Finite Difference PDE Approximations

Chris Rackauckas

February 25, 2019

Now let's take a look at using finite differences to solve some equations!

1 Poisson Equation

Let's take a look at the Poisson equation

$$\Delta u = f$$

For finite differencing, recall that we get some matrix A with a stencil $[1, -2, 1]$ and boundary condition handling. By using $U = [u(i\Delta x)]$ as an array and $F = [f(i\Delta x)]$, we plug in the array approximation and get

$$AU = F$$

which is a linear equation to solve for A. Then by applying Gaussian elimination, LU-factorization, etc. we get the solution for U in terms of F.

1.1 Thomas Algorithm

Let's take a little bit of a dive into how this would be concretely solved. In our case, A isn't just any old matrix, it's a tridiagonal matrix (with maybe a few points on the corners if the BCs are periodic). Wilson's algorithm is an efficient method ($\mathcal{O}(n)$) for $Au = f$ when A is tridiagonal. Let's see how this works.

Look at the equation as

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i$$

Now re-write this in terms of x_{i-1}. At the start you have

$$b_1 x_1 + c_1 x_2 = d_1$$

so

$$x_1 = \frac{d_1 - c_1 x_2}{b_1}$$

Now use this equation to eliminate x_1 from equation two, and then solve for x_2. You can do this all the way down to the end, and equation $n-1$ gives you an equation for x_{n-1} in terms of x_n. But now the last equation is

$$b_n x_{n-1} + c_n x_n = d_n$$

so you plug in your equation for x_{n-1} to eliminate x_{n-1} from the equation, and solve for x_n. Once you have this value, you recursively work back down the chain to calculate each of the x_i.

2 Heat Equation

Now we can do a similar thing to the Heat Equation. The most basic method is to look at the Heat Equation

$$u_t = u_{xx} + f(t,x)$$

and then apply the central difference approximation to u_{xx} and the forward difference approximation to u_t. When this is done, you get the equation:

$$\frac{u(t+\Delta t,x) - u(t,x)}{\Delta t} = \frac{u(t,x+\Delta x) - 2u(t,x) + u(t,x-\Delta x)}{\Delta x^2} + F(t)$$

where $F(t) = [f(t, i\Delta x)]$. We can then isolate $u(t+\Delta t,x)$ and get:

$$u(t+\Delta t,x) = u(t,x) + \frac{\Delta t}{\Delta x^2}\left(u(t,x+\Delta x) - 2u(t,x) + u(t,x-\Delta x)\right) + F(t)$$

This method is called the Forward Time Centered Space (FTCS) method, or the Euler method with central differencing on the spatial operator. When writing this using the matrix notation for the central difference operator, you get

$$U(t+\Delta t) = U(t) + AU(t) + F(t)$$

But that's not the only way you can choose to discretize this equation. You could use the right hand side at time $t+\Delta t$, receiving

$$U(t+\Delta t) = U(t) + AU(t+\Delta t) + F(t+\Delta t)$$

and all sorts of combinations. This here is equivalent to if one had done backwards differencing on the u_t and is known as Backwards Time Centered Space. Notice that in this case, you need to solve for $u(t+\Delta t)$ which requires solving a linear system:

$$(I-A)\,U(t+\Delta t) = U(t) + F(t+\Delta t)$$

3 Convergence

The Lax-Equivalence Theorem states that consistancy + stability implies convergence. Lets dive into the two parts.

3.1 Consistancy

To show that a method is consistant, you need to show that its error tends to zero as $\Delta t \to 0$. For this method, let's let $\epsilon_{i,j} = u(t_i,x_j) - \tilde{u}(t_i,x_j)$ where $\tilde{u}$ is the true solution. From our previous discussions we know:

3.2 Stability

Let's look at the error update equation. Write

$$e_i^n = u(x_j, t_n) - u_j^n$$

For e_i^n, as before, plug it in, add and subtract $u(x_j, t^n) = u_j^n$, and then we get

$$e_i^{n+1} = e_i^n + \mu\left(e_{i+1}^n - 2e_i^n + e_{i-1}^n\right) + \Delta t \tau_i^n$$

where $\tau_i^n \sim \mathcal{O}(\Delta t) + \mathcal{O}(\Delta x^2)$. Stability requires that the homogenous equation goes to zero. Another way of saying that is that the propogation of errors has errors decrease their influence over time. Thus we look at:

$$\begin{aligned} e_i^{n+1} &= e_i^n + \mu\left(e_{i+1}^n - 2e_i^n + e_{i-1}^n\right) \\ &= (1-2\mu)\, e_i^n + \mu e_{i+1}^n + \mu e_{i-1}^n \end{aligned}$$

A necessary condition for decreasing is then for all coefficients to be positive

$$1 - 2\mu \geq 0$$

or

$$\mu \leq \frac{1}{2}$$

A more satisfying way may be to look at the generated ODE

$$u' = Au$$

where A is the matrix $[\mu, 1-2\mu, \mu]$. But finding the maximum eigenvalue is non-trivial. But for linear PDEs, one nice way to analyze the stability directly is to use the Fourier mode decomposition. This is known as Van Neumann stability analysis. To do this, decompose U into the Fourier modes:

$$U(x,t) = \sum_k \hat{U}(t) e^{ikx}$$

Since $x_j = j\Delta x$, we can write this out as

$$U_j^n = \hat{U}^n e^{ikj\Delta x}$$

and then plugging this into the FTCS scheme we get

$$\frac{\hat{U}^{n+1} e^{ikj\Delta x} - \hat{U}^n e^{ikj\Delta x}}{\Delta t} = \frac{\hat{U}^n e^{ik(j+1)\Delta x} - 2\hat{U}^n e^{ikj\Delta x} + \hat{U}^n e^{ik(j-1)\Delta x}}{\Delta x^2}$$

Let G be the growth factor, defined as

$$G = \frac{\hat{U}^{n+1}}{\hat{U}^n}$$

and thus after cancelling we get

$$\frac{G-1}{\Delta t} = \frac{e^{ik\Delta x} - 2 + e^{-ik\Delta x}}{\Delta x^2}$$

Since $e^{ik\Delta x} + e^{-ik\Delta x} = 2\cos(k\Delta x)$, then we get

$$G = 1 - \mu\left(2\cos(k\Delta x) - 2\right)$$

and using the half angle formula

$$G = 1 - 4\mu\sin^2\left(\frac{k\Delta x}{2}\right)$$

In order to be stable, we require $|G| \leq 1$, which means

$$-1 \leq 1 - 4\mu\sin^2\left(\frac{k\Delta x}{2}\right) \leq 1$$

$\mu > 0$ and so ≤ 1 is simple. Since $\sin^2(x) \leq 1$, then we can simplify this to

$$-1 \leq 1 - 4\mu$$

and thus

$$\mu \leq \frac{1}{2}$$

With backwards Euler we get

$$\frac{G-1}{\Delta t} = \frac{G}{\Delta x^2}\left(e^{ik\Delta x} - 2 + e^{-ik\Delta x}\right)$$

and thus get

$$G + 4G\mu\sin^2\left(\frac{k\Delta x}{2}\right) = 1$$

and thus

$$G = \frac{1}{1 + 4\mu\sin^2\left(\frac{k\Delta x}{2}\right)} \leq 1.$$

4 Crank-Nicholson Method

The Crank-Nicholson method is also known as the Trapezoid method when applied to ODEs, and relies on the fact that

$$\frac{u(t) + u(t+\Delta t)}{2\Delta t} \sim u\left(t + \frac{\Delta t}{2}\right) + \mathcal{O}\left(\Delta t^2\right).$$

This follows from a quick Taylor expansion. Thus for

$$u_t = u_{xx}$$

we can approximate at the half time point

$$\frac{\partial u(x_i, t_j + \frac{\Delta t}{2})}{\partial t} = \frac{\partial^2 u(x_i, t_j + \frac{\Delta t}{2})}{\partial x^2}$$

and on the left do central difference, on the right do this averaging to get

$$\frac{U^{n+1} - U^n}{\Delta t} = \frac{1}{2} \left(\frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2} + \frac{U_{i+1}^{n+1} - 2U_i^{n+1} + U_{i-1}^{n+1}}{\Delta x^2} \right)$$

by re-arranging we get

$$-\mu U_{i+1}^{n+1} + 2\left(1+\mu\right) U_i^{n+1} - \mu U_{i-1}^{n+1} = \mu U_{i+1}^n + 2\left(1-\mu\right) U_i^n + \mu U_{i-1}^n$$

which we can write as

$$AU^{n+1} = BU^n$$

and get a solution

$$U^{n+1} = A \backslash BU^n$$

Van Neumann analysis gives us very similar to what we had before, arriving at

$$G = \frac{2 - 4\mu \sin^2\left(\frac{k\Delta x}{2}\right)}{2 + 4\mu \sin^2\left(\frac{k\Delta x}{2}\right)} \leq \frac{2 + 4\mu \sin^2\left(\frac{k\Delta x}{2}\right)}{2 + 4\mu \sin^2\left(\frac{k\Delta x}{2}\right)} = 1$$

Thus the Crank-Nicholson method is a second order unconditionally A-stable method.