the scaler $x(t)$ replaced with the vector $\alpha(t)$ and $f(t,x(t))$ replaced by $M^{1}\ K\ \alpha(t) + M^{1}\ \mathbf{f}(t)$ gives

$$t_{n+1} = t_n + \Delta\ t$$

$$\mathbf{p} = \alpha_n + \alpha'(t_n)\ \Delta t = \alpha_n + f(t_n\ ,\ \alpha_n)\ \Delta\ t$$

$$= \alpha_n + M^{1}\ K\ \alpha_n + M^{1}\ \mathbf{f}(t_n)$$

$$\alpha_{n\,+\,1} = \alpha_n + \frac{f(t_n\ ,\ \alpha_n) + f(t_{n+1}\ ,\ p)}{2}\ \Delta\ t$$

$$= \alpha_n + (M^{1}\ K\ \alpha_n + M^{1}\ \mathbf{f}(t_n))\ \frac{\Delta\ t}{2} + (M^{1}\ K\ \mathbf{p} + M^{1}\ \mathbf{f}(t_{n+1}))\ \frac{\Delta\ t}{2}$$

The latter formula is the generalization of the Modified Euler method to the system of equations.