Numerically Solving Poissons Equation

Maximilian Williams

September 2021

Description of the problem

We wish to solve for the streamfunction ψ given the voricity ω (equation **??**) in a discritized domain $\mathcal{D}$ in equation 1.

$$\Delta^2\psi = -\omega \tag{1}$$

$$\omega = \Delta \times \psi \tag{2}$$

We will do this numerically.

Finite Difference Approximation of derivatives

Given a function $f : R \to R$, the taylor series about the point $x \pm \delta x$ is given by equation 3:

$$f(x \pm \delta x) = f(x) \pm f^{(1)}(x)\delta x + \frac{1}{2!}f^{(2)}(x)\delta x^2 + \mathcal{O}(\delta x^3), \tag{3}$$

where $f^{(n)}(x)$ is the n^{th} derivative of f evaluated at x. By rearanging for $f^{(1)}(x)$ and $f^{(2)}(x)$ in the $x + \delta$ and $x - \delta$ variations of equation 3, the first and second order central finite difference is obtained:

$$f^{(1)}(x) = \frac{f(x + \delta x) - f(x - \delta x)}{2\delta x} + \mathcal{O}\delta x \tag{4}$$

$$f^{(2)}(x) = \frac{f(x + \delta x) - 2f(x) + f(x - \delta x)}{\delta x^2} + \mathcal{O}\delta x^2 \tag{5}$$

The Jacobi Method and its extensions In 2D spherical polars

We discritize our domain $\mathcal{D}$ splitting it into points $(r_i, \phi_j) = (i\Delta r, j\Delta\phi mod 2\pi)$, $i \in \{0, 1, ..., \frac{R}{\Delta r}\}$, $j \in \{0, 1, ..., , \frac{2\pi-1}{\Delta\phi}\}$. For simplisty we denote $f_{i,j}$ to mean f at location (r_i, ϕ_j). Equation 1 is then:

$$\frac{1}{r}\frac{\partial\psi}{\partial r} + \frac{\partial^2\psi}{\partial r^2} + \frac{1}{r^2}\frac{\partial^2\psi}{\partial\phi^2} = -\omega \tag{6}$$

Applying equations 4 and 5 to equation 6 and rearanging for $\phi_{i,j}$ we derive:

$$\psi^*_{i,j} = \frac{(\delta r\delta\psi)^2}{2(\delta\psi^2 + \frac{\delta r^2}{{r_i}^2})}(\frac{1}{r_i}(\frac{\psi_{i,j+1} - \psi_{i-1,j}}{2\delta r}) + w_{i,j} + \frac{\psi_{i+1,j} + \psi_{i-1,j}}{\delta r^2} + \frac{1}{{r_i}^2}\frac{\psi_{i,j+1} + \psi_{i,j-1}}{\delta\phi^2}). \tag{7}$$

We define the error ϵ by:

$$\epsilon = \sum_{\mathcal{D}} \mid \psi_{i,j} - \frac{(\delta r\delta\psi)^2}{2(\delta\psi^2 + \frac{\delta r^2}{{r_i}^2})}(\frac{1}{r_i}(\frac{\psi_{i,j+1} - \psi_{i-1,j}}{2\delta r}) + w_{i,j} + \frac{\psi_{i+1,j} + \psi_{i-1,j}}{\delta r^2} + \frac{1}{{r_i}^2}\frac{\psi_{i,j+1} + \psi_{i,j-1}}{\delta\phi^2}) \mid \tag{8}$$

We now perform the following algorithm.

1. pick an error $\epsilon' > 0$. When the error $\epsilon < \epsilon'$ the program will terminate.
2. Guess a solution $\psi_{i,j}$ for all points in $\mathcal{D}$
3. compute $\psi^*_{i,j}$ using equation 7
4. set $\psi_{i,j} = \psi^*_{i,j}$

5. compute ϵ, if $\epsilon < \epsilon'$ then terminate and return $\psi_{i,j}$, else repeat step 1.

Several extensions can be made to this method. The first, dubbed the Gauss-Seidel Method, replaces $\psi_{i,j}$ by $\psi^*_{i,j}$ immediately and means you only need to store one streamfunction for each point rather than two. The second extension is the Successive Over-Relaxation method, (SOR). The SOR method introduces a new parameter β that modulates how much of the previous guess contributes to the updated guess. It amounts to replacing equation 7 in step 3 to equation 9.

$$\psi^*_{i,j} = (1-\beta)\psi_{i,j} + \beta\frac{(\delta r\delta\psi)^2}{2(\delta\psi^2 + \frac{\delta r^2}{{r_i}^2})}(\frac{1}{r_i}(\frac{\psi_{i,j+1} - \psi_{i-1,j}}{2\delta r}) + w_{i,j} + \frac{\psi_{i+1,j} + \psi_{i-1,j}}{\delta r^2} + \frac{1}{{r_i}^2}\frac{\psi_{i,j+1} + \psi_{i,j-1}}{\delta\phi^2}). \tag{9}$$