A Multigrid Poisson Solver on General 3-Dimensional Domains

Marjan Šterk and Roman Trobec

Jožef Stefan Institute, Ljubljana, Slovenia,
marjan.sterk@ijs.si

Abstract. In this paper we present our practical experience with solving the Poisson equation on arbitrary 3-dimensional domains using finite difference approximation and Neumann boundary conditions. The equation is presented and arguments for the choice of numerical methods are given. Discretization is described and the resulting system of linear equations is analysed. Our practical implementation of the multigrid method for the presented problem on general domains is described. Results of convergence tests are given and analysed for multigrid and other, simpler methods.

1 Introduction

The Poisson equation

$$\nabla^2 u(\mathbf{x}) = f(\mathbf{x}) \tag{1}$$

is an elliptic time-independent partial differential equation (PDE) that appears in many computations, notably in heat conduction and incompressible fluid flow simulations. The latter requires coupling the Navier-Stokes equation with the continuity equation, resulting in the need to solve the Poisson equation for pressure correction at each time-step [1], which becomes the most computationally intensive part of the simulation.

For internal flow problems Neumann boundary conditions are prescribed for the pressure correction [2], i.e. the normal derivative

$$\frac{\partial u}{\partial n} = 0 \qquad \text{on all boundaries.} \tag{2}$$

There are infinitely many solutions u that satisfy (1) and (2). These solutions differ only in an additive constant. Because the absolute values of pressure are not important in this kind of problems, any particular solution has to be found.

An important field where fluid dynamics simulations are used is medicine, such as blood flow in vessels and water circulation in open heart surgeries where water is used to cool the heart muscle [3]. Simulations in 3 dimensions and irregular domains are required. The models of the body organs are usually created using bitmaps from the Visual Human Dataset or similar sources, which

R. Wyrzykowski et al. (Eds.): PPAM 2003, LNCS 3019, pp. 1052–1058, 2004.

produces 3-dimensional bitmap models [4]. It is thus natural to discretize the problem using finite differences.

The discretization with finite differences results in a sparse system of linear equations, whose sparseness pattern does not lend itself to the use of direct solvers. Iterative methods are thus needed to efficiently solve the system, such as the well-known Gauss-Seidel (GS) and SOR methods or the more sophisticated conjugate gradient (CG) method [5].

The weakness of GS is that although it reduces the high-frequency components of the error vector quickly, i.e. it smoothes the solution, many iterations are needed to reduce the low-frequency components. It is therefore beneficial to do parts of the calculation on coarser grids, where these components become high-frequency and are thus easily reduced. The solution is then interpolated to the original fine grid, where a few additional iterations are performed to obtain the final solution [6]. This idea is the basis of multigrid methods, which are generally regarded as best suited to the problems of this sort.

In the following section the discretization of the domain and the Poisson equation is described. The resulting system of linear equations is analysed. Section 3 focuses on the details of the multigrid solver for this particular problem, i.e. the interpolation and restriction operators. In Section 4, convergence rates of multigrid and other, simpler methods are given and analysed on a regular cubic domain as well as on an irregular domain.

2 Discretization

The domain is discretized to cubes of size $h \times h \times h$. Each internal cube can be either liquid or solid, while all boundary cubes are solid to form a closed cavity. The pressure correction u is defined in the centres of liquid cubes.

The second-order accurate central difference second derivative approximation is used to discretize (1) for a cube with 6 liquid neighbours to:

$$\frac{u_{x-1,y,z} - 2u_{x,y,z} + u_{x+1,y,z}}{h^2} + \frac{u_{x,y-1,z} - 2u_{x,y,z} + u_{x,y+1,z}}{h^2} + \\ + \frac{u_{x,y,z-1} - 2u_{x,y,z} + u_{x,y,z+1}}{h^2} = f_{x,y,z}, \qquad (3)$$

where $u_{x,y,z}$ stands for $u(xh, yh, zh)$. The discrete Neumann boundary conditions (2) state that $\frac{\partial u}{\partial n} = 0$ on the faces of all solid cubes. If e.g. the cube centred at $(x, y, z-1)$ is solid then the boundary condition is $\frac{\partial u}{\partial z} = 0$ on its upper face, i.e. at $(x, y, z-0.5)$. Using the central difference approximation we obtain

$$\left.\frac{\partial u}{\partial z}\right|_{x,y,z-0.5} = \frac{u_{x,y,z} - u_{x,y,z-1}}{h} = 0 \;\Rightarrow\; u_{x,y,z} = u_{x,y,z-1}. \qquad (4)$$

The latter form allows us to remove both $u_{x,y,z}$ and $u_{x,y,z-1}$ from (3) in this case so that no values outside the domain appear in the solution. In general,

numbering all the liquid cubes in the domain gives the linear system $A\mathbf{u} = \mathbf{b}$, where $\mathbf{u} = \{u_{x,y,z}\}$, $\mathbf{b} = \{h^2 f_{x,y,z}\}$ and

$$A = [a_{i,j}]_{N\times N} \;\; ; \;\; a_{i,j} = \begin{cases} 6-s & i=j, \text{the cube } i \text{ has } s \text{ solid neighbours} \\ -1 & i \neq j, \text{cubes } i \text{ and } j \text{ are neighbours} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

N stands for the total number of liquid cubes in the domain. The discretization for the case of different resolutions for the dimensions x, y and z can be derived in a similar manner.

The matrix A is symmetric. Its rank is $N-1$ because the sum of rows is 0. Also $\sum b_i = 0$ so, as expected, the system has infinitely many solutions, any of which is to be found. Iterative methods based on the Gauss-Seidel iteration, such as GS itself, SOR and multigrid, can be used on this singular system, which follows from the eigenvectors of the Gauss-Seidel iteration matrix

$$G = M^{-1}N = (D+L)^{-1}(-U), \tag{6}$$

where D is the diagonal of A, and U and L the strict upper and lower triangular portions of A, respectively. The convergence criterion requests that all the eigenvalues of G have moduli less than 1 [5]. In our case, all the eigenvalues' moduli are less than 1 except for a single eigenvalue 1 with the corresponding eigenvector $\mathbf{v}_1 = (1, 1, \ldots, 1)^T$. The error will thus be reduced through the iterations in all directions except for $\mathbf{v}_1$. Since our goal is to find any of the solutions differing in an additive constant, i.e. in a multiple of $\mathbf{v}_1$, changes of the approximate solution in the direction of $\mathbf{v}_1$ are irrelevant.

3 Multigrid Implementation

We implemented the full multigrid method [7,8]. First, the system is restricted to the coarsest grid. The resulting small system is solved directly. The solution is then interpolated to twice-finer grid, where the smoother (a few iterations of GS method) is applied. Further transitions between grid levels are shown in Figure 1. Theoretically this method achieves constant number of iterations, independent of the system size.

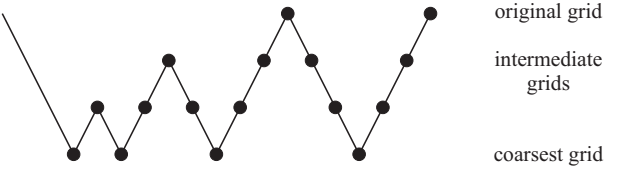

Fig. 1. The full multigrid method consists of restriction (descending lines), interpolation (ascending lines) and applying the smoother (dots).

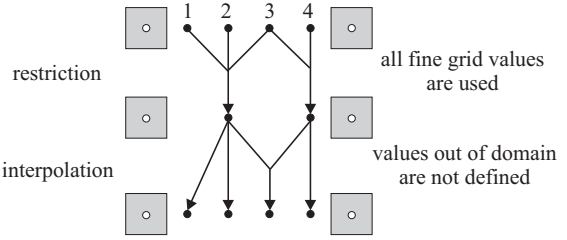

Fig. 2. The restriction and interpolation operators.

The restriction and interpolation operators have to be chosen carefully in order not to introduce a larger error into the solution than can be reduced by the subsequent application of the smoother. We used operators that are shown in Figure 2. Restriction uses a weighted average of all fine grid values to obtain coarse grid values. Note that on an irregular domain the system matrix A is derived implicitly from the domain shape, i.e. the solid-liquid pattern of the cubes. This pattern is restricted using the same restriction operator, which gives the domain shape on the coarser grid. The system matrix on the coarser grid is again derived implicitly from the shape.

In the inner parts of the domain the interpolation is taken as the transpose of the restriction operator. Fine grid values next to a boundary must be obtained without using values outside the domain, which are not prescribed by Neumann boundary conditions. Fine grid values next to a boundary are thus equal to those $1.5h$ away from the boundary, which most closely follows the boundary conditions (see the lower left arrow in Figure 2).

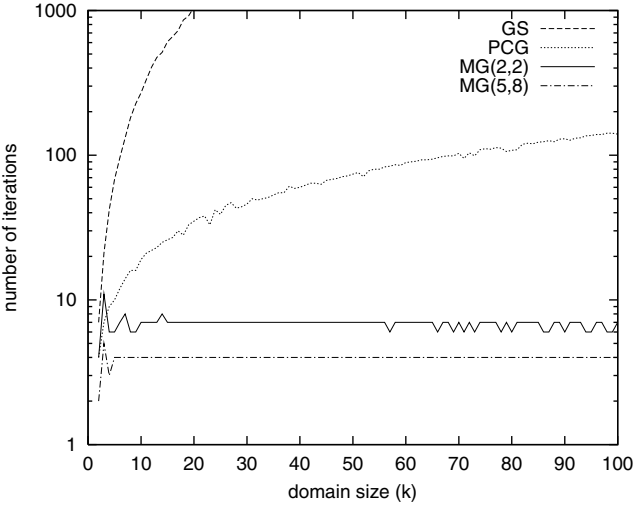

Fig. 3. Comparison of methods on a regular cubic domain.

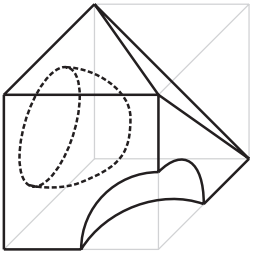

Fig. 4. The test irregular domain.

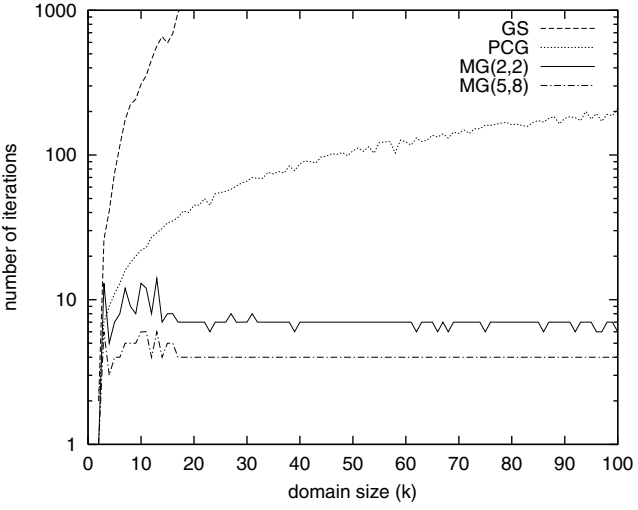

Fig. 5. Comparison of methods on the irregular domain.

4 Results

The solver was first implemented in Matlab for regular cubic domains in order to analyse the linear system, test various methods, and verify the results by comparing with those obtained by the built-in solver. The following methods were tested: Gauss-Seidel (GS), conjugate gradient with incomplete Cholesky preconditioning (PCG), which reduces the number of iterations for the CG method approximately by a factor of 3, and full multigrid (MG). High-performance general-domain versions of the methods were then implemented in C++ as a part of a fluid-flow simulation project [9].

Figure 3 shows the number of iterations needed to solve the Poisson equation on a regular cubic domain consisting of $k \times k \times k$ points. MG(i, j) stands for full multigrid method with i Gauss-Seidel iterations at each grid level and j or more grid points at the coarsest level. The iteration stopping criterium was $||\mathbf{r}||_\infty \leq 10^{-6}$.

The Gauss-Seidel and PCG methods require approximately $1,5k^{2,1}$ and $1,4k$ iterations, respectively. Both are outperformed by the full MG method, where the number of iterations is independent of k. MG(5,8) solves the system in about 20 % less time than MG(2,2).

The methods were also tested on an irregular domain consisting of a trimmed cube hollowed out by two spheres, which is shown in Figure 4.

The narrow bands, e.g. in the far right corner, are potential trouble areas for multigrid because their shape will inevitably be lost on coarse grids. However, Figure 5 shows that the number of iterations of the full MG method on the irregular domain remains the same as for a regular domain. The number of iterations for GS and PCG methods increases.

5 Conclusions

In this work a multigrid solver for the Poisson equation with Neumann boundary conditions is described. It uses finite difference approximation and works on arbitrary 3-dimensional domains. It was developed together with Gauss-Seidel and conjugate gradient solvers as a part of a fluid flow simulation project. The performance of all solvers is compared. The results show that the multigrid outperforms other methods because the number of iterations is independent of the domain size, provided that the restriction and interpolation operators are implemented correctly. It is also shown that irregularity of the domain is not a significant problem even though the shape cannot be adequately represented on the coarser grids.

Directions for future work include improving the performance of the presented solver by using an optimised numerical library, e.g. Blitz++ [10]. A further improvement of the fluid-flow simulation would be the use of finite element method, which would presumably result in a smaller but less regular Poisson system matrix.

References

1. C. W. Hirt and J. L. Cook. Calculating three-dimensional flows around structures. *J. Comput. Phys.*, 10:324–340, 1972.
2. C. A. J. Fletcher. *Computational Techniques for Fluid Dynamics.* Springer Verlag, 1988.
3. R. Trobec, B. Slivnik, B. Geršak, and T. Gabrijelčič. Computer simulation and spatial modelling in heart surgery. *Computers in Biology and Medicine*, 4:393–403, 1998.
4. R. Trobec, G. Pipan, P. Trunk, and J. Močnik. Spatial heart model derived from VHD. In *Bioimages for Europe '99, 2nd International Workshop of the Visible Human Dataset*, Milan, 1999.
5. M.T. Heath. *Scientific Computing: An Introductory Survey, 2nd Ed.* WCB/ McGraw-Hill, 2002.
6. A. Brandt. Multi-level adaptive solutions to boundary value problems. *Math. Comput.*, 31:333–390, 1977.

7. G. Golub and J. M. Ortega. *Scientific Computing - An Introduction with Parallel Computing*. Academic Press Inc., Boston, 1993.
8. P. Wesseling. *An Introduction to Multigrid Methods*. John Wiley and Sons, 1991.
9. M. Šterk, R. Trobec, and M. Praprotnik. Comparison of incompressible fluid flow simulation methods. In *Parallel Numerics '02, Theory and Applications*. Jožef Stefan Institute and University of Salzburg, 2002.
10. T. Veldhuizen. Blitz++ user's guide, 2001.