

Multigrid Solution of Poisson's Equation in 1D

John Burkardt
Department of Scientific Computing
Florida State University

.....
http://people.sc.fsu.edu/~jburkardt/presentations/multigrid_poisson_1d.pdf

December 13, 2011

Abstract

This document is a sequel to a study of the use of a Jacobi iterative solver to compute approximate solutions to a discretization of Poisson's equation in 1D. It was found that the convergence rate for the Jacobi iteration decreased drastically as the grid size increased. For that reason, it was decided to investigate the use of a multigrid solver for the same problem. This document discusses the implementation of a multigrid solver that is based, as far as possible, on the existing Jacobi solver. Some options for the multigrid method are considered. Results are reported for a standard calculation on a single grid. A convergence study is also carried out, to compare the costs of the multigrid and Jacobi methods as the grid size increases. The hope is that the multigrid method will enable us to solve the 1D problem more quickly, and to proceed to the 2D problems that are of greater interest.

1 Ooops

I am stumbling over my preference for n to be the number of nodes, and the popular use of n for the number of intervals! Could we let M be intervals and N be nodes? $N = M + 1$? That would almost make both sides happy.

2 Shortcomings of the Jacobi Iteration

We are considering the iterative solution of the linear system associated with the solution of a discretization of a boundary value problem over a one-dimensional spatial interval $[a, b]$, known as the Poisson equation, and having the form

$$u''(x) = f(x) \text{ for } a \leq x \leq b$$

with boundary conditions given as:

$$u(a) = ua, u(b) = ub$$

By discretizing the interval into n equal subintervals and approximating the second derivative by a difference quotient, we were able to write down a system of $n+1$ equations, involving the boundary conditions at the first and last nodes, and the discretized differential equation at the $n-1$ interior nodes. To solve this discretized system, we used a Jacobi iterative solver, specifying a tolerance on the residual as our stopping criterion.

We conducted simple experiments in which we doubled the value of n and observed the number of iterations necessary to achieve convergence. The iteration count grew at a surprisingly high rate compared to n . It is a common observance that the solver on a coarse grid quickly approximates the overall behavior

of the solution, and that a fine grid solver might profitably “reference” such information in order to get a better starting point, or, indeed to speed up an iteration that seems to have stalled.

This is the inspiration for the multigrid method. In this document, we investigate methods for modifying our solution procedure for the Poisson equation to take the multigrid approach. We first make a very simple experiment, in which we use a single pair of coarse and fine grids, determining whether the coarse grid data can give a useful and efficient head start to the fine grid iteration.

The results of the two-grid experiment encourage us to proceed to a full multigrid approach, involving a sequence of grids, and a process for transferring information from fine to coarse grids and back, and to decide which particular grid to concentrate on at each phase of the iteration. Once the new implementation is completed, we return to test problems we considered for the Jacobi solver, and compare the costs.

3 The Two-Grid Procedure

We suppose that we have two geometric grids, nested in such a way that the fine grid contains all the nodes of the coarse grid. For clarity, we will assume that the fine grid is twice as fine as the coarse grid, and that the coarse grid can be constructed by selecting just the nodes with odd index from the fine grid.

Using the superscripts f and c for the fine and coarse grids respectively, it should be clear that there are simple relationships between the linear systems A^f and A^c , and the solutions u^f and u^c . In particular, since both u^f and u^c are pointwise approximations to the solution function $u(x)$, it can be expected that where the fine and coarse grids share a node, the corresponding solution vectors will be close.

As a strategy for reducing the cost of iteratively solving the linear system associated with the fine grid, we might be able to make use of an approximate solution of the coarse grid, using linear interpolation to fill in the missing entries.

Conversely, we can start with a linear system on the fine grid, and construct a related system for the coarse grid, either by simply deleting the even-indexed entries of the right hand side, or by some form of agglomeration.

Experience has shown that the coarse grid iteration is likely to converge much more quickly. Therefore, we might be able to speed up the fine grid iteration by, at some point, transferring the current fine grid approximation to the coarse grid and doing a substantial number of iterations there, before returning to the fine grid to finish up.

As a test, we know that the pure Jacobi iteration, applied to the 33 point grid $k=5$, is accepted with an RMS residual tolerance of 0.000001 after 3,088 steps. Therefore, we will make a simple test of the two grid procedure by taking 500 steps on the fine grid, taking 500 further steps on the coarse grid, and then returning to the fine grid and running until the residual tolerance is satisfied. If the two grid procedure is helpful, we might expect that the total number of iterations is significantly reduced, leaving aside the fact that the coarse grid iterations are half as expensive to carry out.

4 Moving Data Between Grids

We now outline a multigrid approach to the solution of the linear system associated with the discretized 1D Poisson equation. We assume that we are interested in a solution vector $u^{(n)}$ associated with a mesh index n , and that for this value of n , we have associated quantities $A^{(n)}$ and $f^{(n)}$. Moreover, given these quantities, we define the residual

$$r^{(n)} = f^{(n)} - A^{(n)} \cdot u^{(n)}$$

We restrict our choice of n so that it always has the form

$$n = 2^k + 1$$

where \mathbf{k} will be called the *mesh index*, with “useful” values of k starting at 1.

Now we consider the relationship between two grids determined by successive values of k . When considering a related pair of grids, we may prefer a different naming convention to associate the grid order with the corresponding variable names. Thus, we use ‘ c ’ and ‘ f ’ to distinguish the coarse and fine grids, respectively, denoting the coarse data by kc, nc, xc, uc, Ac, rc and the fine data by kf, nf, xf, uf, Af, rf .

In this case, in particular, we naturally have:

$$\begin{aligned} kf &= kc + 1; \\ nf &= 2 * nc - 1; \\ xf(2i - 1) &= xc(i), \quad i = 1 \dots nc; \\ xf(2i) &= 0.5 * (xc(i - 1) + xc(i + 1)), \quad i = 1 \dots nc - 1. \end{aligned}$$

For any n , the values of f and A are easily determined directly; however, in the case of the estimated solution vector u , we suspect that the iteration carried out on one grid may be speeded up by referencing data on another grid. We now consider how these operations may be carried out.

A mapping that takes a vector vf on the fine grid and determines a corresponding coarse grid vector vc is known as a *restriction*. A simple restriction is:

$$vc(i) = vf(2i - 1), \quad i = 1, \dots, nc.$$

although other restrictions are possible which use averaging to include data at the intermediate fine grid nodes which are otherwise ignored by the mapping shown here.

A mapping that takes a vector vc on the coarse grid and determines a corresponding fine grid vector vf is known as a *prolongation*. A simple prolongation can be found that is essentially the same as that obtaining between the coarse and fine node coordinates:

$$\begin{aligned} vf(2i - 1) &= vc(i), \quad i = 1 \dots nc; \\ vf(2i) &= 0.5 * (vc(i - 1) + vc(i + 1)), \quad i = 1 \dots nc - 1. \end{aligned}$$

It might be natural to assume that we would want to use prolongation to take an estimated solution uc on the coarse grid and prolong it to a solution uf on the fine grid. It turns out that this is precisely what we must **not** do! Instead, we will reserve the prolongation operation to distributing small corrections to the fine solution, as computed on the coarse grid. The importance of this subtle point will become clear in a demonstrative experiment we will perform shortly.

5 The V Cycle

References

- [1] RICHARD BARRETT, MICHAEL BERRY, TONY CHAN, JAMES DEMMEL, JUNE DONATO, JACK DONGARRA, VICTOR ELJKHOUT, ROIDAN POZO, CHARLES ROMINE, HENK VAN DER VORST, **Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods**, SIAM, 1994.
- [2] WILLIAM BRIGGS, VAN EMDEN HENSON, STEVE MCCORMICK, **A Multigrid Tutorial**, SIAM, 2000.

- [3] HOWARD ELMAN, ALISON RAMAGE, DAVID SILVESTER, **Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics**, Oxford, 2005.