PHYS486-HW4: Relaxation

Nicholas Cemenenkoff

May 12, 2018

Abstract

We implement the Jacobi relaxation method to solve Laplace's equation on a 2D square region with inner and outer boundary conditions. The outer boundary of the parent square is fixed at $V = 0$, and centered within it is a square reservoir with a fixed potential of $V = 1$ (arbitrary units). How the precision of the solution varies as a function of the number of relaxation iterations performed, how the relative error varies as a function of grid size, and how the initial conditions of the grid affect the number of iterations performed are all investigated. Due to the numerical nature of this analysis, our results show that larger grid sizes require more iterations to achieve the same relative error, relative error drops dramatically as the number of iterations increases, and more iterations are required for smaller nonzero relative error tolerance values.

1 Problem Statement

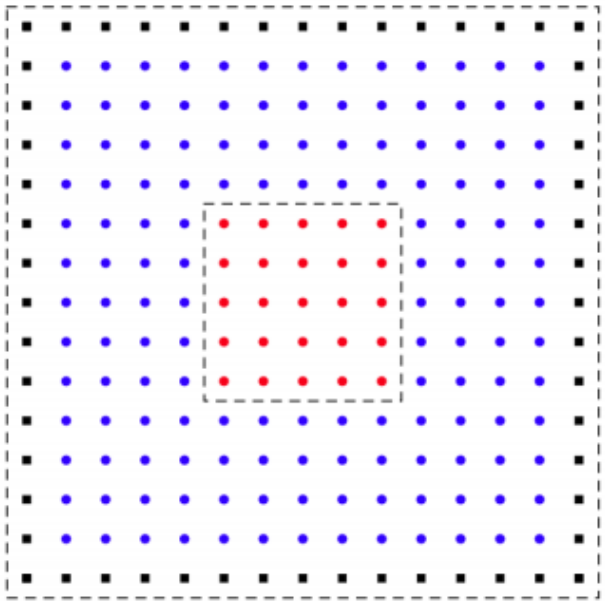

Figure 1: A 15x15 parent grid with an outer boundary (black) fixed at $V = 0$ with a 5x5 center grid (red) fixed at $V = 1$. The lattice points able to undergo relaxation in this example grid are shown in blue.

Consider a square conducting plate that is held at a fixed electric potential of 1 (in certain units). Far away from the conducting plate, the electric potential is zero. Use the relaxation method to solve Laplace's equation and determine the potential at all points on a 2D grid contained within a larger square that maintains the system's boundary conditions. The goal of this problem is to determine how the precision of your solution depends on the geometry of the system, its initial conditions, and the number of iterations of the relaxation method that are performed.

2 Numerical Solution

In 2D regions of space that do not contain any electric charges, the electric potential obeys Laplace's equation, $\nabla^2 V = \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = 0$. This differential equation can be rewritten as the finite-difference equation $V(i,j) = \frac{1}{4}\left[V(i+1,j) + V(i-1,j) + V(i,j-1) + V(i,j+1)\right]$ which defines the Jacobi relaxation method. Note, in this situation, i is the row index and j is the column index for a set of lattice points arranged in a square grid. This equation stipulates that for the i^{th} lattice point whose potential is allowed to change (i.e. not located on the inner or outer boundary), the value that it evolves into after one iteration of relaxation is the average between its up-, down-, left-, and right-neighbor. Armed with this iterative method, our main numerical implementation involves three workhorse functions:

1. `generate()` which generates a 2D initial guess array of random samples from a uniform distribution over $[0, 1)$,

2. `update()` which updates the initial guess array for one iteration of Jacobi relaxation, and

3. `relax()` which repeatedly applies `update()` until a relative error tolerance is reached (or a maximum number of iterations is reached).

`generate()` populates `relax()`-able lattice points with random numbers rather than simply zeros to ensure that enough iterations are performed in `relax()` before the global average change in the `update()`-ing grid reaches tolerance. If initialized with all 0s, the average up-down left-right difference between neighbors for each `relax()`-able lattice point will be nearly zero before the first iteration is even performed, meaning a minimum number of iterations would have to be coded into `relax()` for the code to return an accurate equilibrium solution. Initializing with random numbers between 0 and 1 not only circumvents the need to include a minimum number of iterations, but accurately represents the physics of the situation in that all points between the outer and inner boundaries should fall within this range when equilibrium is achieved.

To clarify, in this numerical implementation, relative error is defined as the difference in the total average of the average absolute difference between up-down and left-right neighbors for each updated grid sub-square for each call of `update()`. When the relative error derived from the difference of two adjacent grid configurations in `relax()` becomes less than the user-specified error tolerance, the grid stops `update()`-ing and `relax()` returns the final configuration. This method of tracking error was chosen over saving multiple 2D arrays because it compresses the error information into a single float (which makes it much easier to work with computationally).

Taken together, these three main functions solve Laplace's equation for our given boundary conditions in a reasonable amount of time and allow us to effectively probe the physics of the situation.

3 Physical Analysis

Mathematically, Laplace's equation is a second-order partial differential equation and is the simplest example of an elliptic partial differential equation. This equation is very important physically, however, because it describes behavior of the electric potential, gravitation potential, and also heat conduction. Because the heat equation contains the Laplace operator, we can visualize the process of solving Laplace's equation in the context of heat conduction. If the outer $V = 0$ boundary is considered a "cold" region whereas the inner conduction plate is "hot", successive iterations of Jacobi relaxation allow thermal energy to flow from the inner region to the outer boundary until a stationary state is reached for all points in the domain. This situation is harder visualize in terms of electric potential, but if the inner grid is only 1x1, the relaxed solution numerically approximates the electric potential of a point charge on a 2D surface (with the approximation getting better as the parent grid gets larger).

Since the solutions to Laplace's equations are harmonic functions (i.e. twice continuously differentiable functions), an accurate numerical solution must exhibit a high degree of smoothness. Additionally, harmonic functions satisfy the following maximum principle: if K is a nonempty compact subset of some open subset of $\mathbb{R}^n$ that satisfies Laplace's equation, then f restricted to K attains its maximum and minimum on the boundary of K. This corroborates the thermodynamic interpretation of the problem in that no point in the domain within the boundaries should be less than 0 or exceed 1 by conservation of energy.

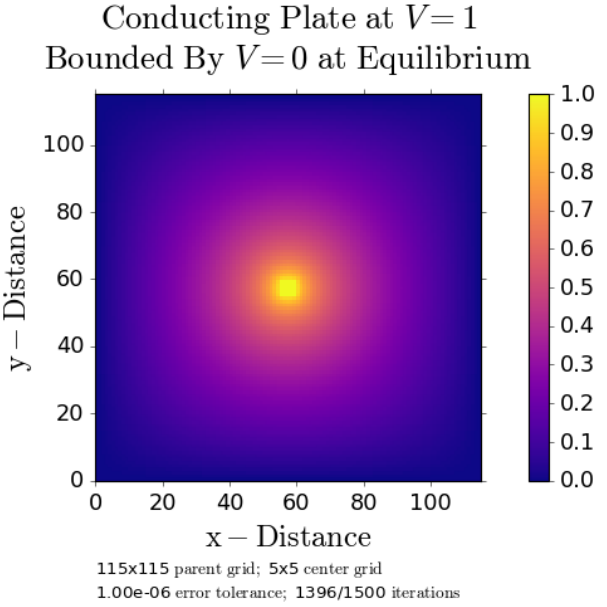

Figure 2: This final configuration for a 115x115 parent grid with outer boundary fixed at $V = 0$ and 5x5 center grid at $V = 1$ shows both a high degree of smoothness and symmetry to the boundary conditions, corroborating the both the precision and physical accuracy (respectively) of our numerical implementation.

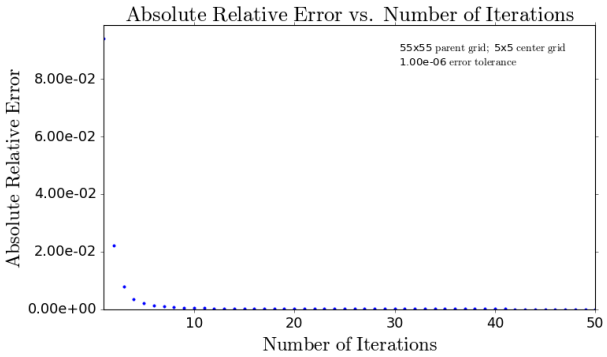

Figure 3: Notice that the absolute relative error (defined as the difference in the total average of the average absolute difference between up-down and left-right neighbors for each updated grid sub-square for each full grid update) drops off dramatically with successive iterations of Jacobi relaxation. This downturn becomes less pronounced, however, as the dimension of the parent square increases.

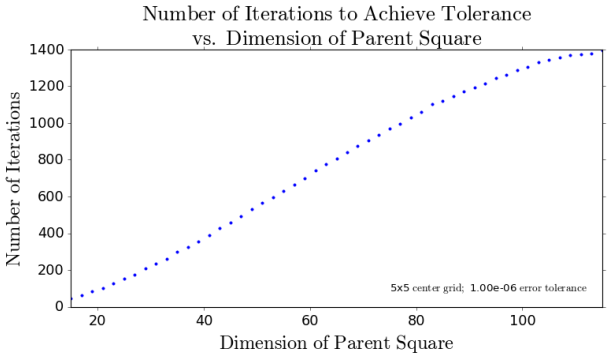

Figure 4: As the dimension of the parent square increases, the number of iterations to reach a tolerance of 10^{-6} given a 5x5 center grid increases nearly linearly. Interestingly, there is a slight upwards and the downwards curvature that almost resembles $y = \arctan x$.

4 Error Analysis

To understand the extent to which we may trust our numerically implemented Jacobi relaxation method, we investigate several visualizations of scaling relations between relative error and other parameters (look to the figure captions for analytical details). Looking to Figure 3, we confirm that relative error decreases dramatically with successive iterations. Additionally, Figure 4 shows that more iterations are needed to reach the same relative error tolerance as the dimension of the parent grid increases. To quantitatively investigate relative error as a function of distance between the boundaries and the conducting plate, Figure 5 shows that the relative error of the innermost 15x15 grid decreases significantly as the dimension of the parent grid increases. Lastly, Figure 6 shows that as the relative error tolerance decreases, the number of iterations to achieve the desired tolerance increases dramatically. Each data point in Figure 6 is taken from a new, randomly initialized starting grid configuration made by calling `generate()`. Because of the random nature of the `generate()` function, Figure 6 exhibits a small amount of noise, but a clear functional form is still evident.

Concluding, Figures 3-6 taken together confirm the precision, accuracy, and computational efficacy of our numerical implementation of Laplace's equation via Jacobi relaxation. A logical next step to extend this analysis would be to deviate from square boundary conditions, or perhaps move to 3D.

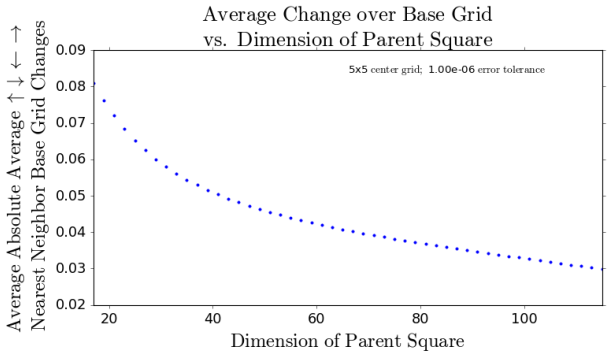

Figure 5: This plot tracks the average change over just the center 15x15 grid for several grid sizes after equilibrium has been reached. Notice that as the dimension of the parent square increases, average differences between lattice points on the centered 15x15 sub-grid become smaller.

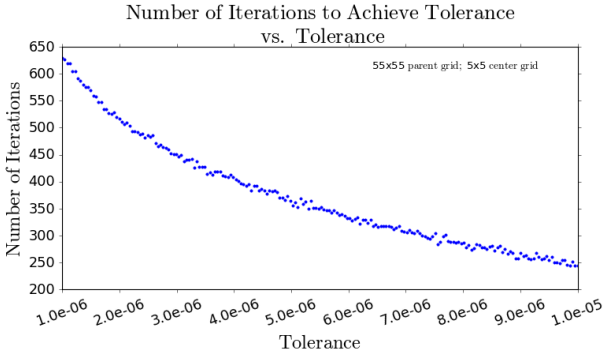

Figure 6: More iterations are required to reach tolerance for smaller tolerance values. This corroborates the intuitive notion that a higher-precision solution should arise from an increased number of iterations of Jacobi relaxation, thus lending credibility to our numerical implementation.