The Fast 3D Immersed Interface Method for Poisson Equations on Irregular Domains

Elgaddafi Elamami, PhD
Southern Arkansas University
278 Old McNeil Rd. Magnolia
AR, 71753, USA

Abstract

In this paper, Based on the IIM proposed by LeVeque and Li, [SIAM J. Numer. Anal., 31(1994), pp. 1019-1044] we will investigate and analyze some applications of the fast 3D solver, we developed embedding techniques to solve interior as well as exterior Poisson equations on complicated regions with Dirichlet or Neumann boundary conditions. The number of iterations in solving the Schur complement system appears to be independent of both the jump in the coefficient and the mesh size.

Keywords and phrases: immersed interface method, domain embedding technique, interface problem, jump conditions, fast, interior/exterior Poisson equation.

1. Introduction.

Sometimes a problem on an irregular domain can be handled more easily as an interface problem by embedding the domain into a cubic domain and then solving the equation on a Cartesian grid in the cube. The original boundary then becomes an interface. As an example, suppose we want to solve a three-dimensional elliptic equation on an irregular domain Ω. We can embed the domain in a larger cubic domain R. For example, we could solve the following Laplace equation with a Dirichlet boundary condition

$$\begin{aligned} &u_{xx} + u_{yy} + u_{zz} + ku = 0, && \text{in} \quad \Omega, \\ &u = v, && \text{on} \quad \partial\Omega. \end{aligned}$$

By extending it to the problem

$$\begin{aligned} &\Delta u = \iint F(\boldsymbol{X})\delta(x - X)\delta(y - Y)\delta(z - Z)dS, && \text{in} \quad R, \qquad (1.1) \\ &\mathrm{u} = 0, && \text{on} \quad \partial\mathrm{R}. \end{aligned}$$

The problem is then to determine F(**X**) so that the boundary condition $u = v \;\; on \;\; \partial\Omega$ is satisfied.

The solution is still continuous on the enlarged region R, but not smooth across the interface $\partial\Omega$.

This particular problem has been extensively studied in the past and a number of domain embedding procedures have been developed, e.g., capacitance methods [3,8,17,18] and methods based on solving integral equations along $\partial\Omega$. With the idea of the augmented approach, we can also develop an embedding technique to solve elliptic equations on complicated regions with Dirichlet, Neumann, or Robin boundary conditions. In this paper, we show how the augmented approach can be utilized to solve exterior / interior Poisson equations on irregular domains.

First, consider 3D interface Poisson equation with Dirichlet boundary condition

$$\begin{aligned} &\Delta u + ku = f, && \text{in} \quad \Omega, \qquad (1.2) \\ &u = v, && \text{on} \quad \partial\Omega. \end{aligned}$$

Where Ω is a cubic volume with an arbitrary closed void region, $\partial\Omega$ is the interior boundary of Ω and ∂R is the exterior boundary of Ω, see Figure (1.1) below for an illustration.

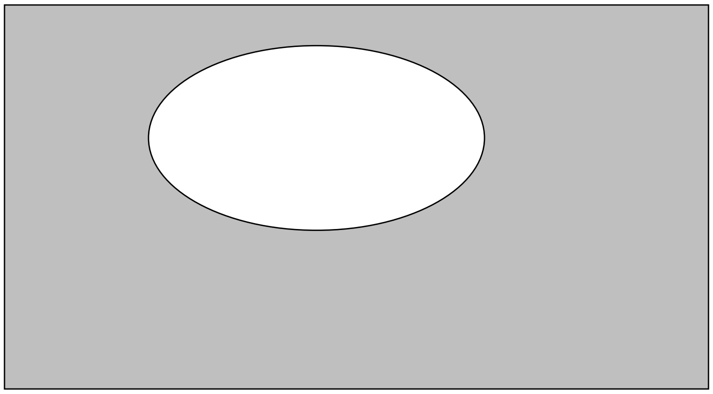

Figure 1.1 :A diagram for exterior Poisson problem.

From the embedding or the fictitious techniques, we can imagine the irregular region as an embedded region into a cube R. We may treat the original problem as an interface problem and use the IIM to solve it. However, recall that to make the resulting interface problem well-posed, usually we need to know not only the jump in the solution [u], but also the jump in the normal derivative of the solution $[u_n]$, across the boundary (now it becomes the interface). We may simply use the Dirichlet boundary condition as the jump condition in the solution, but unfortunately there seems no way to know the exact jump in the normal derivative of the solution. Thus, it will be assumed to be an augmented variable.

Based on the same idea as used in the augmented approach, we can solve the above problem by choosing an initial guess on $[u_n]$, and then updating it until the original boundary condition is satisfied. Below we begin to describe this approach in more detail.

We extend the source term in the Poisson equation by zero outside Ω. On the irregular boundary $\partial\Omega$, we allow a finite jump $[u]$ in the solution itself. One particular choice is just to use the original boundary condition v as $[u]$ and let $u = 0$ on the boundary ∂R of the cube R. As for the jump in the normal derivative of the solution $[u_n]$, we may use an initial guess, say, $[u_n] = g$ (usually 0). This extension leads to the following interface problem

$$\Delta u + ku = \begin{cases} f, & if \quad (x,y,z) \in \Omega, \\ 0, & if \quad (x,y,z) \notin \Omega, \end{cases}$$

$$\begin{aligned} &[u] = v, && \text{on} \quad \partial\Omega, \\ &[u_n] = g, && \text{on} \quad \partial\Omega, \qquad (1.3) \\ &u = 0, && \text{on} \quad \partial\text{R}. \end{aligned}$$

We then use the GMRES iteration to update g until the original boundary condition is satisfied, i.e.,

$$u^- = v, \quad on \;\; \partial\Omega, \qquad (1.4)$$

where u^- is the limiting value of the solution on the boundary from the inside of $\partial\Omega$.

Now the augmented approach can be used and only some minor changes are needed. First, instead of using $[\beta u_n] = g$ as the convergence-checking rule, we use $u^- = v$. Therefore, instead of interpolating $u_n^+ \; and \; u_n^-$ with the knowledge of u_{ijk}'s, we need to find u^+ and u^-. The same least square approach can still be used here. For example, our interpolation formula for u^-can be written in the following:

$$u^- \approx \sum_{i,j,k} \gamma_{ijk} \, u_{ijk} - Q. \qquad (1.5)$$

The same idea also applies to interior Poisson problem with Dirichlet boundary conditions as follows

$$\begin{aligned} &\Delta u + ku = f, && \text{in} \quad \partial\Omega, \\ &\text{u} = \text{v}, && \text{on} \;\; \partial\Omega, \qquad (1.6) \\ &u = u_0, && \text{on}\, \partial\text{R}. \end{aligned}$$

where Ωis an arbitrary closed region in 3D space, see Figure (1.2) for an illustrations.

By extending the source term in the Poisson equation by zero outside Ωand forming interface conditions across $\partial\Omega$, we get the following interface problem

$$\Delta u + ku = \begin{cases} f, & if\ (x,y,z) \in \ \Omega \\ 0, & if\ (x,y,z) \notin \Omega \end{cases}$$

$$\begin{aligned} &[u] = v, && \text{on} \quad \partial\Omega, \\ &[u_n] = g, && \text{on} \quad \partial\Omega, \qquad (1.7) \\ &u = u_0, && \text{on} \quad \partial\text{R}. \end{aligned}$$

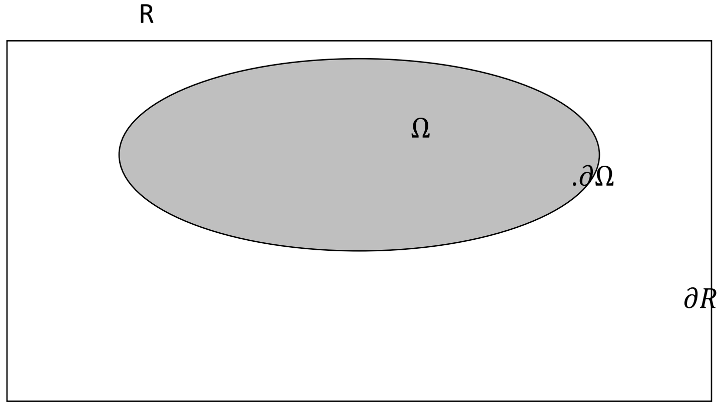

Figure 1.2: A diagram for interior Poisson problem.

Similarly, we then use the GMRES iteration to update g until the original interior boundary condition is satisfied, i.e.,

$$u^+ = v, \qquad \text{on} \quad \partial\Omega. \qquad (1.8)$$

where u^+ is the limiting value of the solution on the interior boundary from the outside of $\partial\Omega$, which can still be obtained by the same least square approach as in $[5]. i.e.$,

$$u^+ \approx \sum_{i,j,k} \gamma_{i,j,k} u_{i,j,k} - Q. \qquad (1.9)$$

We have tested the fictitious or embedding techniques by solving some exterior or interior Poisson equations with Dirichlet boundary conditions. As mentioned in [7] the computations are done by using **Dell Precision 690 Workstation** running **RHEL4, OS: RedHat Enterprise Linux, ws release 4 RHEL4, CPU: 1 XEON 5160, 2 cores (HT4 cores), memory 32GB.** and by using **gfortran** compiler. The tolerance for the GMRES is taken as 10^{-5}.

2. Numerical experiments

We will provide two examples in solving three-dimensional Poisson equations on exterior and interior irregular domains, respectively, to show the efficiency of the augmented approach. We want to show second order accuracy of thesolution, and moreimportantly also, show that the number of iterations is nearly independent of the mesh size in this chapter.

Example 2.1

In this example, the domain is the exterior of the sphere $x^2 + y^2 + z^2 = 1/4$. The differential equation is

$$u_{xx} + u_{yy} + u_{zz} = 20r^2 + \frac{\log(e)}{r^2}$$

Dirichlet boundary condition is chosen from the following exact solution and level set function:

$$u(x,y,z) = r^4 + \log(2r) + \frac{3}{16},$$

where$r = \sqrt{x^2 + y^2 + z^2}$. and on $\Gamma, r = r_0 = 1/2$

Below, Figure 2.1(outside of the sphere) shows a slice of the computed solution: $-u(x,y,0)$. The sphere is embedded into a unit cube [-1,1]×[-1,1]×[-1,1].

Again we did the simulations, but on a 104×104×104 grids. The mesh size is $h = 1/52$.

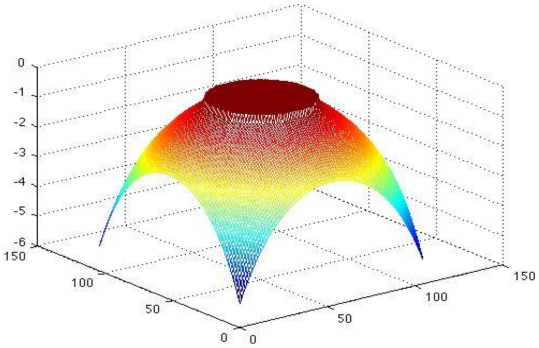

Figure 2.1: Plot of a slice of the computed solution $-u(x, y, 0)$ for example (2.1) with $l = m = n$ =104.

In Figure 2.1 (outside of the sphere) shows a slice of the computed solution :$-u(x, y, 0)$ on a 104×104×104 grids. The mesh size is $h = 1/52$. Both the solution and the flux $[\beta u_n]$ are discontinuous across the interface Γ. The source term f is discontinuous across the interface as well. The mesh size is $h = 1/52$. The interface is a sphere and the computational domain is a unit cube [-1,1]×[-1,1]×[-1,1]. The plot of the solution is composed of two pieces. We see that our method does accurately give the jumps in the solution and in the normal derivative of the solution, without smearing out the solution. The discontinuity in the solution and the flux is captured sharply by our numerical method.

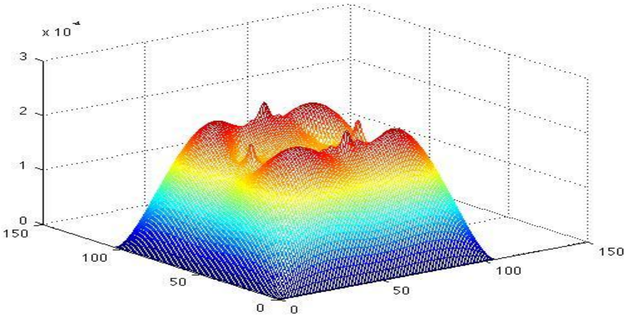

Figure 2.2 : Error plot of the slice of the computed solution for example (2.1)
with $l = m = n = 104$.

Figure 2.2 is a plot of the error in the infinity norm of the slice of the computed solution plotted in Figure 2.2 on a 104×104×104 grids. The mesh size is $h = 1/52$. The largest error in magnitude is about 1.2× 10^{-4}.The number of iterations for solving the Schur complement system using a GMRES is almost independent the mesh size h.

Example 2.2

The differential equation is

$$u_{xx} + u_{yy} + u_{zz} = f,$$

in the interior of the sphere $x^2 + y^2 + z^2 = 1/4$

with

$$f(x, y, z) = \begin{cases} -200r^2, & if\ r < 1/2, \\ 20r^2, & if\ r \geq\ 1/2. \end{cases}$$

Dirichlet boundary condition is chosen from the exact solution and level set function:

$$u(x, y, z) = -10r^4.$$

where $r = \sqrt{x^2 + y^2 + z^2}$and on Γ, $r = r_0 = 1/2$.

Again the domain is embedded into the unit cube [-1,1]×[-1,1]×[-1,1].

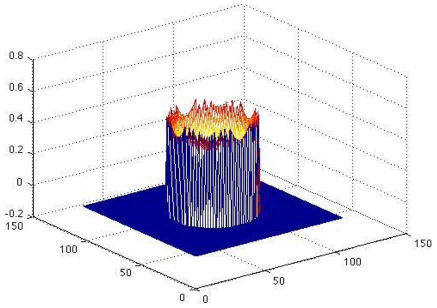

Figure 2.3: Plot of a slice of the computed solution $-u(x,y,0)$ for example (2.2) with $l=m=n=$104.

Figure 2.3 (inside on the top) shows a slice of the computed solution:$-u(x,y,0)$ on a 104×104×104 grids. The mesh size is $h=1/52$.Both the solution and the flux $[\beta u_n]$ are discontinuous across the interface Γ. The source term f is discontinuous across the interface as well. The mesh size is $h=1/52$. The interface is a sphere and the computational domain is a unit cube [-1,1]×[-1,1]×[-1,1]. The plot of the solution is composed of two pieces. We see that our method does accurately give the jumps in the solution and in the normal derivative of the solution, without smearing out the solution. The discontinuity in the solution and the flux is captured sharply by our numerical method.

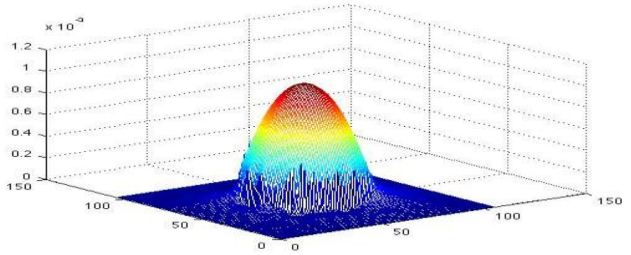

Figure 2.4: Error plot of the slice of the computed solution for example (2.2) with $l=m=n=104$.

Figure 2.4 is a plot of the error in the infinity norm of the slice of the computed solution on a 104×104×104 grids. The mesh size is $h=1/52$.The largest error in magnitude is about 0.5×10^{-3}.The number of iterations for solving the Schur complement system using a GMRES is almost independent the mesh size h.

3. **Grid refinement analysis**.

Tables (3.1)-(3.2) show the results of a grid refinement analysis, where $l=m=n$ is the number of uniform grid points in the $x, y,$ and z direction, respectively. The infinity norm forthe maximum relative error over all grid points and order of convergence are defined as in (3.1) and (3.2) respectively.

$$\|E_n\|_\infty = \frac{\max_{i,j,k}|u(x_i,y_j,z_k)-u_{ijk}|}{\max_{i,j,k}|u(x_i,y_j,z_k)|}, \qquad (3.1)$$

where$u_{i,j,k}$ is the computed approximation of $u(x_i,y_j,z_k)$. We also display the ratio of two successive errors and order of accuracy, respectively, as

$$\text{Ratio} = \|E_n\|/\|E_{2n}\|, \qquad \text{order} = \log(\|E_n\|/\|E_{2n}\|)/\log 2 \qquad (3.2)$$

For a first order method, the ratio approaches to 2, and for a second order method, the ratio approaches to 4. We will use the same notation for other examples in this thesis.

We see that an average ratio of 4 indicates that the augmented approach is a second order accuracy.

Show that the approach is second order accurate and we notice that the number of calls to the fast Poisson solver on the cubic domain is independent of the mesh size similar to the case of two space dimensions proposed in [13]although it may depend on the geometry of the domain. They also show the error in the infinity norm and other information. In those tables, N_{irreg} and N_{cntr} are the number of total irregular grid points and the number of control points respectively; N_{iter} is the number of iterations of the GMRES method or the number the calls of the 3D fast Poisson solver.

Table 3.1**: the grid refinement analysis of Example 2.1. Using Dell Precision Workstation 690**

n	N_{irreg}	N_{cntr}	CPU(s)	N_{iter}	$\|E_n\|_\infty$	Ratio(order)
26	920	506	2.1637	5	0.3898E-04	
52	3528	1820	8.6127	5	0.9849E-04	3.9577(1.98)
104	14048	7172	37.1104	5	0.2419E-04	4.0715(2.03)

Table 3.1 above shows the results of a grid refinement study with errors in the infinity norm and other information. We can see that the method still has second order accuracy when we use the embedding technique. We see that only a few iterations (only 5) are needed and the number is independent of the mesh size. The CPU time column (in seconds) shows that our method is very fast considering the very large condition number of the system of equations from a direct discretization even for a regular domain such as a cube. Furthermore, in this Table we can see that the number of iterations (only 5) is independent of the mesh size as in the case of two space dimensions. The CPU time does not increase much. Thus, our method here is fast in terms of number of GMRES iterations and in terms of CPU time.

Table 3.2**: the grid refinement analysis for Example 2.2. Using Dell Precision Workstation 690**

n	N_{irreg}	N_{cntr}	CPU(s)	N_{iter}	$\|E_n\|_\infty$	Ratio(order)
26	920	506	2.1577	5	0.2445E-01	
52	3528	1820	8.5897	5	0.6882E-02	3.9577(1.83)
104	14048	7172	36.9984	5	0.1744E-02	4.0715(1.98)

Table 3.2 above shows the results of a grid refinement study with errors in the infinity norm and other information. Again, we can see that the method still has second order accuracy when we use the embedding technique. Furthermore, in this Table we can see that the number of iterations (only 5) is independent of the mesh size as in the case of two space dimension. The CPU time does not increase much.

So far, in the discussion for a Poisson equation on an irregular domain, we form an interface problem that requires a known fixed jump in the solution and set an unknown jumpin the normal derivative of the solution. Then we iteratively update the jump in the normal derivative using the GMRES iteration to 10^{-5} tolerance until the original boundary condition is satisfied. Alternatively, we can set a known fixed jump in the normal derivative and an unknown jump in the solution. Then we use a similar GMRES iteration to update the jump in the solution until the original boundary condition is satisfied. For example, the following interior Holmholtz/Poisson equation with a Neumann boundary condition

$$\begin{aligned} &\Delta u + ku = f, && \text{in} \quad \Omega, (3.3) \\ &u_n = q, && \text{on} \quad \partial\Omega. \end{aligned}$$

It may be treated as the following interface problem

$$\Delta u = \begin{cases} f, & if\ \ (x,y,z) \in \Omega, \\ 0, & if\ (x,y,z) \notin \Omega, \end{cases}$$

$$\begin{aligned} &[u] = g, && \text{on} \quad \partial\Omega, (3.4) \\ &[u_n] = q, && \text{on} \quad \partial\Omega, \\ &u = 0. && \text{on} \quad \partial\mathrm{R}. \end{aligned}$$

Again, the solution u is a linear functional of g. We determine $g(s_1, s_2)$ such that the solution $u(g)$ satisfies the original boundary condition in (5.10) above. i.e., $u_n(g) = q$. This can be solved using the GMRES iteration exactly as we discussed earlier in Chapter 3.

4. *Conclusions*

In this paper, we showed that the fast solver can be applied to Holmholtz/Poisson problems on irregular domains which may have many other applications. The GMRES iteration is employed to solve the Schur complement system derived from the discretization. Numerical experiments showed that the fast algorithm was very successful and efficient.

We also have investigated some applications of the fast 3D solver. We developed embedding techniques to solve interior or exterior Poisson equations on complicated regions with Dirichlet or Neumann boundary conditions. The idea is to embed the irregular region into a cube to extend the Poisson equation to the entire cubic domain to introduce suitable jump conditions and to get an interface problem.

5. Acknowledgements.

It is my pleasure to acknowledge the encouragements and advice from various people including Professor Zhilin Li for his assistance, support, and encouragement to publish this paper. Also I would like to thank **the North Carolina Super-Computing Center (NCSC)** for letting me use the computing facilities there.

References

[1] L.M. Adams. A multigrid algorithm for immersed interface problems. In Proceedings of Copper Mountain Multigrid Conference, NASA Conference Publication 3339, pages 1-14,1995.

[2] J. Adams, P. Swarztrauber, and R. Sweet. Fishpack. Efficient Fortran sub-programs for the solution of separable elliptic partial differential equations. http://www.cisl.ucer.edu/css/software/fishpack; http://netlib.org/fishpack/.

[3] B.L. Buzbee, F.W. Dorr, J. A. George, and G.H. Golub. The direct solution of the discrete Poisson equation on irregular grids. SIAM J. *Numer. Anal., 8:722-736, 1971.*

[4] J.W. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1997.

[5] S. Deng. Immersed Interface Method for Three Dimensional Interface Problems and applications. J. *Comput. Phys., 184:215-243, 2003.*

[6] S. Deng, K. Ito, and Z. Li. Three dimensional elliptic solvers for interface problems and applications. J. *Comput. Phys., 184:215-243, 2003.*

[7]E**.** Elamami. An Augmented Approach for Solving 3D Elliptic Interface Problems. Aligrah Bulletin of Mathematics. AMU University. Aligrah. India. Volume 34, Numbers 1-2 (2015) 1-26. ISSN: 0304-9787.

[8] V. Faber, A. White, and R. Sweet. In extension of the implicit capacitance matrix algorithm or solving Poisson equations on irregular regions. In R. Vichnevetsky and R.S. Stepleman, editors, advances in Computer Methods for partial Differential Equations, pages 339-342. Pub. IMACS, 1981.

[9] R. L. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations, SIAM, 2007.

[10] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J. Numer. Anal., 31:1019-1044, 1994.

[11] Z. Li. The immersed interface Method- A Numerical Approach for Partial Differential Equations with Interfaces. Ph.D. thesis, University of Washington, 1994.

[12] Z. Li. A note on immersed interface methods for three-dimensional elliptic equations. Comput. Math. Appl., 31:9-17, 1996.

[13]Z. Li. A fast iterative algorithm for elliptic interface problems. SIAM J. Numer. Anal.,35:230-254, 1998.

[14] Z. Li. An overview of the immersed interface method and its applications. *Taiwanese J. Mathematics, 7:1-49, 2003.*

[15] Z. Li. IIMPACK, a collection of fortran codes for interface problems. Anonymous ftp at ftp.ncsu.edu /pub/math/zhilin/Package and http://www4.ncsu.edu/zhilin/IIM, last updated 2005.

[16] Z. Li and K. Ito, The immersed interface method, Numerical solutions of PDEs involving interfaces and irregular domains, SIAM, 2006.

[17] A. Mayo. The fast solution of Poisson's and the biharmonic equations on irregular regions. SIAM J. Num. Anal., 21:285-299, 1984.

[18] W. Proskurowski and O. Widlund. On the numerical solution of Helmholtz's equation by the capacitance matrix method. Math. Comp., 30:433-468, 1976.

[19] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving non symmetric linear systems. SIAM J. Sci. Statist. Comput., 7:856-869, 1986.