February 14, 2005

Lecture 8. Numerical methods for differential equations.

A numerical method for a differential equation is an algorithm that has the equation and the initial data as input, and has numbers as output. For a first-order ordinary differential equation the simplest algorithm is known as Euler's method. There's a geometric and an algebraic way to describe the method, and we will present both.

Geometric description. In simple words, the differential equation determines a slope field on the tf plane. Choose a *step size* h (a positive number). Starting at the initial value, follow the given slope in a straight line, until you have walked distance h in the t-axis. You are now at a new point. This new point is part of your output. Now repeat this procedure from the new point, using the same h. The path you will describe is a polygonal path, with vertices located at the places where you changed direction. This polygonal path is an approximation of the "real" solution to the initial-value problem. The smaller the value of h, the better this approximation will be. The sequence of outputs we obtained are the *discretization* of the solution. They form a recursive sequence, because the next point depends on the previous one, etc, all the way down to the seed (initial value). Now let's put these words into equations.

We are given the problem $f'(t) = G(t, f(t))$ with initial data $f(t_0) = f_0$. Here t_0 is a fixed initial time (often $t_0 = 0$), and f_0 is a given number. We choose a number $h > 0$, called the *step size* of the method. Once we have h we produce the discretization of the time interval, by setting $t_n = t_0 + nh$, so that $t_1 = t_0 + h$, $t_2 = t_1 + h$, etc. The true solution f to the initial-value problem will have value $f(t_n)$ at time t_n. We will produce an approximation to that true value, and will call it f_n. So, $f_n \approx f(t_n)$. The f_n will be a recursive sequence, and the seed is the given value f_0. Next we show how to obtain f_{n+1} from f_n.

Suppose we just obtained the approximation f_n corresponding to time t_n. This corresponds to the point (t_n, f_n) on the plane, and at that point the differential equation gives the slope $m = G(t_n, f_n)$. The straight line with slope m, going through the point (t_n, f_n) has equation

$$y = mt + f_n - G(t_n, f_n)t_n.$$

Setting $t = t_{n+1} = t_n + h$ and $y = f_{n+1}$ in this equation, we get

$$f_{n+1} = f_n + h\, G(t_n, f_n).$$

This is the recursive sequence for our approximation.

Example 1. Let $f' = f$, $f(0) = 1$ be our problem, and we want to solve it numerically, using the above method. The first thing to do is to choose a step size; the smaller the step, the better the approximation. Let's take $h = .01$. For this equation we have $G(t, f) = f$, and so $G(t_n, f_n) = f_n$. This gives us the recursive equation $f_{n+1} = f_n + h\, f_n = f_n(1 + h)$. The successive values will be $f_0 = 1$, $f_1 = 1.01$, $f_2 = (1.01)^2$, etc, $f_n = (1.01)^n$.

Example 2. $f' = (1-f)\,f = f - f^2$, $f(0) = 1/2$. The recursive equation is

$$f_{n+1} = f_n + h\,G(t_n, f_n) = f_n + h\,(1-f_n)\,f_n = (1+h)\,f_n - h\,f_n^2.$$

For $h = .01$ we get the values $f_1 = (1.01)/2 - .01/4 = .5025$, $f_2 = .5049999375$, etc.

Algebraic description. The derivative is a limit:

$$f'(t) = \lim_{h\to 0} \frac{f(t+h) - f(t)}{h}.$$

Let's approximate the limit by the quotient, for a very small value of h:

$$G(t, f(t)) = f'(t) \approx \frac{f(t+h) - f(t)}{h}.$$

Rewrite this as

$$f(t+h) \approx f(t) + h\,G(t, f(t)).$$

The equation above is telling us that in order to obtain information at time $t+h$ (left hand side), all we need is to know information at time t (right hand side). Since we do know information at time $t = t_0$ (initial data), we get information at time $t_1 = t_0 + h$, and using this information we obtain new information at time $t_2 = t_1 + h$, and so on, recursively. The formula becomes

$$f_{n+1} = f_n + h\,G(t_n, f_n).$$

This is the same recursive formula we had before.

Further ideas. Here is an *a posteriori* interpretation of Euler's method. What the above formula gives us is

$$f_{n+1} = f_n + h\,(\text{some slope}).$$

The slope we used in the above formula was $G(t_n, f_n)$, but there are other options for that slope. For example, a better slope is given by the average between the slopes $G(t_n, f_n)$ and $G(t_{n+1}, f_{n+1})$ (with f_{n+1} as given above, by Euler's method), we get a new formula

$$f_{n+1} = f_n + \frac{h}{2}\left[G(t_n, f_n) + G(t_{n+1}, f_n + h\,G(t_n, f_n))\right].$$

This is called Heun's method (or improved Euler's method).

Example. Let $f' = t^2 + f^2$, $f(0) = 1$. Then $G(t, f) = t^2 + f^2$, and Heun's method gives

$$\begin{aligned} f_{n+1} &= f_n + \frac{h}{2}\left[t_n^2 + f_n^2 + G(t_{n+1}, f_n + h(t_n^2 + f_n^2))\right] \\ &= f_n + \frac{h}{2}\left[t_n^2 + f_n^2 + t_{n+1}^2 + (f_n + h(t_n^2 + f_n^2))^2\right] \end{aligned}$$

Even further ideas. The idea that a derivative may be approximated by the difference quotient is very powerful, and as an idea it can also be iterated. Here's what I mean.

Consider the (equivalent) limit for derivatives:

$$f'(t) = \lim_{h\to 0} \frac{f(t+h/2) - f(t-h/2)}{h}.$$

We can now approximate *the second derivative* as follows:

$$f''(t) \approx \frac{f'(t+h/2) - f'(t-h/2)}{h} \approx \frac{\frac{f(t+h)-f(t)}{h} - \frac{f(t)-f(t-h)}{h}}{h}.$$

In other words,

$$f''(t) \approx \frac{f(t+h) - 2f(t) + f(t-h)}{h^2}.$$

This is called the *second difference* quotient for the second derivative. It produces recursive formulas depending on the previous two values of the recursive sequence it determines. We use the second difference quotient when dealing with second order equations. So, the equation

$$f''(t) + 2f'(t) - 3f(t) = 1$$

gives way to (for instance)

$$\frac{f(t+h) - 2f(t) + f(t-h)}{h^2} + 2\frac{f(t) - f(t-h)}{h} - 3f(t) = 1.$$

Rearranging terms we obtain

$$f_{n+1} = (2 + 2h + 3h^2)f_n - (1 + 2h)f_{n-1} + h^2.$$

This recursive sequence needs two seeds, and these seeds are given by the initial data of the differential equation.

Turning the tables. Given a recursive formula, if we know it came from a differential equation, can we figure out the equation? The answer is *yes*, and the way to do it is to use the Taylor polynomial. The Taylor polynomial says that

$$f(t+h) \approx f(t) + h\, f'(t) + \frac{h^2}{2}\, f''(t) + \dots$$

In a recursive formula f_{n+1} plays the role of $f(t+h)$, and f_n plays the role of $f(t)$, so we just replace them. Here's an example.

Given the formula $f_{n+1} = (1+h)f_n + ht_n$, let's recover the differential equation. Substituting f_{n+1}, f_n, and t_n, we get

$$f + hf' + \frac{h^2}{2}f'' = (1+h)f + ht.$$

Simplifying terms we get

$$f' + \frac{h}{2}f'' = t + f,$$

and letting h go to zero we obtain $f'(t) = t + f(t)$. If you use Euler's method in this equation you will recover the recursive formula.

Here's another example:

$$f_{n+1} = (1 + h + \frac{h^2}{2})f_n + (h + \frac{h^2}{2})t_n + \frac{h^2}{2}.$$

Substitute once more, cancel terms, and set $h = 0$ to get $f' = t + f$. If you now use Heun's method in this equation you will obtain the recursive sequence for this example.

A last word. We've seen two numerical methods – two of the simplest, not two of the best. You should be aware of three things. First, that there are better numerical methods available in practice (even though the basic idea for these methods is the same as in the methods we've seen). Second, that no discussion on approximation is complete unless we include an analysis of the error in the approximation. We didn't do it here. This analysis is what allows us to understand why Heun's method is better than Euler's, and why other methods are even better. Third (and this one is really subtle to understand), just because we have a numerical method, it does not mean that the method will give us the solution we are looking for (even approximately). We will understand this better when we look at numerical methods for partial differential equations. Finally, for partial differential equations we will need the multivariable version of Taylor's polynomial, namely

$$\begin{aligned} f(x+u, y+v) \quad &\approx \quad f(x,y) + u\frac{\partial f(x,y)}{\partial x} + v\frac{\partial f(x,y)}{\partial y} + \\ &\qquad \frac{1}{2}\left(u^2\frac{\partial^2 f(x,y)}{\partial x^2} + 2uv\frac{\partial^2 f(x,y)}{\partial x \partial y} + v^2\frac{\partial^2 f(x,y)}{\partial y^2}\right) + \dots \end{aligned}$$

Problems.

1. In this problem we will use Euler's method to find an approximation for e. Consider the differential equation $f' = f$ with initial data $f(0) = 1$. We know the solution is $f(t) = e^t$, therefore $f(1) = e$. Using as step size $h = .1$, and $f_0 = 1$, use Euler's method to obtain f_{10}. Your answer is an approximation to $f(t_{10}) = f(1)$.

2. Same as above, with $h = .05$, find f_{20}.

3. Same as problem 1, with Heun's method, $h = .1$, find f_{10}.

4. Same as problem 1, with Heun's method, $h = .05$, find f_{20}.

5. Knowing that the recursive formula $f_{n+1} = (1 - 2h)f_n$ was obtained from a differential equation, try to determine the original equation.