Numerical Solutions to PDE's II

Mathematical Modelling Week 6
Kurt Bryan

Approximating Derivatives; Other Methods

Some simple modifications to our finite difference approximations to the derivatives in the PDE can produce better (more accurate) results, although such modifications may change the stability requirements on the stepsizes.

As an example, consider again the advection equation $u_t + cu_x = 0$ for $x > 0$, $t > 0$, with initial condition $u(x, 0) = f(x)$, boundary condition $u(0, t) = g(t)$. As before we use stepsizes h_x and h_t, set $u_{ij} = u(ih_x, jh_t)$, and approximate u_t with the difference quotient $\frac{u_{i,j+1} - u_{ij}}{h_t}$.

But we can use a better approximation for u_x, namely

$$u_x(x,t) \approx \frac{u(x+h_x,t) - u(x-h_x,t)}{2h_x}. \tag{1}$$

This is called a *centered difference* approximation, as opposed to the one-sided approximation we used before. It's more accurate for small h_x, for you can use Taylor's Theorem to see that

$$\begin{aligned} u(x-h_x,t) &\approx u(x,t) - h_x u_x(x,t) + \frac{1}{2}h_x^2 u_{xx}(x,t) + O(h_x^3) \\ u(x+h_x,t) &\approx u(x,t) + h_x u_x(x,t) + \frac{1}{2}h_x^2 u_{xx}(x,t) + O(h_x^3). \end{aligned}$$

Inserting the above expressions into the centered difference formula (1) shows that

$$\frac{u(x+h_x,t) - u(x-h_x,t)}{2h_x} = u_x(x,t) + O(h_x^2).$$

Thus the centered difference formula is said to be *second order* accurate in space, since the error term is order h_x^2.

Contrast this to the previous one-sided difference, in which you can easily show that

$$\frac{u(x,t) - u(x-h_x,t)}{h_x} = u_x(x,t) + O(h_x)$$

which is first order accuracy in x.

Exercise

1. Show the one-sided difference is first order accurate in x.

Replacing the u_x term in the PDE with the centered difference approximation and the u_t term with the one-sided approximation yields the scheme

$$u_{i,j+1} = u_{ij} + \frac{ch_t}{2h_x}(u_{i+1,j} - u_{i-1,j}). \tag{2}$$

Because the centered difference is a better approximation to the second derivative (in x), we might expect this to provide better results.

There's only one problem: This method is UNSTABLE for any choice of h_t and h_x! I won't prove that here, but you might be amused to write it out in a matrix form and see that the relevant matrix always has eigenvalues larger than 1, no matter how small h_t is.

The Lax Method

One way to fix up equation (2) to preserve second order spatial accuracy and get stability is to replace the $u_{i,j}$ term in the TIME derivative approximation with the average $\frac{1}{2}(u_{i+1,j} - u_{i-1,j})$, so we approximate

$$u_t(x_i, t_j) \approx \frac{u_{i,j+1} - \frac{1}{2}u_{i+1,j} - \frac{1}{2}u_{i-1,j}}{h_t}.$$

Inserting this into the PDE $u_t + cu_x = 0$ yields

$$u_{i,j+1} = \left(\frac{1}{2} - \frac{ch_t}{2h_x}\right)u_{i+1,j} + \left(\frac{1}{2} + \frac{ch_t}{2h_x}\right)u_{i-1,j}. \tag{3}$$

This scheme turns out to be second order accurate in space (first order in time) and is STABLE for $ch_t/h_x \leq 1$, just like the first method we looked at.

There are many other methods for advection-like equations (the leapfrog method, the Lax-Wendroff method, etc.) but the stable ones I've shown you here are sufficient for our modelling purposes. For more information, check out Richtmeyer and Morton, *Difference Methods for Initial Value Problems* (Wiley, 1967) or W.F. Ames, *Numerical Methods for Partial Differential*

Equations, Academic Press, 1977. The *Numerical Recipes* Books (Cambridge Univ. Press) also has a quick treatment of PDE methods.

Methods for Diffusion

Consider the heat equation $u_t - ku_{xx} = 0$ where k is some constant. With notation $u(x_i, t_j) = u_{ij}$ as above, let's approximate the derivatives as

$$u_t(x_i, t_j) \approx \frac{u_{i,j+1} - u_{ij}}{h_t}, \quad u_{xx}(x_i, t_j) \approx \frac{u_{i+1,j} - 2u_{ij} + u_{i-1,j}}{h_x^2}.$$

Inserting these into the heat equation and solving for $u_{i,j+1}$ gives the FTCS (forward time, centered difference) scheme

$$u_{i,j+1} = \frac{kh_t}{h_x^2} u_{i-1,j} + \left(1 - \frac{2kh_t}{h_x^2}\right) u_{ij} + \frac{kh_t}{h_x^2} u_{i+1,j}, \tag{4}$$

and explicit time-marching scheme.

It turns out that the method is stable provided $\frac{2kh_t}{h_x^2} \leq 1$, not entirely surprising given the pattern of earlier methods (but not obviously true either!) One drawback to this method is that if we decrease h_x to get better spatial resolution of the solution, we must decrease h_t by a factor of FOUR to preserve stability. Thus if h_x is very small we end up taking a huge number of tiny time steps, which may be wasteful. Still, this method will suffice for our purposes.

Again, see the references above for other methods for diffusion type problems. In particular, there are so-called *implicit* methods in which we must solve a system of linear equations to find the solution at iteration $j + 1$ (we solve for all $u_{i,j+1}$ simultaneously). These implicit methods are often stable for large time steps, and so this may offset the extra work of solving the linear system at each iteration.

A Note on Boundary and Initial Conditions

We implement boundary conditions like $u(0, t) = g(t)$ by simply $u_{0,j} = g(t_j)$, as we've been doing. However, there are other types of boundary conditions, e.g., Neumann conditions. These too are easy to implement. For example, if we require $u_x(0, t) = g(t)$ we could approximate $u_x(0, t) \approx$

$(u_(h_x, t) - u(0,t))/h_x$. At $t = t_j$ this becomes $u_x(0, t_j) \approx (u_{1,j} - u_{0,j})/h_x$. The Neumann boundary condition at $t = t_{j+1}$ would then be implement as

$$\frac{u_{1,j+1} - u_{0,j+1}}{h_x} = g(t_{j+1})$$

or

$$u_{0,j+1} = u_{1,j+1} - h_x g(t_{j+1}). \tag{5}$$

At iteration $j+1$ we'd use the PDE method to compute $u_{1,j+1}$ from quantities at time t_j, then compute $u_{0,j+1}$ using equation (5).

Problems

1. Devise and implement a numerical method for solving the wave equation $u_{tt} - u_{xx} = 0$ (wave speed is 1 here) for $0 < x < 1$, $t > 0$, with initial conditions $u(x,0) = f(x)$, $u_t(x,0) = g(x)$, and boundary conditions $u(0,t) = h_0(t)$, $u(1,t) = h_1(t)$. Hint: Estimate $u_{i,j+1}$ using the value of u at iteration j AND iteration $j-1$. Also, computing $u_{i,0}$ from f is easy; compute $u_{i,1}$ using the u_t initial condition.

   Use your scheme to compute the solution to the wave equation with $f(x) = \sin(\pi x)$, $g(x) = 0$, $h_0(t) = h_1(t) = 0$. This models the motion of a string of length one deformed to initial position $\sin(\pi x)$ with zero initial velocity and ends tied to height zero at both ends at all times. Use $h_x = 0.05$. Does the result seem reasonable? What time step is needed for stability?