Exploring the Heat Equation with Multigrid

Misha Padidar*

*padidar@umail.ucsb.edu,3280559

March 18, 2017

1 Introduction

Solving the steady state heat equation is a well known problem with a defined analytic solution. This, however, makes it the perfect problem to test Multigrid as a preconditioner for Iterative methods. Multigrid is a state-of-the-art method for increasing the convergence rates of Iterative methods when solving linear systems. These methods are particularly effective when dealing with systems with sparse matrix representations, like the heat equation. The two dimensional steady state heat equation is a parabolic partial differential equation that describes heat transfer through a plate. In order to solve this problem using iterative methods, we must first transform our problem into a linear system. This can be done through discretization of the Laplacian with finite differences over our domain. We aim to solve this linear system with Multigrid methods as a preconditioner for Gauss-Siedel, and then compare the effects of Multigrid to the standard iterative technique. We will also compare our estimates to the analytic solution to check the effectiveness of our methods.

2 The Heat Equation

The two-dimensional heat equation is a partial differential equation given by

$$\Delta u = -D\frac{\partial u}{\partial t} - Q(x,y,t)$$

where D is the thermal diffusivity and $Q(x,y,t)$ is a source term. As in the steady state there is no dependence on time, thus the steady state heat equation is given by

$$\Delta u = -Q(x,y) \tag{1}$$

We approach the heat equation on the domain Ω with the boundary conditions $u(\partial\Omega,t)=0$.

3 Formulating the Problem as a Linear System

To formulate the steady state heat equation as a linear system we can use the method of finite differences. To do this we first discretize our domain into a N by N grid then at each lattice site we apply the finite difference to create an equation to model the temperature at that node. At a lattice site $x_{n,m}$ the Laplacian is modeled as

$$\frac{u_{n+1,m}-2u_{n,m}+u_{n-1,m}}{h^2}+\frac{u_{n,m+1}-2u_{n,m}+u_{n,m-1}}{h^2} = -Q_{nm} \tag{2}$$

where $u_{n,m}=u(x_{n,m})$ and $Q_{n,m}=Q(x_{n,m})$. As we have known boundary conditions given by $u(\partial\Omega)=0$ we only need solve the finite difference at the internal $(N-2)^2$ lattice sites. Between the finite difference and the boundary conditions, we have N^2 equations to model the heat profile on our domain, and N^2 variables $u_{i,j}$. Thus we can reformat our system of equations into a solvable matrix problem

$$Au = -Q \tag{3}$$

where A is the discretization of the Laplacian Operator. If we look at A a bit more closely we can note that A is sparse. More specifically the discretization of A is a block matrix of the form

$$A = \begin{pmatrix} A_1 & A_2 & A_1 & 0 & \cdots & 0 \\ 0 & A_1 & A_2 & A_1 & \ddots & \vdots \\ \vdots & \ddots & & \ddots & & 0 \\ 0 & \cdots & 0 & A_1 & A_2 & A_1 \end{pmatrix} \tag{4}$$

where A_i is the second finite difference along a rod. Now that we have a linear system at hand, we can now apply our numerical linear algebra techniques to solve this system, namely iterative methods.

4 Iterative Method and Multigrid Routine

We have two methods at hand to numerically solve the linear system formed from the discretization of the heat equation, namely Gaussian Elimination and Iterative Methods. Gaussian Elimination offers an accurate solution with the trade-off of $O(n^3)$ complexity. Iterative Methods, on the other hand, have an accuracy that the user determines, and thus a complexity that can vary. For our problem, Iterative methods are preferable as our linear map A is sparse as noted in equation (4). When used on sparse matrices, Iterative methods can yield a satisfying solution at a significantly less complexity than Gaussian Elimination. We have implemented the Gauss-Siedel method to solve the linear system. When solve the system on the lattice site $x_{n,m}$, Gauss-Siedel has the form

$$u^{k+1}_{n,m} = (D-L)^{-1}Uu^k_{n,m} + (D-L)^{-1}Q_{n,m} \tag{5}$$

where D, U, L are the diagonal, upper triangle, and lower triangle of A respectively. To increase the rate of convergence of this iterative method we implemented Multigrid as a preconditioner for Gauss-Sidel. Though more complex methods may have preferable convergence rates, we opted for using a two step V-cycle. The V-cycle runs iterations of Gauss-Siedel on coarser grids of our domain, to come up with an outline of the solution, and then uses this outline as the starting point u^0 for the iterative method on the fine grid. Our algorithm is as so:

Algorithm 1: Two Grid V-cycle

1) Iterate on fine grid with initial guess.
2) Restrict to coarse grid.
3) Iterate with guess 0.
4) Linearly interpolate back to finer grid.
5) Iterate on fine grid with interpolation as guess.

Using the V-cycle as a preconditioner for Gauss-Siedel will increase the rate of convergence when compared to the implementation of Gauss-Siedel without the Multigrid, as the iteration on the coarse grid relaxes the low frequency modes on the fine mesh grid. You may have noticed that in one iteration of this algorithm we run three iterations of the Gauss-Siedel method. This will be explained and accounted for in the results section.

5 Results

It is important to note that at no point do we solve the heat equation analytically. Rather, we compare our Multigrid method to the solution given by Numpy's Gaussian Elimination algorithm executed on the linear system constructed from the discretization of the heat problem. Although this is not the true solution, it is well known that as the number of lattice sites in our discretization increase, the closer the solution to the linear system is to the analytic solution. Thus comparison of our multigrid method to the Numpy solution is sufficient for error analysis, when using a large number of grid points. We solved the heat equation using the Gauss-Siedel iterative method with a a two step V-cycle as a preconditioner. The domain of our system was the square of side length two around the origin, $\Omega = \{(x, y)|x, y \in [-1, 1]\}$. We used boundary conditions $u(\partial\Omega) = 0$ and a source function $Q(x, y)$ given by

$$Q(x, y) = \sin\left(10((x-1)^2 + (y-1)^2)\right) \tag{6}$$

Solving our linear system via our multigrid scheme yielded the following results

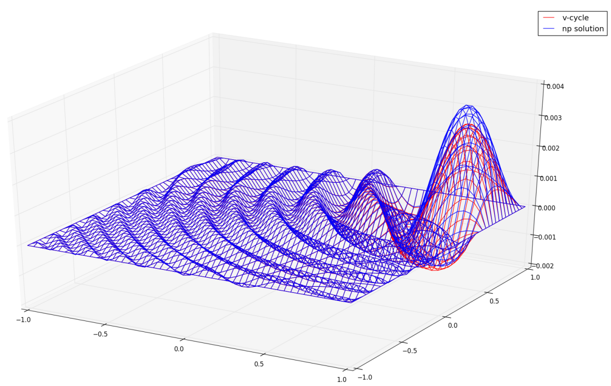

Figure 1: Temperature profile of plate from V-cycle(red) and numpy(blue). Relaxed 5 iterations on each grid with 51^2 lattice sites.

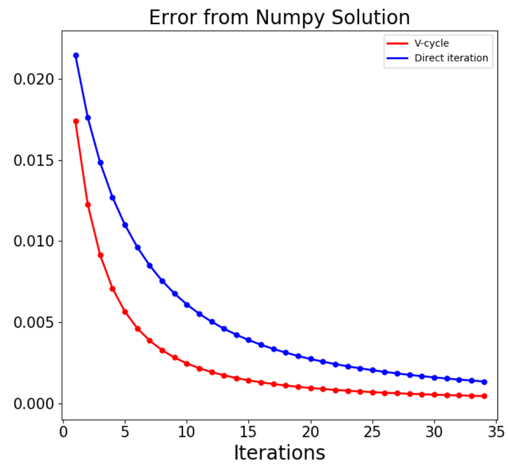

Figure 2: Error comparison of V-cycle with Gauss-Siedel method with 51^2 lattice sites. The Direct Gauss-Siedel iterated for three times the number of iterations shown.

We can see a significant increase in the convergence rates of the Gauss-Siedel method when the V-cycle is applied. This can be accounted for by the relaxation of the low frequency modes on the fine grids as performed by the V-cycle. Through *Figure 1* we can see the heat profile of our system as solved with Numpy(blue) or our Multigrid methods(red). As the number of iterations increases, our heat profile converges to the true solution rapidly. Note that we iterated the direct Gauss-Siedel method for three times as many iterations as our Multigrid method, as in every iteration of our multigrid method we perform three iterations of our iterative method. Nevertheless, to get equal accuracy to Multigrid, the standard iterative method would have to iterate orders of magnitude more times. Thus we can see that even the simplest of Multigrid methods are more effective in solving matrix systems than standard iteration.

6 Closing Thoughts

We have solved the steady-state two dimensional heat equation on numerically using Multigrid methods. Through this study it has become clear that Multigrid methods can significantly enhance the convergence rates of Iterative methods, and should be considered when solving large linear systems, or more suitably sparse linear systems. To further study this problem and the power of Multigrid methods for solving linear systems one should consider employing more complex multigrid methods such as deeper V-cycles, W-cycles, or the Full Multigrid Method(FMG). Also implementation of other iterative techniques such as Jacobi Iteration or the method of Successive Over Relaxation(SOR) could assist in finding an optimal Iterative/Multigrid method pair for solving such linear systems as the heat equation.