Journal of Scientific Computing, Vol. 17, Nos. 1–4, December 2002 (© 2002)

A Hierarchical 3-D Poisson Modified Fourier Solver by Domain Decomposition

Moshe Israeli,[1] Elena Braverman,[2] and Amir Averbuch[3]

Received August 30, 2001; accepted (in revised form) November 2, 2001

We present a Domain Decomposition non-iterative solver for the Poisson equation in a 3-D rectangular box. The solution domain is divided into mostly parallelepiped subdomains. In each subdomain a particular solution of the non-homogeneous equation is first computed by a fast spectral method. This method is based on the application of the discrete Fourier transform accompanied by a subtraction technique. For high accuracy the subdomain boundary conditions must be compatible with the specified inhomogeneous right hand side at the edges of all the interfaces. In the following steps the partial solutions are hierarchically matched. At each step pairs of adjacent subdomains are merged into larger units. In this paper we present the matching algorithm for two boxes which is a basis of the domain decomposition scheme. The hierarchical approach is convenient for parallelization and minimizes the global communication. The algorithm requires $O(N^3 \log N)$ operations, where N is the number of grid points in each direction.

KEY WORDS: 3-D Poisson solver; modified Fourier method; domain decomposition.

1. INTRODUCTION

Fast and accurate solution of elliptic equations is an important step towards resolution of problems which appear in computational physics or fluid dynamics (CFD). These equations arise in the determination of the pressure field for incompressible CFD, in the implicit solution of viscous and heat transfer problems, in the solution of the Maxwell equations for lithographic exposure, in the solution of reaction-diffusion equations for baking and dissolution processes in semiconductor manufacture, and in

[1] Computer Science Department, Technion-Israel Institute of Technology, Haifa 32000, Israel. E-mail: israeli@cs.technion.ac.il

[2] Computer Science Department, Technion, Haifa 32000, Israel. E-mail: maelena@cs.technion.ac.il. On leave at Yale University, Department of Mathematics, 10 Hillhouse Avenue, New Haven, Connecticut 06520-8283.

[3] School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: amir@math.tau.ac.il

many other applications. For instance, the semi-implicit discretization in time (see [10]) of the incompressible Navier–Stokes equations gives rise to one Poisson equation for the pressure and three modified Helmholtz equations for the momentum equations.

We present a non-iterative domain decomposition algorithm for a high order (spectral) solution of the 3-D Poisson equation. Most Poisson and Laplace solvers were initially developed for the 2-D case, such as Fast Multipole Method (FMM) in [9], the boundary integral method in [11] and modified Fourier method in [3, 4]. An adaptive algorithm for a fast solution of the 2-D Poisson equation by decomposition of the domain into square domains and the subsequent matching of these solutions by the FFM was developed in [8].

The present 3-D algorithm is based on the fast spectral Poisson solver developed in [5]. It incorporates the application of the FFT with a preliminary subtraction technique. In this paper we generalize the algorithm of [2] to 3-D case. The efficiency of the algorithm is especially vital for 3-D problems which usually require heavy computations. The method which is presented here enjoys the properties of the 2-D algorithm: fast convergence (i.e., small N necessary to achieve the prescribed accuracy) and comparatively small number of operations per point ($O(\log N)$).

We present a high order "corrected" 3-D Fourier spectral algorithm for the solution of elliptic equations based on domain decomposition. Similar algorithms in 2-D case were developed in [2, 8]. For illustration we consider a box with eight subdomains (see Fig. 1).

We solve the Poisson equation

$$\Delta u = u_{xx} + u_{yy} + u_{zz} = f(x, y, z) \tag{1}$$

in 3-D domain Ω with Dirichlet

$$u = \Phi(x, y, z) \qquad \text{on } \partial\Omega \tag{2}$$

boundary conditions by the Domain Decomposition (DD) methods. We assume that Φ is twice differentiable and compatible with the right hand side in the corners.

The proposed algorithm consists of the following steps.

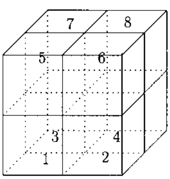

Fig. 1. The domain is decomposed into 8 subdomains.

Step 1. Introduction of consistent boundary conditions at the interfaces. Each subdomain is covered by a $N \times N \times N$ grid. The right hand side of the Poisson equation is evaluated on the grid knots and Dirichlet boundary conditions φ are determined at the interfaces, taking care to avoid singularities at the corners and edges. These boundary conditions in each subdomain match for adjacent subdomains and also match the right hand side at the edges where the Laplacian can be computed by the boundary conditions.

Step 2. Solution of the Poisson equation in each subdomain with the prescribed boundary conditions. The solution of the Poisson equation is found in each subdomain using modified Fourier spectral method of high accuracy developed in [5, 6]. The complexity of this step is $O(N^3 \log N)$.

Step 3. Computation of the jumps of the first derivative at the interfaces. The solution obtained with the prescribed boundary conditions has discontinuities in the normal derivatives at interfaces. In order to remove these discontinuities the difference between the first derivatives on the two sides of the interfaces is computed.

Step 4. Matching of discontinuities. Harmonic functions are added to both sides of each interface. In this procedure and the successive matching procedures the "correction" functions can be evaluated only on the boundaries of two adjacent (merged) subdomains. The accumulation of all the corrections determines the final local boundary conditions. The complexity of this step is $O(k^2 N^2 \log N)$, where k is the number of boxes that are matched simultaneously, N is the number of collocation points in each direction in each box.

Step 5. Solution of the homogeneous equations in subdomains. The Laplace equation is solved in each subdomain with the boundary conditions which were determined at the previous step. Combining this result with the solution of the Poisson equation leads to a smooth global solution. The complexity of this step is $O(N^3 \log N)$.

Step 6. Global solution. The smooth global solution which was computed at the previous step does not match the prescribed values at the boundaries. A global Laplace equation is solved to satisfy the prescribed boundary conditions of the global domain. The complexity of this step is $O(N^3 \log N)$.

In our previous work by [5, 6] we showed that the solution in a 3-D box (i.e., in each subdomain) can be computed to high order accuracy (corresponding to the second, fourth order of accuracy etc.) depending on the order of our "subtraction procedure."

The interface jump removal can become inexpensive if initially (and later in each step) only adjacent boxes are matched. The present hierarchical approach matching only two adjacent boxes at each level requires only local corrections at the boundaries of these boxes, such that only the solution in the adjacent subdomains are coupled at each matching step, then

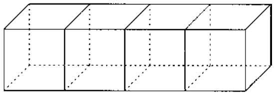

Fig. 2. The domain is decomposed into L subdomains (here $L=4$).

these joint subdomains are matched etc. If originally we had k^3 3-D subdomains, then after $3\log k$ steps we obtain a smooth global solution.

In this paper we present the basic algorithm which is the solution of the Poisson equation and matching of solution in two adjacent boxes. Then the matching of 3-D box divided into eight subdomains (see Fig. 1) can be completed in three steps: first box 1 is matched with box 2, 3 with 4, 5 with 6, 7 with 8. Then the merged box 1, 2 is matched with 3, 4 and 5, 6 with 7, 8. Finally, the whole slice 1, 2, 3, 4 is patched to 5, 6, 7, 8.

For L boxes in one line (see Fig. 2) the hierarchical matching takes $\log L$ steps. Each original or merged subdomain is matched to an adjacent subdomain. When non-hierarchical matching was performed, the influence of each derivative jump should be computed at each interface.

2. BOUNDARY CONDITIONS AT THE INTERFACES

Boundary conditions at the interfaces should satisfy the Poisson equation at the edges. First consider the simplest case when two subdomains are matched and the boundary conditions are defined at $x=0, 2,\ y=0, 1,\ z=0, 1$. Thus we have to introduce the boundary conditions only at the interface $x=1$: $u(1, y, z)=\varphi(y, z)$.

In the right hand side of the following equality

$$\frac{\partial^2 u}{\partial y^2}(1, y, 0)+\frac{\partial^2 u}{\partial z^2}(1, y, 0)=f(1, y, 0)-\frac{\partial^2 u}{\partial x^2}(1, y, 0)\equiv g_1(y) \tag{3}$$

the first function is known and the second one can be computed by the boundary condition at the interface $z=0$. Similarly

$$\frac{\partial^2 u}{\partial y^2}(1, y, 1)+\frac{\partial^2 u}{\partial z^2}(1, y, 1)=f(1, y, 1)-\frac{\partial^2 u}{\partial x^2}(1, y, 1)\equiv g_2(y) \tag{4}$$

$$\frac{\partial^2 u}{\partial y^2}(1, 0, z)+\frac{\partial^2 u}{\partial z^2}(1, 1, z)=g_3(z), \qquad \frac{\partial^2 u}{\partial y^2}(1, 1, z)+\frac{\partial^2 u}{\partial z^2}(1, 1, z)=g_4(z) \tag{5}$$

are evaluated. Then the 2-D function $\varphi(y, z)$ satisfies the Poisson equation

$$\Delta\varphi(y, z)=g(y, z) \tag{6}$$

The function $\varphi(y, z)$ is known at the wire frame ABCD together with its 2-D Laplacian g. We should construct a smooth function g inside the rectangle ABCD and assume $\varphi(y, z)$ is a solution of (6) with the above boundary conditions at the frame ABCD. One can obtain φ as a solution of a biharmonic equation.

Function $g(x, y)$ is smooth and satisfies at the frames of the interface

$$g(y, 0) = g_1(y), \qquad g(y, 1) = g_2(y), \qquad g(0, z) = g_3(z), \qquad g(1, z) = g_4(z) \tag{7}$$

We construct a smooth function g satisfying boundary conditions (7). In principle we could consider g to be a harmonic function if the following equality were satisfied at corner (1, 1) in (y, z)

$$a = \frac{\partial^2 g_1}{\partial y^2}(1) = -\frac{\partial^2 g_3}{\partial z^2}(1) = -b \tag{8}$$

and the same in all the other corners. However generally $a \neq -b$. Nevertheless g can be presented as a sum of a known function h which subtracts Laplacian in the corners and some harmonic function. We can subtract the Laplacian in each corner separately. Any function that vanishes at all the corners and has vanishing second derivatives in all the corners but (1, 1) can be chosen as a subtraction function $h_{(1,1)}$. For example, the difference of the normalized hyperbolic sine functions of y vanishes in all the corners and has vanishing second derivatives for $y = 0$. So the function $z(\sinh(\lambda_1 y)/\sinh \lambda_1 - \sinh(\lambda_2 y)/\sinh \lambda_2)$ vanishes in the corners and has vanishing second derivatives in z everywhere and in y for $z = 0$ or $y = 0$. Hence the function

$$h_{(1,1)}(y, z) = \frac{a+b}{\lambda_1^2 - \lambda_2^2}\left(\frac{\sinh(\lambda_1 y)}{\sinh \lambda_1} - \frac{\sinh(\lambda_2 y)}{\sinh \lambda_2}\right) \tag{9}$$

has the same Laplacian at (1, 1) as g. The subtraction of $h_{(1,1)}$ (9) does not change the Laplacian in the other corners. Three other corners are treated similarly. Then $g - h_{(1,1)} - h_{(1,0)} - h_{(0,1)} - h_{(0,0)}$ can be constructed as a harmonic function with boundary conditions obtained as a difference. Once g is known then $\varphi(y, z)$ is assumed to be a solution of the 2-D Poisson equation (6).

3. MATCHING OF BOXES

After we have found a solution of the Poisson equation in each subdomain, there is a jump in the first derivative in x-direction. To match the solutions, we add symmetric harmonic functions to the solution. Among harmonic functions, multiplications of sine and hyperbolic sine functions are appropriate for matching and subtraction, see [2, 3, 5, 6]. These functions vanish at zero, in addition, the hyperbolic sine function quickly decays when approaching zero. The matching is performed in the following way. Let the jump in the first derivative be

$$\frac{\partial u}{\partial x}(1+, y, z)-\frac{\partial u}{\partial x}(1-, y, z)=\phi(y, z) \tag{10}$$

Since the first derivative in x is continuous at the frame $x=1$, $y=0, 1$ or $z=0, 1$, then $\phi(y, z)$ vanishes at the boundary. Then the second derivative of ϕ in z is periodic at the boundaries $z=0, 1, 0 \leqslant y \leqslant 1$, and so is the second derivative of ϕ in y at the boundaries $y=0, 1, 0 \leqslant z \leqslant 1$. We expand the second derivative into the sine series

$$\frac{\partial^2 \phi}{\partial z^2}(y, 0)=\sum_k a_k \sin(\pi k y) \tag{11}$$

After the following harmonic functions are added to the solution

$$\sum_k \frac{a_k}{2(\lambda_1^2-\lambda_2^2)} \sin(\pi k y)\left[\frac{\sin(\lambda_1(1-z))}{\sin(\lambda_1)} \frac{\sinh(\sqrt{\lambda_1^2+\pi^2 k^2}\,(2-x))}{\sqrt{\lambda_1^2+\pi^2 k^2}\cosh(\sqrt{\lambda_1^2+\pi^2 k^2})}\right.$$
$$\left.-\frac{\sin(\lambda_2(1-z))}{\sin(\lambda_2)} \frac{\sinh(\sqrt{\lambda_2^2+\pi^2 k^2}\,(2-x))}{\sqrt{\lambda_2^2+\pi^2 k^2}\cosh(\sqrt{\lambda_2^2+\pi^2 k^2})}\right] \tag{12}$$

for $x \geqslant 1$ and a symmetric with respect to the plane $x=1$ function for $x \leqslant 1$, the second derivative of the solution in y is matched at the edge $y=0, 0 \leqslant z \leqslant 1$. Function (12) and a symmetric one vanish at the interfaces $x=0, 2$. Similar functions are added for three other boundaries.

After this the remaining jump in the derivative $\phi_1(y, z)$ vanishes along all the boundary together with its second derivatives in y and in z. Thus the function $\phi_1(y, z)$ can be accurately expanded into 2-D sine series

$$\phi_1(y, z)=\sum_i \sum_k a_{ik} \sin(\pi i y) \sin(\pi k z) \tag{13}$$

Then after adding the function

$$\sum_i \sum_k \frac{1}{2} a_{ik} \sin(\pi i y) \sin(\pi k z) \frac{\sinh(\pi \sqrt{i^2+k^2}\,(2-x))}{\pi \sqrt{i^2+k^2}\cosh(\pi \sqrt{i^2+k^2})} \tag{14}$$

for $x \geqslant 1$ and a symmetric with respect to the plane $x=1$ function for $x \leqslant 1$ we obtain the matched first derivative in x at $x=1$. Function (14) is harmonic, provides an appropriate derivative jump and, besides, vanishes at all the interfaces except $x=1$ (see Fig. 3). Similarly the derivative is matched at the other interfaces. Certainly we do not evaluate all the "addition" functions in all the domain but at the boundary and other interfaces only.

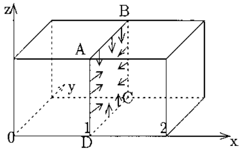

Fig. 3. We define the primary values at the interface $x=1$; at the frame of the interface the values and the second normal derivatives are known.

Afterwards the sum of values of all these functions is computed and the Laplace equation with the corresponding boundary conditions is solved. The addition of this solution to the previous ones in each subdomain matches the first derivative of the solution at interfaces, i.e., we obtain a smooth solution.

4. NUMERICAL RESULTS

Assume that u is the exact solution and u' is the computed solution. Let u_i and u_i' be the values of u and u_i' in the collocation points, respectively. In the following examples we will use the following measures to estimate the errors:

$$\varepsilon_{\mathrm{MAX}} = \max \|u_i' - u_i\|, \quad \varepsilon_{\mathrm{MSQ}} = \sqrt{\frac{\sum_{i=1}^{N} (u_i' - u_i)^2}{n}}, \quad \varepsilon_{\mathscr{L}^2} = \sqrt{\frac{\sum_{i=1}^{N} (u_i' - u_i)^2}{\sum_{i=1}^{N} u_i^2}}$$

Example 1. We solve the Poisson equation in $[0,1]\times[0,1]\times[0,2]$ with the right hand side and the boundary conditions corresponding to the exact solution $u(x,y,z)=\cos(x-0.4)\cos(y-0.5)\cos(z-1.0)$. Numerical results are presented in Table I.

Example 2. We solve the Poisson equation in $[0,1]\times[0,1]\times[0,2]$ with the right hand side and the boundary conditions corresponding to the exact solution $u(x,y,z)=\exp\{-5((x-0.4)^2+(y-0.5)^2+z^2)\}$. Numerical results are presented in Table II.

Table I. MAX, MSQ and $\mathscr{L}^2$ Errors for the Solution of the Poisson Equation by Matching Two Boxes for the Exact Solution $u(x,y,z)=\cos(x-0.4)\cos(y-0.5)\cos(z-1.0)$

$N_x \times N_y \times N_z$	$\varepsilon_{\mathrm{MAX}}$	$\varepsilon_{\mathrm{MSQ}}$	$\varepsilon_{\mathscr{L}^2}$
$8\times8\times8$	2.7e-5	5.2e-6	2.1e-5
$16\times16\times16$	1.3e-6	2.5e-7	9.7e-7
$32\times32\times32$	9.7e-8	2.0e-8	7.6e-8
$64\times64\times64$	6.7e-9	1.4e-9	5.5e-9

Table II. MAX, MSQ and $\mathscr{L}^2$ Errors for the Solution of the Poisson Equation by Matching Two Boxes for the Exact Solution Being a Steep Gaussian Function $u(x, y, z) = \exp\{-5((x-0.4)^2+(y-0.5)^2+z^2)\}$

$N_x \times N_y \times N_z$	$\varepsilon_{\mathrm{MAX}}$	$\varepsilon_{\mathrm{MSQ}}$	$\varepsilon_{\mathscr{L}^2}$
$8 \times 8 \times 8$	2.0e-3	3.2e-4	1.6e-3
$16 \times 16 \times 16$	3.5e-5	6.3e-6	3.1e-5
$32 \times 32 \times 32$	6.2e-7	1.2e-7	6.1e-7
$64 \times 64 \times 64$	2.5e-8	2.9e-9	1.4e-8

We observe that in Examples 1, 2 the rate of convergence is $O(h^4)$ (the error decays 16 times if the number of points in each direction is doubled). This matches the theoretical results of [7], see also [5] for 3-D case.

5. SUMMARY AND DISCUSSION

The above algorithm for matching of two adjacent subdomains is a part of a more general algorithm. Consider the case when the domain is divided into some parallelepiped subdomains (see Fig. 4). We can set arbitrary values of solution u at the horizontal interfaces.

If u is assumed to vanish at the horizontal interfaces then its derivatives in x and y vanish and therefore the second derivative in z can be evaluated. Let us define f_1, f_2 as in 2-D case, these are functions of z which vanish at the ends of the vertical edges, with the second derivatives computed above. Using the above function, we can interpolate Dirichlet values for u in the xz plane. Thus the compatible boundary conditions are determined at the cylindrical envelope.

The boundary conditions at yz planes are defined as in Section 3.

As a summary the following remarks can be done.

1. The developed spectral algorithm for the Poisson equation achieves high accuracy of 10^{-7}–10^{-8} for $64 \times 64 \times 64$ points in the subdomains. Based on [6], similar algorithm can be developed for Helmholtz type 3-D equations.
2. The algorithm takes $O(N^3 \log N)$ operations, where N is the number of points in each direction. Similar to 2-D case, the hierarchical matching procedure reduces the number of computations.

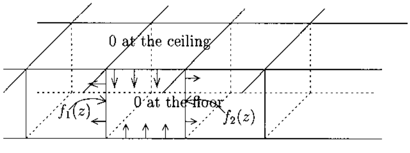

Fig. 4. At the initial step we define consistent boundary conditions at all the interfaces.

3. The algorithm is expected to be applicable for parallel implementation as its previous 2-D version developed in [1].

ACKNOWLEDGMENTS

The research of M. Israeli was supported by the VPR fund for promotion of research at the Technion.

REFERENCES

1. Averbuch, A., Braverman, E., and Israeli, M. (2000). Parallel adaptive solution of a Poisson equation with multiwavelets. *SIAM J. Sci. Comput* **22**, 1053–1086.
2. Averbuch, A., Braverman, E., and Israeli, M. (2001). A new low communication parallel algorithm for elliptic partial differential equations. In *Parallel Computational Fluid Dynamics. Trends and Applications*, Proceedings of the Parallel CFD 2000 Conference, Trondheim, Norway, May 22–25, Elsevier, Amsterdam, pp. 275–282.
3. Averbuch, A., Israeli, M., and Vozovoi, L. (1997). On fast direct elliptic solver by modified Fourier method. *Numerical Algorithms* **15**, 287–313.
4. Averbuch, A., Israeli, M., and Vozovoi, L. (1998). A fast Poisson solver of arbitrary order accuracy in rectangular regions. *SIAM J. Sci. Comput.* **19**, 933–952.
5. Braverman, E., Israeli, M., Averbuch, A., and Vozovoi, L. (1998). A fast 3-D Poisson solver of arbitrary order accuracy. *J. Comput. Phys.* **144**, 109–136.
6. Braverman, E., Israeli, M., and Averbuch, A. (1999). A fast spectral solver for 3-D Helmholtz equation. *SIAM J. Sci. Comput.* **20**, 2237–2260.
7. Gottlieb, D., and Orszag, S. A. (1977). *Numerical Analysis of Spectral Methods: Theory and Applications*, SIAM, Philadelphia.
8. Greengard, L., and Lee, J.-Y. (1996). A direct adaptive Poisson solver of arbitrary order accuracy. *J. Comput. Phys. 125*, 415–424.
9. Greengard, L., and Rokhlin, V. (1987). A fast algorithm for particle simulations. *J. Comput. Phys.* **73**, 325–348.
10. Karniadakis, E. G., Israeli, M., and Orszag, S. A. (1991). High-Order splitting methods for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **97**, 414–443.
11. McKenney, A., Greengard, L., and Mayo, A. (1995). A fast Poisson solver for complex geometries. *J. Comput. Phys.* **118**, 348–355.