Discretizing the Laplace Equation and Computing a Numerical Solution: short documentation for DiscretizedLaplaceEqn.py

Varadarajan Srinivasan

Let us first define the numerical derivative of a function f(x) at a point x_0 as the function's mean rate of change between $x_0 - \Delta x$ and x_0 averaged with its mean rate of change between x_0 and $x_0 + \Delta x$. If the function is well-behaved (differentiable at x_0), this can numerically approximate the true analytical value of the derivative to an arbitrarily high accuracy.

In index notation, for a numerically defined function f_n with step size Δx, we thus define the numerical derivative as

$$\mathrm{nD}(f_n) = \frac{1}{2}\left(\frac{f_n - f_{n-1}}{\Delta x} + \frac{f_{n+1} - f_n}{\Delta x}\right) = \frac{f_{n+1} - f_{n-1}}{2\Delta x} \tag{1}$$

We can similarly define the second numerical derivative as

$$\mathrm{nD}^2(f_n) = \mathrm{nD}(\mathrm{nD}(f_n)) = \frac{\mathrm{nD}(f_{n+1}) - \mathrm{nD}(f_{n-1})}{2\Delta x} \tag{2}$$

where distribution is possible because nD is a linear transformation, in principle.

$$= \frac{1}{2\Delta x}\left(\frac{f_{n+2} - f_n}{2\Delta x} - \frac{f_n - f_{n-2}}{2\Delta x}\right) = \frac{f_{n+2} - 2f_n + f_{n-2}}{(2\Delta x)^2} \tag{3}$$

We can contract the step sizes in Eq. 3 such that $f_{n\pm2} \to f_{n\pm1}$ and $2\Delta x \to \Delta x$. Thus,

$$\mathrm{nD}^2(f_n) = \frac{f_{n+1} - 2f_n + f_{n-1}}{(\Delta x)^2} \tag{4}$$

By the same process, we can extend Eq. 4 to two dimensions:

$$\mathrm{nD}^2(f_{m,n}) = \frac{f_{m,n+1} - 2f_{m,n} + f_{m,n-1}}{(\Delta x)^2} + \frac{f_{m+1,n} - 2f_{m,n} + f_{m-1,n}}{(\Delta y)^2} \tag{5}$$

which, as we are only concerned with an evenly spaced grid ($\Delta x = \Delta y \equiv \Delta$), becomes

$$\mathrm{nD}^2(f_{m,n}) = \frac{f_{m,n+1} + f_{m,n-1} + f_{m+1,n} + f_{m-1,n} - 4f_{m,n}}{\Delta^2} \tag{6}$$

Now, we consider Laplace's Equation $\nabla^2\phi = 0$ and replace the Laplacian with our second numerical derivative. This gives us

$$\mathrm{nD}^2(\phi_{m,n}) = \frac{\phi_{m,n+1} + \phi_{m,n-1} + \phi_{m+1,n} + \phi_{m-1,n} - 4\phi_{m,n}}{\Delta^2} = 0 \tag{7}$$

$$\Rightarrow \phi_{m,n} = \frac{1}{4}(\phi_{m,n+1} + \phi_{m,n-1} + \phi_{m+1,n} + \phi_{m-1,n}). \tag{8}$$

It follows that Laplace's Equation is numerically satisfied for a function ϕ over an a×b grid when, at each point, ϕ takes the average of its value at the 4 directly neighboring grid points. Given the boundary conditions of the grid (i.e. the value of $\phi_{m,n}$ is known for all points along the edges of the grid $\phi_{m,0}, \phi_{m,b}, \phi_{0,n}, \phi_{a,n}$ for m=0,1,...,a and n=0,1,...,b), Eq. 8 generates a solvable system of equations for all the points in the interior of the grid.

One way of solving this system on a grid is to iteratively solve every point and update the points in a symmetric fashion and then iterate the whole process, which approaches arbitrarily close to the solution. That is, for an a×b grid with boundary conditions given along the grid, we can first set all the interior points in the grid to 0. Then, we take the largest rectangle that fits within the boundary and compute Eq. 8 for each of those points using the previous grid's values such that we update all of the values along that rectangle *together*. This way, no iteration of Eq. 8 uses any values computed in the same iteration step. We continue this process for progressively smaller rectangles approaching the center of the grid symmetrically.

Now, we must iterate that entire iterative process. As this metaiterative count approaches infinity, our grid values approach the solution. We can set an iteration-tolerance so that the process is repeated until the successive changes to the grid values are all within the tolerance.