Math 257/316: Finite difference methods

1 Finite Differences

Remember the definition of a derivative

$$f'(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{1}$$

Also recall Taylor's formula:

$$f(x+\Delta x) = f(x) + \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) + \frac{\Delta x^3}{3!} f^{(3)}(x) + \dots \tag{2}$$

or, with $-\Delta x$ instead of $+\Delta x$:

$$f(x-\Delta x) = f(x) - \Delta x f'(x) + \frac{\Delta x^2}{2!} f''(x) - \frac{\Delta x^3}{3!} f^{(3)}(x) + \dots \tag{3}$$

Forward difference

On a computer, derivatives are approximated by *finite difference* expressions; rearranging (2) gives the *forward difference* approximation

$$\frac{f(x+\Delta x) - f(x)}{\Delta x} = f'(x) + O(\Delta x), \tag{4}$$

where $O(\Delta x)$ means 'terms of order Δx', *ie.* terms which have size similar to or smaller than Δx when Δx is small.[1] So the expression on the left approximates the derivative of f at x, and has an error of size Δx; the approximation is said to be 'first order accurate'.

[1] The strict mathematical definition of $O(\delta)$ is to say that it represents terms y such that

$$\lim_{\delta \to 0} \left(\frac{y}{\delta}\right) \quad \text{is finite.}$$

So δ^2 and δ are $O(\delta)$, but $\delta^{1/2}$ or 1, for instance, are not. Another commonly used notation is $o(\delta)$, which represents terms y for which

$$\lim_{\delta \to 0} \left(\frac{y}{\delta}\right) = 0.$$

In other words, $o(\delta)$ represents terms that are *strictly* smaller than δ as $\delta \to 0$. So δ^2 is $o(\delta)$, but δ is not.

Backward difference

Rearranging (3) similarly gives the *backward difference* approximation

$$\frac{f(x) - f(x - \Delta x)}{\Delta x} = f'(x) + O(\Delta x), \tag{5}$$

which is also first order accurate, since the error is of order Δx.

Centered difference

Combining (2) and (3) gives the *centered difference* approximation

$$\frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} = f'(x) + O(\Delta x^2), \tag{6}$$

which is 'second order accurate', because the error this time is of order Δx^2.

Second derivative, centered difference

Adding (2) and (3) gives

$$f(x + \Delta x) + f(x - \Delta x) = 2f(x) + \Delta x^2 f''(x) + \frac{\Delta x^4}{12} f^{(4)}(x) + \dots \tag{7}$$

Rearranging this therefore gives the centered difference approximation to the second derivative:

$$\frac{f(x + \Delta x) - 2f(x) + f(x - \Delta x)}{\Delta x^2} = f''(x) + O(\Delta x^2), \tag{8}$$

which is second order accurate.

2 Heat Equation

Dirichlet boundary conditions

To find a numerical solution to the heat equation

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < L, \tag{9}$$

$$u(0, t) = A, \quad u(L, t) = B, \quad u(x, 0) = f(x), \tag{10}$$

approximate the time derivative using forward differences, and the spatial derivative using centered differences;

$$\frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} = \kappa \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} + O(\Delta t, \Delta x^2). \tag{11}$$

This approximation is second order accurate in space and first order accurate in time. The use of the forward difference means the method is *explicit*, because it gives an explicit formula for $u(x, t + \Delta t)$ depending only on the values of u at time t.

Divide the interval $0 < x < L$ into $N + 1$ evenly spaced intervals, with spacing Δx (*i.e.* $\Delta x = L/(N+1)$). Let $x_n = n\Delta x$, for $n = 0, 1, \ldots, N$, and divide time into discrete levels $t_k = k\Delta t$ for $k = 0, 1, \ldots$. Then seek the solution by finding the discrete values

$$u_n^k = u(x_n, t_k). \tag{12}$$

From (11), these satisfy the equations

$$\frac{u_n^{k+1} - u_n^k}{\Delta t} = \kappa \frac{u_{n+1}^k - 2u_n^k + u_{n-1}^k}{\Delta x^2}, \tag{13}$$

or, rearranging,

$$u_n^{k+1} = u_n^k + \frac{\kappa \Delta t}{\Delta x^2} \left(u_{n+1}^k - 2u_n^k + u_{n-1}^k \right). \tag{14}$$

If the values of u_n at timestep k are known, this formula gives all the values at timestep $k + 1$, and it can then be iterated again and again to march forward in time.

The initial condition gives

$$u_n^0 = f(x_n), \tag{15}$$

for all n, and the boundary conditions require

$$u_0^k = A, \quad u_{N+1}^k = B, \tag{16}$$

for all values of $k > 0$.

Stability

Note, this method will only be *stable*, provided the condition

$$\frac{\kappa \Delta t}{\Delta x^2} \leq \frac{1}{2}, \tag{17}$$

is satisfied; otherwise it will not work. One can demonstrate this by plugging the trial solution $u_n^k = \phi^k \exp(in\theta\Delta x)$ into (14) and re-arranging, using some trig identities:

$$\phi^{k+1} = f\phi^k, \qquad f = 1 - \frac{4\kappa\Delta t}{\Delta x^2} \sin^2 \left(\frac{1}{2}\theta\Delta x \right). \tag{18}$$

Hence $\phi^k = \phi^0 f^k$, which will grow with k if $|f| > 1$, leading to oscillations on the scale of the grid. To avoid this grid instability, we must demand that $-1 \leq f \leq 1$, which is guaranteed by (17) since $\sin^2 \Theta \leq 1$.

Neumann boundary conditions

With the Neumann conditions,

$$\frac{\partial u}{\partial x}(0,t) = C \qquad \text{and} \qquad \frac{\partial u}{\partial x}(L,t) = D, \tag{19}$$

we do not know the value of $u(x,t)$ on the boundaries. Hence we include the boundary points on the grid of positions where we compute the solution. We can use the same grid as before and include the two extra points $n = 0$ and $n = N+1$, giving a grid of $N+2$ points. Or, we can re-order the grid point positions so that there are still N points, with $x_1 = 0$ and $x_N = L$. Following the latter choice, we notice that the centered difference version of the first boundary condition would be

$$\frac{u(0+\Delta x,t) - u(0-\Delta x,t)}{2\Delta x} = C, \tag{20}$$

and therefore

$$u(-\Delta x,t) = u(\Delta x,t) - 2C\Delta x. \tag{21}$$

$x = -\Delta x$ is outside the domain of interest, but this relation gives its value regardless. The discretised equation (14) can then be used for $x = 0$ ($n = 1$):

$$u_1^{k+1} = u_1^k + \frac{2\kappa\Delta t}{\Delta x^2}\left(u_2^k - u_1^k - C\Delta x\right). \tag{22}$$

Similarly for $n = N$, we have

$$u_{N+1} = u_N + 2D\Delta x \qquad \text{and} \qquad u_N^{k+1} = u_N^k + \frac{2\kappa\Delta t}{\Delta x^2}\left(u_{N-1}^k - u_N^k + D\Delta x\right). \tag{23}$$

If there is a mix of boundary conditions (*e.g.* $u(0,t) = A$ and $u_x(L,t) = D$), then we can include the boundary with the Neumann condition on the mesh, but not the boundary with the Dirichlet condition, then follow the relevant scheme outlined above for each.