Using Geometric Multigrid to Solve the 2D Heat Equation

David Rower*
College of Creative Studies, University of California Santa Barbara, Santa Barbara CA 93106

We explore the speed of convergence for two iterative matrix methods: a geometric multigrid method, and the Gauss-Seidel method. To motivate the methods, we consider the problem of solving the steady-state heat equation on a square patch with a source, and formulate this problem as a linear system by discretizing the heat equation. We then discuss general iterative methods, and further motivate the geometric multigrid method. A specific example is detailed, and the methods are tested. The 2-grid V-cycle method converges much more rapidly than a direct Gauss-Seidel method.

INTRODUCTION

Linear systems appear abundantly in every field of science. It is important to know how to handle them in different contexts. For small well-posed systems, direct solvers may be used with no penalty. However, with larger systems, direct solvers may not produce results fast enough to be useful. In systems that are not well-posed, direct solvers may suffer from the propagation of numerical errors in row reduction and back substitution. There are many methods to deal with this.

In this letter, we focus on a basic example of geometric multigrid. We motivate the method with the problem of heat flowing on a square patch. The heat equation governs the behavior of this system. We discretize the equation, and formulate the problem as a linear system. We then solve this linear system by direct iterative methods, and then by the geometric multigrid method. The geometric multigrid method displays faster convergence than the Gauss-Seidel method.

PROBLEM STATEMENT

Suppose we have a square plate on $0 < x, y < 1$ with a temperature profile given by $u(x, y, t)$, and a source of heat given by $f(x, y)$ under it. We hold the edges of the plate at zero degrees, giving the boundary conditions $u(x, 1) = u(1, y) = 0$ and $u(x, 0) = u(0, y) = 0$. After letting the plate sit for a while, $u(x, y, t) \to u(x, y)$. We would like to find this equilibrium temperature profile.

RELEVANT BACKGROUND

Iterative Methods

Suppose we have a linear system of equations in the form $\mathbf{A}x = b$. Decomposing $\mathbf{A}$ into $\mathbf{B} - \mathbf{C}$, we then have

$$(\mathbf{B} - \mathbf{C})x = b \tag{1}$$
$$\mathbf{B}x = \mathbf{C}x + b \tag{2}$$
$$x = \mathbf{B}^{-1}\mathbf{C}x + \mathbf{B}^{-1}b \tag{3}$$

Let's consider the iteration

$$x^{n+1} = \mathbf{B}^{-1}\mathbf{C}x^n + \mathbf{B}^{-1}b \tag{4}$$

With $\mathbf{M} = \mathbf{B}^{-1}\mathbf{C}$ and $\mathbf{N} = \mathbf{B}^{-1}$,

$$x^{n+1} = \mathbf{M}x^n + \mathbf{N}b \tag{5}$$

Let x be the true solution satisfying $x = \mathbf{M}x + \mathbf{N}b$, and let our error after iteration n to be $\epsilon^n = x - x^n$. Then

$$\epsilon^{n+1} = x - x^{n+1} \tag{6}$$
$$= \mathbf{M}(x - x^n) \tag{7}$$
$$= \mathbf{M}\epsilon^n \tag{8}$$
$$= \mathbf{M}^{n+1}\epsilon^0 \tag{9}$$

Notice that $\epsilon^{n+1} \to 0$ when the spectral radius satisfies

$$\rho(\mathbf{M}) = \rho(\mathbf{B}^{-1}\mathbf{C}) < 1 \tag{10}$$

Gauss-Seidel

We can decompose $\mathbf{A}$ into the negation of entries below the diagonal, $\mathbf{L}$, the negation of entries above the diagonal, $\mathbf{U}$, and the diagonal, $\mathbf{D}$, so that $\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}$. For example, if $\mathbf{A}$ is a square matrix with shape 3×3, we have

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \tag{11}$$

$$\mathbf{D} = \begin{pmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{pmatrix} \tag{12}$$

$$\mathbf{L} = -\begin{pmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & a_{32} & 0 \end{pmatrix} \tag{13}$$

$$\mathbf{U} = -\begin{pmatrix} 0 & a_{12} & a_{13} \\ 0 & 0 & a_{23} \\ 0 & 0 & 0 \end{pmatrix} \tag{14}$$

Then, as detailed in the previous section, we have the iteration

$$x^{n+1} = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U}x^n + (\mathbf{D} - \mathbf{L})^{-1}b \tag{15}$$

Heat Equation

With u as the temperature profile and f as the net source/sink term, the heat equation simplifies to the following in two dimensions (on the xy plane):

$$\nabla^2 u + f(x,y) = \frac{1}{\alpha^2}\frac{\partial^2 u}{\partial t^2} \tag{16}$$

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x,y) = \frac{1}{\alpha^2}\frac{\partial^2 u}{\partial t^2} \tag{17}$$

We are interested in the steady state solution, after the plane is given some time to relax. The heat equation then becomes

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -f(x,y) \tag{18}$$

DISCRETIZING THE HEAT EQUATION

Let us divide our domain $0 < x, y < 1$ into a square grid. We can do this by subdividing the x and y axes into N portions each, giving us N^2 grid points The spacing is $h = 1/N$ for each dimension. Let the temperature profile $u(x_n, y_m) = u_{nm}$. Using a central difference scheme, we have

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{n+1,m} - 2u_{n,m} + u_{n-1,m}}{h^2} \tag{19}$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{u_{n,m+1} - 2u_{n,m} + u_{n,m-1}}{h^2} \tag{20}$$

We can then reorganize our grid as a vector of length N^2, so our temperature and heat source vectors become

$$u = \begin{pmatrix} u_{11} \\ \vdots \\ u_{1N} \\ u_{21} \\ \vdots \\ u_{NN} \end{pmatrix} \quad f = \begin{pmatrix} f_{11} \\ \vdots \\ f_{1N} \\ f_{21} \\ \vdots \\ f_{NN} \end{pmatrix} \tag{21}$$

With this format, we can form a block tridiagonal matrix

$$\mathbf{A} = \begin{pmatrix} \mathbf{D} & -\mathbf{I} & & & \\ -\mathbf{I} & \mathbf{D} & -\mathbf{I} & & \\ & \ddots & \ddots & \ddots & \\ & & -\mathbf{I} & \mathbf{D} & -\mathbf{I} \\ & & & -\mathbf{I} & \mathbf{D} \end{pmatrix} \tag{22}$$

where

$$\mathbf{D} = \begin{pmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{pmatrix} \tag{23}$$

such that we recover the linear problem

$$\mathbf{A}u = -f \tag{24}$$

GEOMETRIC MULTIGRID

Motivation

We can now solve our problem by using an iterative method on the system, although different iterative methods will have different rates of convergence. For our problem, if heat in the system could be propagated on larger length scales, and then refined using smaller length scales, the convergence of the algorithm would require less iterations.

Formulation for a 2-grid V-Cycle

Let's consider the system $\mathbf{A}x = b$ again. For a certain guess x_G, we can define the residual of this guess as the difference $r = b - \mathbf{A}x_G$. Note, the error ϵ^h of our guess causing this residual satisfies

$$\mathbf{A}(x - x_G) = b - \mathbf{A}x_G \tag{25}$$

$$\mathbf{A}\epsilon_G = r \tag{26}$$

Suppose we desire a solution accurate to length-scale h, so we solve our problem on a grid of spacing h in each dimension. We use Ω^h to represent this domain.

Starting with an arbitrary initial guess x_0, we can relax our guess with a few iterations of an iterative method to obtain a more smooth guess x^h. The residual of our current estimate is then

$$r^h = b - \mathbf{A}^h x^h \tag{27}$$

Then, we can restrict our system to the domain Ω^{2h}

$$r^h \to r^{2h} \tag{28}$$

$$\mathbf{A}^h \to \mathbf{A}^{2h} \tag{29}$$

We can relax the system $\mathbf{A}^{2h}\epsilon^{2h} = r^{2h}$, smoothing the error across our coarse grid. Then, we can prolong our error back to Ω^h

$$\epsilon^h \leftarrow \epsilon^{2h} \tag{30}$$

To correct our guess x^h, we can simply add our prolonged coarse error: $x^h \leftarrow x^h + \epsilon^h$. We can then relax our guess a few more times on Ω^h to get our final solution.

Other Cycles

The process described above involved only two grids, and transforming once to each. More involved algorithms involving multiple grids and multiple steps up and down to each one can lead to other convergence properties.

Choice of Iteration

We can relax or smooth our estimate for the solution of the linear systems we encounter by using any iterative method of our choice. In this letter, we use Gauss-Seidel with a set number of iterations during each relaxation. It is important to note that this is not always the case. By stepping to coarser grids, it is possible to reduce the system a dimension such that a set number of relaxations is just as expensive as a direct solver. In this case, a direct solver would be used.

Prolongation and Restriction

To perform the transformations between Ω^h and Ω^{2h}, we can come up with two operators. The restriction operator,

$$\mathbf{I}_h^{2h} : \Omega^h \to \Omega^{2h} \tag{31}$$

and the prolongation operator,

$$\mathbf{I}_{2h}^h : \Omega^{2h} \to \Omega^h \tag{32}$$

The restriction operator would change the dimension of our grid from $N \times N$ to $\frac{N+1}{2} \times \frac{N+1}{2}$. Similarly, the prolongation would change the dimension of our grid from $\frac{N+1}{2} \times \frac{N+1}{2}$ to $N \times N$. The details on formulating this operator for a two dimensional grid are out of the scope of this letter, but the interested reader can browse the resources listed in the citations.

2-grid V-cycle Algorithm

1. Relax with initial guess $\mathbf{A}^h x_0 = b$ on Ω^h
2. Calculate residual $r^h = b - \mathbf{A}x^h$
3. Restriction: $r^{2h} = \mathbf{I}_h^{2h} r^{2h}$, $\mathbf{A}^{2h} = \mathbf{I}_h^{2h}\mathbf{A}^h\mathbf{I}_{2h}^h$
4. Relax coarse error: $\mathbf{A}^{2h}\epsilon^{2h} = r^{2h}$ on Ω^{2h}
5. Prolongation: $\epsilon^h = \mathbf{I}_{2h}^h \epsilon^{2h}$
6. Correct final guess: $x^h \leftarrow x^h + \epsilon^h$
7. Relax final guess: $\mathbf{A}^h x_h = b$ on Ω^h

RESULTS

With a source term

$$f(x, y) = \sin[10(x-1)^2 + (y-1)^2] \tag{33}$$

we compute the solution on a grid of size 21×21, using 5 iterations during each relaxation. We plot the solution against the Numpy linalg.solve function, which gives a reasonably precise estimate of the exact solution, in Fig. 1.

Since the convergence of the iterative methods described is guaranteed with enough iterations, we are interested in the advantages of the multigrid method when compared to other iterative methods, such as Gauss-Seidel. The magnitude of the residual is plotted as a function of the number of iterations used (iterations used during each smoothing for the V-cycle) for both methods mentioned in Fig. 2.

CONCLUSION

It is apparent that the 2-grid V-cycle geometric multigrid method offers much faster convergence than a direct Gauss-Seidel iteration. Although this is the most basic multigrid method, the results were significant. This can be explored with other direct iterative methods.

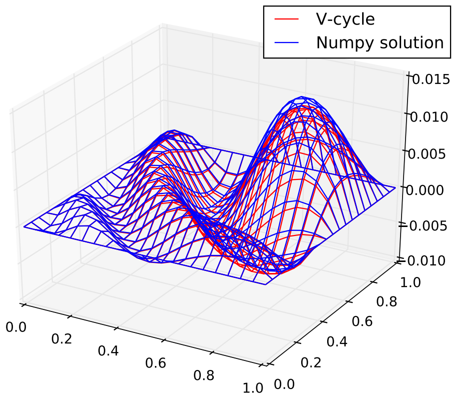

FIG. 1: The V-cycle algorithm (red) plotted against a precise estimate of the exact solution (blue). The V-cycle algorithm uses 5 iterations during each relaxation step. The solution can be improved by increasing the number of iterations used, or by performing a more complex multigrid cycle.

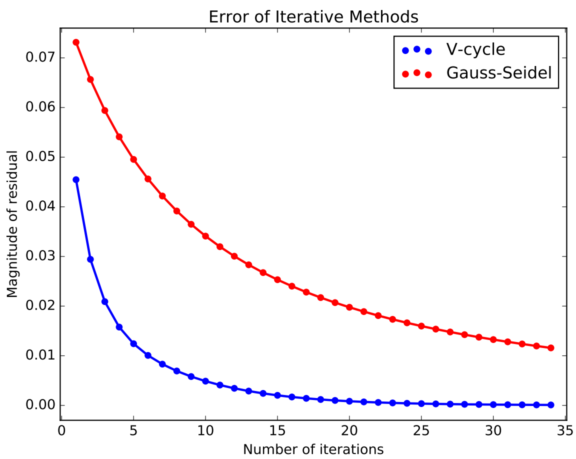

FIG. 2: The error of the V-cycle algorithm (blue) plotted with the error of a direct Gauss-Seidel iteration. The V-cycle method converges much more rapidly than the Gauss-Seidel method, as expected.

Using more complex cycles and different prolongation-restriction operators will also yield different characteristics regarding the convergence.

* Electronic address: `davidrower@umail.ucsb.edu`

[1] Briggs, W.: "A Multigrid Tutorial," SIAM (1987)
[2] Strong, D.: "Iterative Methods for Solving Ax=b - Convergence Analysis of Iterative Methods," Journal of Online Mathematics and its Applications (2005)
[3] Sderlind, G.: "Multigrid Methodology and Applications, Introduction to Multigrid Methods," Numerical Analysis, Lund University (2011)