Adv Comput Math (2008) 29:113–133
DOI 10.1007/s10444-007-9043-6

A fast finite difference method for biharmonic equations on irregular domains and its application to an incompressible Stokes flow

Guo Chen · Zhilin Li · Ping Lin

Received: 3 September 2005 / Accepted: 22 December 2006 /
Published online: 24 July 2007

Abstract Biharmonic equations have many applications, especially in fluid and solid mechanics, but is difficult to solve due to the fourth order derivatives in the differential equation. In this paper a fast second order accurate algorithm based on a finite difference discretization and a Cartesian grid is developed for two dimensional biharmonic equations on irregular domains with essential boundary conditions. The irregular domain is embedded into a rectangular region and the biharmonic equation is decoupled to two Poisson equations. An auxiliary unknown quantity Δu along the boundary is introduced so that fast Poisson solvers on irregular domains can be used. Non-trivial numerical examples show the efficiency of the proposed method. The number of iterations of the method is independent of the mesh size. Another key to the method is a new interpolation scheme to evaluate the residual of the Schur complement system. The new biharmonic solver has been applied to solve the incompressible Stokes flow on an irregular domain.

Keywords Biharmonic equation · Irregular domain · Augmented method · Immersed interface method · Incompressible Stokes flow

Communicated by Z. Chen.

G. Chen · Z. Li (✉)
Department of Mathematics, North Carolina State University, Raleigh, NC 27695, USA
e-mail: zhilin@math.ncsu.edu

G. Chen
e-mail: gchen@unity.ncsu.edu

Z. Li
Center for Research in Scientific Computation,
North Carolina State University, Raleigh, NC 27695, USA

P. Lin
Department of Mathematics, National University of Singapore,
2 Science Drive 2, Kent Ridge, Singapore 117543
e-mail: matlinp@nus.edu.sg

Mathematics Subject Classifications (2000) 65N06 · 65N50

1 Introduction

In this paper, we consider a biharmonic equation defined on an irregular domain Ω

$$\begin{aligned} \Delta^2 u(x,y) &= f(x,y), \qquad (x,y) \in \Omega, \\ u(x,y) &= g_1(x,y), \qquad (x,y) \in \partial\Omega, \\ u_n(x,y) &= g_2(x,y), \qquad (x,y) \in \partial\Omega, \end{aligned} \tag{1.1}$$

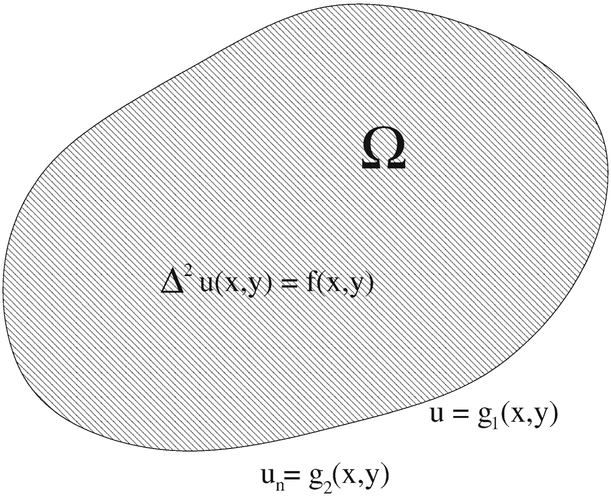
where

$$\Delta^2 \equiv \nabla^4 = \frac{\partial^4}{\partial x^4} + 2\frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4}, \tag{1.2}$$

Ω is a bounded open set in R^2 with a smooth boundary $\partial\Omega$, $u_n = \frac{\partial u}{\partial n}$ is the normal derivative of u on $\partial\Omega$, and n is the unit normal derivative pointing outward, see Fig. 1 for an illustration.

Biharmonic equations arise in many applications. Classical examples can be found in elasticity, fluid mechanics, and many other areas. In fluid mechanics, the solution $u(x, y)$ of equation (1.1) can be used to describe the stream-function of an incompressible two-dimensional creeping flow (zero Reynolds number), see Section 5 and [6, 16] for example. In linear elasticity, $u(x, y)$ can be used to represent the airy stress function, see [35]. In the theory of thin plates, equation (1.1) can be used to represent a "clamped plate" where f is the external load, and the solution $u(x, y)$ is the vertical displacement.

Fig. 1 A diagram of the problem: a biharmonic equation on an irregular domain



If the second boundary condition in (1.1) is replaced by $\Delta u|_{\partial\Omega} = g_2(x, y)$, then the biharmonic equation can be decoupled as two Poisson equations with a Dirichlet boundary condition on the same irregular domain

$$\begin{cases} \Delta v(x, y) = f(x, y), & (x, y) \in \Omega, \\ v(x, y)|_{\partial\Omega} = g_2(x, y), \end{cases} \tag{1.3}$$

$$\begin{cases} \Delta u(x, y) = v(x, y), & (x, y) \in \Omega, \\ u(x, y)|_{\partial\Omega} = g_1(x, y). \end{cases} \tag{1.4}$$

This observation is one of basis of our numerical method in which we set $\Delta u|_{\partial\Omega}$ as an intermediate variable. It is much more difficult to solve the problem when u_n is prescribed along $\partial\Omega$.

Various numerical methods have been developed for biharmonic equations in the literature when u and u_n are prescribed along $\partial\Omega$. One approach is to use a body fitted mesh with a finite element discretization. The discrete system then can be solved using a multigrid method, see [2, 5, 12, 34] for example. There is not much difference between regular or irregular domains in finite element methods except for the cost in the mesh generation and the condition number of the discrete system of equations.

There are limited publications on finite difference methods for biharmonic equations on irregular domains, even fewer with convincing numerical examples. Most of the finite difference methods appeared in the literature are for biharmonic equations on regular (rectangular and circular) domains. For certain domains, a conforming mapping can be used to solve the biharmonic equations defined on the domains [3]. Among a few finite difference methods for biharmonic equations on irregular domains, the remarkable ones are the fast algorithms based on integral equations and/or the fast multipole method [9, 23, 24]. These methods are most effective for homogeneous source term ($f(x, y) = 0$) and the boundary conditions ($u|_{\partial\Omega} = g_1$ and $u_n|_{\partial\Omega} = g_2$ in (1.1)) are replaced by the values of u_x and u_y along the boundary. These methods probably still can be applied with some extra efforts for non-homogeneous source terms and the essential boundary condition in (1.1). The implementations of these methods, especially when they are coupled with the fast multipole method, however, are not trivial.

In this paper, we propose a finite difference method for biharmonic equations based on the fast Poisson solver on irregular domains [19], the embedding technique, and an augmented approach for the decoupled two Poisson equations. The irregular domain is embedded in a rectangle and the biharmonic equation is augmented with an intermediate unknown Δu along the boundary $\partial\Omega$ which is a one-dimensional quantity. We use the GMRES iterative method to solve the discrete unknown $\Delta u|_{\partial\Omega}$. Each iteration involves solving two Poisson equations on the same irregular domain, which we use the available package of the immersed interface method, and a specially designed interpolation scheme to evaluate the residual. The irregular boundary is expressed in terms of a level set function. Non-trivial examples show that the method has second order accuracy in the infinity norm. The number of iterations is a small constant independent of the mesh size. The proposed method works for other type boundary conditions and three dimensional biharmonic equations on irregular domains.

1.1 A brief review of related finite difference methods

Under the assumption that $u(x, y)$ is a classical solution of the biharmonic problems (i.e., $u \in C^4(\Omega) \bigcap C^1(\bar{\Omega})$ and u has piecewise continuous second order derivatives on $\partial\Omega$). A popular technique is the so called coupled equation approach,

$$\begin{cases} \Delta u(x, y) = v(x, y), & (x, y) \in \Omega, \\ u(x, y)|_{\partial\Omega} = g_1(x, y), \end{cases} \tag{1.5}$$

$$\begin{cases} \Delta v(x, y) = f(x, y), & (x, y) \in \Omega, \\ v(x, y)|_{\partial\Omega} = \Delta u(x, y)|_{\partial\Omega} - c\left(u_n|_{\partial\Omega} - g_2(x, y)\right), \end{cases} \tag{1.6}$$

where c is a constant, see [7, 25, 30, 31] and others. If we give an initial guess $v(x, y)$, then an iteration can be generated until $u(x, y)$ converges. In [25], the coupling constant c is taken such that $0 < c < 2\nu_1$, where ν_1 is the smallest eigenvalue of the Dirichlet eigenvalue problem. When Ω is a rectangular domain, this formulation can lead to an iterative scheme which converges for all sufficiently small values of c. In this case, the two Poisson equations can be solved by a fast Poisson solver, for example, the Fishpack on rectangular domains [33]. However, for an irregular domain, it is also challenging to solve Poisson equations efficiently.

One can also try to discretize (1.1) on a uniform grid directly. The classical 13-point stencil for the biharmonic operator is most easily derived by applying the standard 5-point Laplace's operator twice

$$\begin{aligned} \Delta_{13}^2 u_{i,j} = \; & L_5(L_5 u_{i,j}) = \frac{2}{h^4}(20u_{i,j} - 8(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) \\ & + 2(u_{i+1,j+1} + u_{i-1,j+1} + u_{i-1,j-1} + u_{i+1,j-1}) \\ & + (u_{i+2,j} + u_{i-2,j} + u_{i,j+2} + u_{i,j-2})). \end{aligned} \tag{1.7}$$

The local truncation error is order of h^2. The finite difference approximation above needs to be modified at grid points near the boundary. One popular choices is the quadratic extrapolation where the normal derivative boundary condition at the grid points near the boundary are used to extrapolate the 'missing' (exterior) point in the 13 point stencil. This results in a stencil of the form:

$$\begin{aligned} \Delta^2 u_{i,j} = \; & \frac{2}{h^4}(21u_{i,j} - 8(u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}) \\ & + 2(u_{i+1,j+1} + u_{i-1,j+1} + u_{i-1,j-1} + u_{i+1,j-1}) \\ & + (u_{i+2,j} + u_{i,j+2} + u_{i,j-2})) + 2hu_n(x_{i-1}, y_j), \end{aligned} \tag{1.8}$$

when the irregular point u_{ij} is adjacent to the left boundary.

Glowinski and Pironneau [8] made the observation that the 13-point finite difference scheme combined with a quadratic extrapolation formula near the boundary is equivalent to solving the biharmonic equation using a mixed finite element method with piecewise linear elements.

Many similar modifications are discussed in [11]. Proper treatment of the points near the boundary remains a challenging problem with these schemes since inaccurate boundary approximations may affect the accuracy, but too complicated boundary approximations may destroy the matrix structure.

There are other alternative finite difference approximations for the biharmonic operator. Certain second and fourth-order finite difference approximations for the biharmonic equation (1.1) on a 9-point compact stencil are given in [32]. The approach there involves discretizing the biharmonic equation (1.1) using not only the grid values of the unknown solution $u(x, y)$ but also the values of the gradients $u_x(x, y)$ and $u_y(x, y)$ at selected grid points.

The standard iterative methods suffer from slow convergence when used to solve the system of finite difference equations for biharmonic equations, see for example, [10]. Direct solvers can only be used for relatively coarse grids. Bjørstad [27] had introduced a new iterative method that requires only $O(N^2)$ arithmetic operations on a rectangular domain, where N is the number of grid lines in one coordinate direction.

Due to the difficulties in handling the finite difference approximation close to a curved boundary, all the finite difference methods discussed above are restricted to rectangular domains. It is the purpose of this paper to provide an efficient finite difference method for biharmonic equations on irregular domains. An advantage using a Cartesian grid instead of a body fitted grid is that there is almost no cost in the grid generation which is significant for free boundary and moving interface problems that involve solving a biharmonic equation.

The paper is organized as follows. In Section 2, we introduce the main algorithm and the fast Poisson solver for irregular domains. The interpolation scheme to evaluate the residual of the Schur complement system is explained in Section 3. Numerical examples with grid refinement and efficiency analysis are presented in Section 4. The application to the incompressible Stokes flow is presented in Section 5.

2 The numerical method based on an augmented approach

Consider the solution of the following problem

$$\begin{cases} \Delta v(x, y) = f(x, y), & (x, y) \in \Omega \\ v(x, y)|_{\partial\Omega} = g(x, y), \end{cases}$$
$$\begin{cases} \Delta u(x, y) = v(x, y), & (x, y) \in \Omega \\ u(x, y)|_{\partial\Omega} = g_1(x, y). \end{cases} \tag{2.9}$$

The solution apparently is a functional of $g(x, y)$ which is defined only along the boundary $\partial\Omega$. We denote the solution as $u_g(x, y)$ to express the dependency of the solution on $g(x, y)$.

Let the solution of the original problem (1.1) be $u^*(x, y)$, and define

$$g^*(x, y) = \Delta u^*(x, y), \qquad (x, y) \in \partial\Omega. \tag{2.10}$$

Then $u^*(x, y)$ satisfies the second Poisson equation in (1.1) with $g(x, y) \equiv g^*(x, y)$. In other words, $u_{g^*}(x, y) \equiv u^*(x, y)$ and

$$\left.\frac{\partial u^*(x, y)}{\partial n}\right|_{\partial\Omega} = g_2(x, y) \tag{2.11}$$

is satisfied. Therefore, solving the original problem (1.1) is equivalent to finding the corresponding $g^*(x, y)$ and then $u_{g^*}(x, y)$ in (2.9). Notice that $g^*(x, y)$ is only defined along the boundary, so it is one dimensional lower than the solution $u(x, y)$.

We call $g(x, y)$ an augmented (intermediate) variable along the boundary. The auxiliary equation then is

$$\left.\frac{\partial u_g(x, y)}{\partial n}\right|_{\partial\Omega} = g_2(x, y). \tag{2.12}$$

Therefore the system is still closed since we have one more variable and one more equation.

The idea is to start with a guess $g^{(k)}(x, y)$ as an approximation to $g^*(x, y)$. Once we know $g^{(k)}(x, y)$, we can solve the two Poisson equations in (2.9) to get the solution $u_g^{(k)}(x, y)$. From the residual equation (2.12), we hope to get a better approximation $g^{(k+1)}(x, y)$. The process will become clearer after we discretize the system (2.9) and (2.12) in this section.

2.1 The computational frame

We embed the irregular domain into a rectangular domain $R : [a, b] \times [c, d] \supset \Omega$. The original boundary $\partial\Omega$ then becomes an interface within R. We solve the two Poisson equations in (2.9) on a uniform Cartesian grid

$$\begin{aligned} x_i &= a + ih, \qquad 0 \le i \le m, \\ y_j &= c + jh, \qquad 0 \le j \le n, \end{aligned} \tag{2.13}$$

where, for simplicity, we assume that $h = (b - a)/m = (d - c)/n$. The boundary $\partial\Omega$ is expressed as the zero level set of a two dimensional function $\varphi(x, y)$ defined on the entire rectangular region

$$\varphi(x, y) \begin{cases} > 0, & \text{if } (x, y) \in R - \Omega, \\ = 0, & \text{if } (x, y) \in \partial\Omega, \\ < 0, & \text{if } (x, y) \in \Omega. \end{cases} \tag{2.14}$$

There are a number of advantages using the level set method especially for complicated geometries and high dimensions. We refer the readers to [26, 29] for more information on the level set method.

The level set function is defined at all grid points by $\varphi_{ij} = \varphi(x_i, y_j)$. It is important that $\varphi(x, y)$ is a good approximation to the signed distance function at least in the neighborhood of the boundary $\partial\Omega$. If the level set function is not a good approximation to the signed distance function, then a re-initialization is needed to make it a good approximation to the signed distance function, see [13–15] for the re-initialization process.

Using the grid function φ_{ij}, all the grid points can be classified as regular (away from the boundary $\partial\Omega$) or irregular (close to or on the boundary $\partial\Omega$) grid points. Given a grid point (x_i, y_j), define

$$\begin{aligned} \varphi_{ij}^{\max} &= \max\{\varphi_{i-1,j}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}, \varphi_{i,j}\}, \\ \varphi_{ij}^{\min} &= \min\{\varphi_{i-1,j}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}, \varphi_{i,j}\}. \end{aligned} \tag{2.15}$$

A grid point (x_i, y_j) is irregular for our problem if

$$\varphi_{ij}^{\max}\varphi_{ij}^{\min} \le 0 \tag{2.16}$$

is true. Otherwise the grid point is regular.

2.2 The orthogonal projections of irregular grid points on the boundary

The augmented variable $g(x, y)$ and the augmented equation (2.12) are only defined along the boundary $\partial\Omega$. We need to discretize the augmented variable and the equation at certain points along the boundary. Those points are chosen as the orthogonal projection of the interior irregular grid points on the boundary, see Fig. 2 for an illustration.

Let $\mathbf{x}_{ij} = (x_i, y_j)$ be an interior irregular grid point which means $\varphi_{ij} \le 0$, and $\varphi_{ij}^{\max}\varphi_{ij}^{\min} \le 0$. The orthogonal projection of $\mathbf{x}_{ij}$ is approximated by the solution of the following quadratic equation:

$$\mathbf{X}^* = \mathbf{x} + \alpha\,\mathbf{p}, \qquad \mathbf{p} = \frac{\nabla\varphi}{|\nabla\varphi|}, \tag{2.17}$$

where α is determined from the quadratic equation below:

$$\varphi(\mathbf{x}) + |\nabla\varphi|\,\alpha + \frac{1}{2}\left(\mathbf{p}^T\,\mathrm{He}(\varphi)\,\mathbf{p}\right)\alpha^2 = 0, \tag{2.18}$$

where $\mathrm{He}(\varphi)$ is the Hessian matrix of the level set function $\varphi(x, y)$. All the quantities are defined at the interior grid point $\mathbf{x}_{ij} = (x_i, y_j)$ $(\varphi_{ij} \le 0)$ and are evaluated using the second order central finite difference scheme. The orthogonal projection computed using this procedure has third order accuracy.

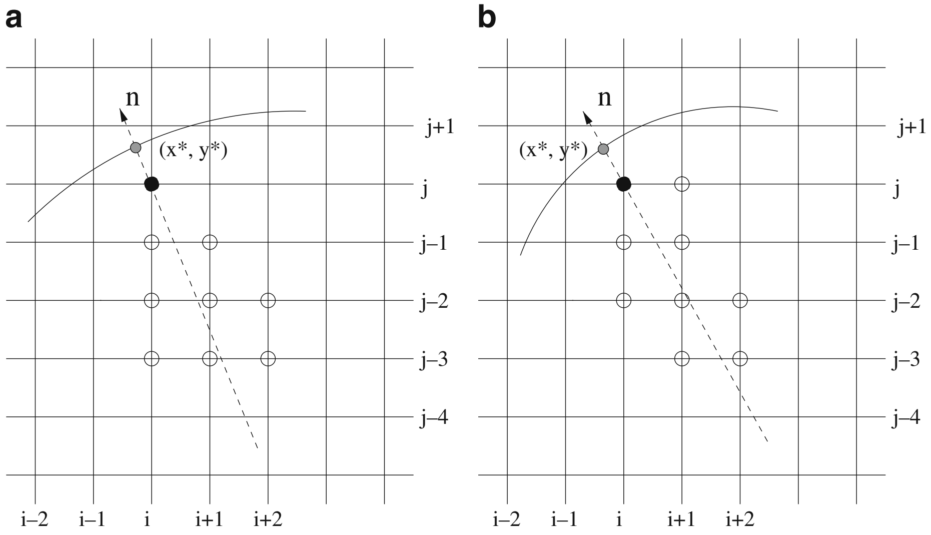


Fig. 2 Two illustrative examples of the orthogonal projection (x^*, y^*) of irregular grid points on the boundary, and the selected grid points for interpolation scheme at the projection to get $\frac{\partial u(\mathbf{X}^*)}{\partial n}$

We will denote these orthogonal projections of the interior irregular grid points by $\mathbf{X}_k = (x^*, y^*)$, $k = 1, 2, \cdots, N_b$, and will omit the dependency on the grid points for simplicity of the notation. These orthogonal projections are not ordered and there is no need to do so. The auxiliary variable $g(x, y)$ is discretely defined at $\mathbf{X}_k$. The augmented equation (2.12) is going to be discretized at $\mathbf{X}_k$ as well. We use the upper case letters such as U_{ij}, V_{ij}, G_k, for the discrete approximations at grid points and at those orthogonal projections on the boundary respectively.

2.3 The fast Poisson solver on irregular domains

The fast Poisson solver for irregular domains used to solve the two Poisson equations in (2.9) is based on the fast immersed interface method (IIM) developed in [18] and a modified version developed in [13, 14, 22]. The modification is needed because the original IIM in [17, 18] is designed for interface problems that are defined in the entire domain with discontinuities occur at the interface. The main idea of the fast Poisson solver on an irregular domain is to extend the Poisson equation to a rectangular domain $R \supset \Omega$. This procedure allows us to use fast Poisson solvers on a fixed Cartesian grid on the rectangular domain, for example, the Fishpack [33]. A short review on fast Poisson solvers on irregular domain can be found in [18].

The fast Poisson/Helmholtz solvers for interior/exterior problems with the boundary represented by the zero level set of a two dimensional function are available to the public [19]. We use the one designed for interior Helmholtz equations. The package also provides the orthogonal projections $\mathbf{X}_k$ of the interior irregular grid points, as well as the tangential and normal directions of the boundary $\partial\Omega$ at those projections.

2.4 The discrete system of equations in matrix vector form

Given a discrete approximation of $g(x, y)$ to the Laplacian $\Delta u|_{\partial\Omega}$ along the boundary at those orthogonal projections $\mathbf{X}_k$, we can use the fast Poisson solver mentioned above to solve the two Poisson equations in (2.9) to get $u_g(x, y)$. We denote the vector of the discretized values of U_{ij} (from the interior grid points) by $\mathbf{U}$; and the vector of the discretized values of $g(x, y)$ at the orthogonal projections of the interior irregular grid points by $\mathbf{G}$. The discrete from (2.9) can be written as

$$A\mathbf{U} + B\mathbf{G} = \mathbf{F}_1 \tag{2.19}$$

for some vector $\mathbf{F}_1$ and matrices A and B.[1] It requires solving two Poisson equations on the same irregular domain Ω with different Dirichlet boundary conditions to get the solution $\mathbf{U}$.

Once we know the solution $\mathbf{U}$ of the augmented system (2.9) given $\mathbf{G}$, we can interpolate U_{ij} linearly to get $\frac{\partial u}{\partial n}$ at those projections $\mathbf{X}_k$, $1 \le k \le N_b$. The

[1] Actually, the discrete form of the first equation in (2.9) can be written as $A_1\mathbf{V} = \mathbf{F} + E_1\mathbf{G}$ for some matrices A_1 and E_1; and the second equation can be written as $A_1\mathbf{U} = \mathbf{V} + E_1\mathbf{G}_1$. Therefore we have $A_1^2\mathbf{U} = \mathbf{F} + E_1\mathbf{G} + A_1E_1\mathbf{G}_1$. However, the matrices A_1, E_1, A, and B are used only for theoretical purposes and never formed explicitly. It would take more time and storage to compute these matrices than to solve the two Poisson equations on irregular domain directly.

interpolation scheme is crucial to the success of our algorithm and will be explained in detail in the next section. Therefore we can write

$$\begin{aligned}\left.\frac{\partial \mathbf{U}}{\partial n}\right|_{\partial\Omega} &= C\mathbf{U}\\ &= CA^{-1}\left(\mathbf{F}_1 - B\mathbf{G}\right) = CA^{-1}\mathbf{F}_1 - CA^{-1}B\mathbf{G},\end{aligned} \tag{2.20}$$

where C is a sparse matrix determined by the interpolation scheme. The matrices and vectors are only used for theoretical purposes but not actually constructed in our algorithm. We need to choose such a vector $\mathbf{G}$ that the second boundary condition $\frac{\partial u}{\partial n}|_{\partial\Omega} = g_2(x, y)$ is satisfied along the boundary $\partial\Omega$. Therefore we have the second matrix vector equation

$$\left.\frac{\partial \mathbf{U}}{\partial n}\right|_{\partial\Omega} = CA^{-1}\mathbf{F}_1 - CA^{-1}B\mathbf{G} = \mathbf{G}_2, \tag{2.21}$$

where $\mathbf{G}_2$ is the vector formed from the boundary condition $\frac{\partial u}{\partial n}|_{\partial\Omega} = g_2(x, y)$ at $\mathbf{X}_k$, $1 \le k \le N_b$. Rewrite the equation above as

$$E\mathbf{G} = \mathbf{G}_2 - CA^{-1}\mathbf{F}_1 = \mathbf{F}_2, \tag{2.22}$$

where $E = -CA^{-1}B$ and $\mathbf{F}_2 = \mathbf{G}_2 - CA^{-1}\mathbf{F}_1$. If we put the two matrix–vector equations (2.19) and (2.22) together we get

$$\begin{bmatrix} A & B \\ 0 & E \end{bmatrix}\begin{bmatrix} \mathbf{U} \\ \mathbf{G} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \end{bmatrix}. \tag{2.23}$$

Note that $\mathbf{G}$ is defined only on a set of points $\mathbf{X}_k$ while $\mathbf{U}$ is defined at all interior grid points. Let N_{in} be the total number of the grid points in the interior or on its boundary $\partial\Omega$, then we should have $O(N_b^2) \sim O(N_{in})$.

The linear system of equations (2.23) is the result of one step of block Gaussian elimination of the following linear system of equations

$$\begin{bmatrix} A & B \\ C & 0 \end{bmatrix}\begin{bmatrix} \mathbf{U} \\ \mathbf{G} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{G}_2 \end{bmatrix}, \tag{2.24}$$

which are the two the linear systems of (2.19) and (2.20) (the first line). The Schur complement linear system of equations above is

$$\left(0 - CA^{-1}B\right) G = \mathbf{G}_2 - CA^{-1}\mathbf{F}_1 \qquad \text{or} \quad E\mathbf{G} = \mathbf{F}_2. \tag{2.25}$$

If we can solve for the system above to get $\mathbf{G}$, then we can get $\mathbf{U}$ easily. Because the dimension of $\mathbf{G}$ is much smaller than $\mathbf{U}$, we expect to get a reasonably fast algorithm for the biharmonic equation on irregular domains if we can solve (2.25) efficiently. In implementation, we use the GMRES [28] to solve (2.25). The GMRES method only requires the matrix vector multiplication. We explain below why we do not need to form the matrix E explicitly.

First we set $\mathbf{G} = 0$ and solve the two Poisson equations in (2.9), or (2.19) in the discrete form, to get $\mathbf{U}(\mathbf{0})$ which is $A^{-1}\mathbf{F}_1$ from (2.19). Note that the residual of the Schur complement for $\mathbf{G} = 0$ is

$$\begin{aligned} R(\mathbf{0}) &= E\,\mathbf{0} - \mathbf{F}_2 = -\mathbf{G}_2 + CA^{-1}\mathbf{F}_1 \\ &= -\mathbf{G}_2 + C\mathbf{U}(\mathbf{0}) = -\mathbf{G}_2 + \left.\frac{\partial \mathbf{U}(\mathbf{0})}{\partial n}\right|_{\partial\Omega} \end{aligned} \tag{2.26}$$

which gives the right hand side of the Schur complement system with an opposite sign. The matrix–vector multiplication of the Schur complement system given $\mathbf{G}$ is obtained from the following two steps:

Step 1: Solve the two Poisson equations in (2.9), or (2.19) in the discrete form, to get $\mathbf{U}(\mathbf{G})$.

Step 2: Interpolate $\mathbf{U}(\mathbf{G})$ to get $\frac{\partial \mathbf{U}(\mathbf{G})}{\partial n}|_{\partial\Omega}$. Then the matrix vector multiplication is

$$E\mathbf{G} = \left.\frac{\partial \mathbf{U}(\mathbf{G})}{\partial n}\right|_{\partial\Omega} - \left.\frac{\partial \mathbf{U}(\mathbf{0})}{\partial n}\right|_{\partial\Omega}. \tag{2.27}$$

This is because

$$\begin{aligned} E\mathbf{G} &= -CA^{-1}B\mathbf{G} = -C\left(A^{-1}\mathbf{F}_1 - \mathbf{U}\right) \\ &= -C\mathbf{U}(\mathbf{0}) + C\mathbf{U}(\mathbf{G}) = \left.\frac{\partial \mathbf{U}(\mathbf{G})}{\partial n}\right|_{\partial\Omega} - \left.\frac{\partial \mathbf{U}(\mathbf{0})}{\partial n}\right|_{\partial\Omega} \end{aligned}$$

from the equalities $E = -CA^{-1}B$, $A\mathbf{U} + B\mathbf{G} = \mathbf{F}_1$, and $\mathbf{U}(\mathbf{0}) = A^{-1}\mathbf{F}_1$.

Now we can see that a matrix vector multiplication is equivalent to solving the two Poisson equations in (2.9), or (2.19) in the discrete form, to get $\mathbf{U}$, and using an interpolation scheme to get $\frac{\partial U}{\partial n}|_{\partial\Omega}$ at the orthogonal projections of the interior irregular grid points.

The computational complexity per GMRES step is two-calls to a fast Poisson solver and N_{in} calls to an interpolation scheme to compute the residual at inside irregular grid points near the boundary. In our implementation, we use the subroutine from Fishpack [1] which requires $O(N\log(N))$ operations, where N is the number of interior grid points of the embedded domain. The interpolation scheme requires $O(N_{in})$ operations which is negligible compared with $O(N\log(N))$.

We could form the matrix E by choosing $G = G_i = e_i, i = 1, 2, \cdots, e_{N_{in}}$, where e_i is the unit ith directional vector. Then we can use the Gaussian elimination method to solve the linear system of equations. The cost would be $O(N_{in}N\log(N))$ to form the matrix and $O(N_{in}^3/3)$ to solve the linear system of equations, where N_{in} is the number of points on the irregular boundary where the $\mathbf{G}$ is defined. The approach of generating the coefficient matrix is much more expensive than that using the GMRES method directly.

In the GMRES iteration, the initial guess is chosen as zero vector. The GMRES iterative method is insensitive to the initial guess.

The original idea of the augmented strategy for interface problems was proposed in [18] for elliptic interface problems with a piecewise but discontinuous coefficient. With a little modifications, the augmented developed in [18] was applied to generalized Helmholtz including Poisson equations on irregular domains in [14, 22]. The augmented approach for the incompressible Stokes equations with a discontinuous viscosity is presented in [20, 21].

There are at least two motivations to use augmented strategies. The first one is to get a faster algorithm compared to a direct discretization, particularly to take advantages of existing fast solvers. The second reason is that, for some interface problems, an augmented approach may be the only way to derive an accurate algorithm. This will be illustrated through the augmented immersed interface method for incompressible Stokes equations in which the jump conditions for the pressure and the velocity are coupled together. The augmented techniques enable us to decouple the jump conditions so that the idea of the immersed interface method can be applied.

Using augmented strategies, some augmented variable **g** of co-dimension one will be introduced. Once we know the augmented variable, it is relatively easy to solve the original problem. In discretization the approximate solution (denoted by **U**) to the original problem and the augmented variable **g** together form a large linear system. If we eliminate **U** from the matrix vector equations, we will get the Schur complement system for the augmented variable, which is generally much smaller than that for **U**. Therefore we can use the GMRES iterative method [28] to solve the Schur complement for the augmented variable. In implementation, there is no need to explicitly form the matrices. The matrix–vector multiplication needed for the GMRES iteration includes mainly two steps: (1) solving the original problem assuming the augmented variable is known; (2) finding the residual of the constraint using the computed approximate solution given the augmented variable. The constraint is often the jump condition or the boundary condition. In this chapter, we explain this technique for some interface problems and problems defined on irregular domains.

While an augmented approach for an interface or irregular domain problem has some similarities with an integral equation approach, or boundary element method, to find a source strength, the augmented methods described in this chapter have a few special features: (1) no Green function is needed; (2) no need to set up the system of equations; (3) applicable to general PDEs with or without source terms; (4) the process does not depend on the boundary condition. On the other hand, we may have less knowledge about the condition of the Schur complement system and how to apply pre-conditioning techniques. The efficiency of the algorithm depends on the number of iterations of the GMRES method.

3 The weighted least squares interpolation scheme

The interpolation scheme (2.20) is crucial to the efficiency (accuracy and the number of iterations of the GMRES) of the method. Since only the information inside the domain Ω is useful, the interpolation scheme at a point **X** on the boundary can be written as

$$\frac{\partial U(\mathbf{X})}{\partial n} = \sum_{i,j,\varphi(i,j)\le 0} \gamma_{ij}\, U_{ij}\, d_\alpha(|\mathbf{X}-\mathbf{x}_{ij}|), \tag{3.28}$$

where $d_\alpha(r)$ is a weighted distance function,

$$d_\alpha(r) = \alpha\delta_{\alpha/2}(r) = \begin{cases} \frac{1}{2}(1+\cos(\pi r/\alpha)) & \text{if } |r| < \alpha \\ 0 & \text{if } |r| \geq \alpha. \end{cases} \tag{3.29}$$

Note that $\frac{\partial U(\mathbf{X})}{\partial n}$ is one of components needed in the matrix vector multiplication $E\mathbf{G}$. We need to determine the parameter α and the interpolation coefficients γ_{ij}. Below we discuss how to determine the coefficients γ_{ij}. Actually, these coefficients are different from point to point on the boundary. So they should really be labeled as $\gamma_{ij,\mathbf{X}}$. But for simplicity of notation, we will concentrate on a single point $\mathbf{X} = (X, Y)$ and drop out the subscript $\mathbf{X}$.

Since it is the normal derivative that we are trying to interpolate, we use the local coordinates at the boundary point (X, Y),

$$\begin{aligned} \xi &= (x-X)\cos\theta + (y-Y)\sin\theta, \\ \eta &= -(x-X)\sin\theta + (y-Y)\cos\theta, \end{aligned} \tag{3.30}$$

where θ is the angle between the x-axis and the normal direction at the point (X, Y). Under such new coordinates, the interface can be parameterized by $\xi = \chi(\eta)$, $\eta = \eta$. Note that $\chi(0) = 0$, and, $\chi'(0) = 0$ as well, assuming that the boundary is smooth enough at (X, Y).

We use an un-determined coefficients method to determine the coefficients γ_{ij}. Let (ξ_i, η_j) be the ξ-η coordinates of (x_i, y_j), we have the following from the Taylor expansion:

$$u(x_i, y_j) = u(\xi_i, \eta_j) = u + u_\xi \xi_i + u_\eta \eta_j + \frac{1}{2}u_{\xi\xi}\xi_i^2 + \frac{1}{2}u_{\eta\eta}\eta_j^2 + u_{\xi\eta}\xi_i\eta_j + O(h^3), \tag{3.31}$$

where for simplicity, we use the same notation for u and its derivatives in the original and the local coordinates, $u, u_\xi, \cdots, u_{\xi\eta}$ are defined at (X, Y) in the original coordinates, or $(0, 0)$ in the local coordinates. Plugging in (3.31) into (3.28) and collecting terms, we have

$$\begin{aligned} \frac{\partial U(\mathbf{X})}{\partial n} &\approx \sum_{i,j,\varphi_{ij}\leq 0} \gamma_{ij}\, u(x_i, y_j)\, d_\alpha(|\vec{X} - \vec{x_{ij}}|) \\ &= a_1 u + a_2 u_\xi + a_3 u_\eta + a_4 u_{\xi\xi} + a_5 u_{\eta\eta} + a_6 u_{\xi\eta} + O(h^3 \max|\gamma_{ij}|) \end{aligned} \tag{3.32}$$

where the a_i's are given by

$$\begin{aligned} a_1 &= \sum_{i,j} \gamma_{ij} d_\alpha(|\mathbf{X} - \mathbf{x_{ij}}|) & a_4 &= \sum_{i,j} \frac{1}{2}\xi_k^2 \gamma_{ij} d_\alpha(|\mathbf{X} - \mathbf{x_{ij}}|) \\ a_2 &= \sum_{i,j} \xi_k \gamma_{ij} d_\alpha(|\mathbf{X} - \mathbf{x_{ij}}|) & a_5 &= \sum_{i,j} \frac{1}{2}\eta_k^2 \gamma_{ij} d_\alpha(|\mathbf{X} - \mathbf{x_{ij}}|) \\ a_3 &= \sum_{i,j} \eta_k \gamma_{ij} d_\alpha(|\mathbf{X} - \mathbf{x_{ij}}|) & a_6 &= \sum_{i,j} \xi_k \eta_k \gamma_{ij} d_\alpha(|\mathbf{X} - \mathbf{x_{ij}}|). \end{aligned} \tag{3.33}$$

From the local coordinate transformation, we have $u_n = u_\xi$, hence we can set up the linear system of equations for the coefficients γ_{ij} as

$$\begin{aligned} &a_1 = 0, \qquad a_2 = 1, \qquad a_3 = 0, \\ &a_4 = 0, \qquad a_5 = 0, \qquad a_6 = 0. \end{aligned} \tag{3.34}$$

If more than six different interior grid points are involved, then there is at least one solution. Usually we choose an neighborhood of $\mathbf{X}$ that contains more than six different interior grid points so that we have an under-determined system. We use the singular value decomposition (SVD) to find the least squares solution with the least l-2 norm. In this way, the coefficients γ_{ij} have roughly the same magnitude $O(1/h)$, and $\gamma_{ij}^* d_\alpha(|\mathbf{X} - \mathbf{x_{ij}}|)$ is roughly a decreasing function of the distance measured from $\mathbf{X}$. The least squares interpolation plays an important role in the stability of the algorithm. In practice, only a hand full of grid points, controlled by the parameter α and the normal direction at the boundary point (X, Y), are involved. Those grid points which are closer to (X, Y) have more influence than those which are further away.

The only trade-off of our weighted least square interpolation is that we have to solve an extra under-determined system of linear equations. The larger α is, the more computational cost in solving (3.34). However, the size of the linear system is small and the coefficients can be pre-determined before the GMRES iteration. We will see that the extra cost in dealing with the boundary is only a fraction of the total CPU time compared to the Poisson solvers. We also tried the third order interpolation scheme (10 equations) and the numerical results are similar.

3.1 Selecting grid points for the interpolation scheme

If the interpolation points are chosen radially from the center of the interpolation point $\mathbf{X}$ until enough interior grid points ($6 \sim 9$ for second order schemes, $10 \sim 15$ for third order scheme) are included, the method works fine if the curvature at $\mathbf{X}$ is not too large. However, if the curvature is large and there are fewer grid points in a thin tube of the normal direction compared with that of the tangential direction, then we may either need a large circle to include more interior points in the tube, or we may not have a good accuracy for the interpolated normal derivative. Either case will affect the efficiency of the numerical algorithm.

Our new interpolation scheme is to select the grid points in a thin cone. The vortex of cone is the interpolation point $\mathbf{X}$ and the axis is the normal line passing though $\mathbf{X}$, see Fig. 2 for an illustration, and see [4] for more details. This approach works better because we need the directional derivative of u along the normal direction. The angle of the cone is a parameter such that $\tan\psi$ is between [1/10, 1/2] in our choice.

4 Numerical examples

We have done a number of numerical experiments which confirm the expected order of accuracy. All the computations are done using Sun Ultra 10 workstations. The fast Poisson solver on irregular domain used is from the IIM-pack [19], and the tolerance for the convergence is 10^{-6} for the 2-norm of the residual vector. In all the examples, the initial approximation is chosen as zero vector.

Example 4.1 In this example we consider a biharmonic equation defined on a circle $x^2 + y^2 = 1/4$ with the constructed exact solution

$$u(x, y) = x^2 + y^2 + e^x, \qquad (x, y) \in \Omega. \tag{4.35}$$

The forcing term $f(x, y)$ is obtained by applying the biharmonic operator to the exact solution $u(x, y)$

$$f(x, y) = e^x. \tag{4.36}$$

The normal derivative of the solution on the boundary is

$$u_n(x, y) = 8x^2 + 4xe^x + 8y^2, \qquad (x, y) \in \partial\Omega, \tag{4.37}$$

where $\partial\Omega$ is the boundary of the circle. The computation domain is chosen as the square $[-1, 1] \times [-1, 1]$.

Table 1 shows the results of a grid refinement study, where the first column is the number of uniform grid points in the x and y directions. The maximum error is defined over all the interior grid points,

$$\|E_n\|_\infty = \max_{i,j,\varphi(i,j)\leq 0} |u(x_i, y_j) - U_{ij}|, \tag{4.38}$$

where U_{ij} is the computed approximation at the interior grid points (x_i, y_j). The third column is the ratio of two consecutive errors defined as

$$r_1 = \frac{\|E_n\|_\infty}{\|E_{2n}\|_\infty}. \tag{4.39}$$

For a second order accurate method, the ratio should approach number four while the ratio should approach number two for a first order method. We can see clearly second order accuracy of our method. The fourth column is the number of iterations. We see that only a few iterations are needed and the number is independent of the mesh size. The fifth column is the CPU time which show the 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 rectangles. We will use the same notations for the rest of examples in this section.

To better understand the nature of the Schur complement system, we have generated the coefficient matrix **G** and found the condition numbers. The condition numbers are 170.3239, 5897.5, and 28509 for the mesh sizes 64 by 64, 128 by 128, and 256 by 256 respectively for Example 4.1. This is expected for the bi-harmonic

Table 1 A grid refinement analysis for Example 4.1

Mesh size	$\|E_n\|_\infty$	r_1	No.	CPU(s)
64×64	3.0003×10^{-4}		4	0.25
128×128	8.0076×10^{-5}	3.74678	5	0.7709
256×256	1.9710×10^{-5}	4.06270	5	2.3530
512×512	4.7654×10^{-6}	4.13604	4	10.375

system. The cost is $O(N_{in}N\log(N))$ to form the matrix and $O(N_{in}^3/3)$ to solve the linear system of equations. This approach is much expensive than that using the GMRES method directly. Our method has fast convergence because the GMRES method generally converges fast at first few iterations and we just need to iterate to the level that is much smaller than the local truncation errors. The interpolation scheme using a cone is an important factor in fast convergence.

Example 4.2 In this example, the boundary is a skinny ellipse $\frac{x^2}{0.5^2}+\frac{y^2}{0.15^2}=1$. The differential equation is:

$$\Delta^2 u = \frac{24y}{(1+x)^5} - \frac{12x}{(1+y)^2} - \frac{6x^3}{(1+y)^4}. \tag{4.40}$$

We use the Dirichlet boundary condition which is determined from the exact solution

$$u(x,y) = x^3\ln(1+y) + \frac{y}{1+x}, \qquad (x,y)\in\Omega, \tag{4.41}$$

and $u_n(x,y)$, the normal derivative of $u(x,y)$ is also computed from the exact solution. Note that the level set function is used to find the normal and tangential directions. The computation domain is chosen as $[-0.6, 0.6]\times[-0.3, 0.3]$. In other words, we can use a rectangle that is just large enough to close the irregular domain.

Table 2 shows the results of a grid refinement study. We can see that the method still has average second order accuracy, requires only a few iterations, and it is very fast in terms of the CPU time. For this example, the curvature is quite large at two ends of the major axes of the ellipse. The new interpolation scheme is crucial to the algorithm. For irregular domain problems, the errors of the computed solution using embedding techniques do not decline monotonically but rather fluctuate. This is because the relative position of the irregular boundary and the underlying Cartesian grid does not change in a fixed pattern. In other words, while the error $C(h,\partial\Omega)h^2$ converges quadratically, the error constant $C(h,\partial\Omega)$ is not a constant, see [18]. Nevertheless, the error constant $C(h,\partial\Omega)$ is bounded. It is more reasonable to do the linear regression analysis to find the convergence order which is shown in Fig. 3. The average order of convergence is 2.0240.

Example 4.3 In this example we consider a biharmonic equation defined on a non-convex, non-concave region defined by

$$\begin{cases} x = (0.6+0.25\sin(5\theta))\cos(\theta), \\ y = (0.6+0.25\sin(5\theta))\sin(\theta), \end{cases} \qquad \theta\in[0,2\pi). \tag{4.42}$$

Table 2 A grid refinement analysis for Example 4.2

Mesh size	$\|E_n\|_\infty$	r_1	No.	CPU(s)
64 × 32	3.6576×10^{-4}		8	0.7310
128 × 64	9.5430×10^{-5}	3.8327	4	1.0810
256 × 128	2.0889×10^{-5}	4.5684	5	4.3360
512 × 256	4.9899×10^{-6}	4.1862	6	21.000

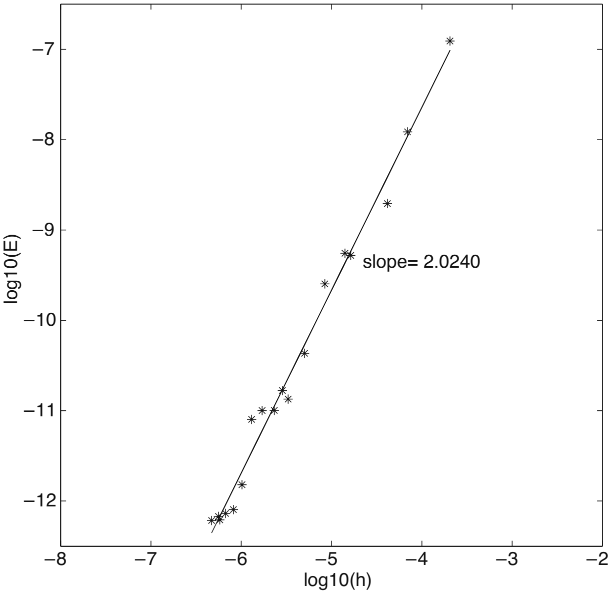


Fig. 3 The linear regression analysis on the errors for Example 4.2

The differential equation is:

$$\Delta^2 u = 0. \tag{4.43}$$

The essential boundary condition $u|_{\partial\Omega}$ and $u_n|_{\partial\Omega}$ are determined from the constructed exact solution

$$u(x, y) = x^2 + y^2 + e^x \cos(y), \tag{4.44}$$

in connection with the level set function.

Table 3 shows the results of a grid refinement study. Figure 4 shows the mesh plot of the exact solution. Figure 5 shows the linear regression analysis. The average convergence order is 1.9300.

4.1 Algorithm efficiency analysis

We have seen from previous examples that the number of iterations are fairly small. For most of the examples, it is between 4 $\sim$ 10. Another concern about the algorithm is how much overhead cost is needed for dealing with the interior irregular points.

Table 3 A grid refinement analysis for Example 4.3

Mesh size	$\|E_n\|_\infty$	r_1	No.	CPU(s)
64 × 64	8.14721×10^{-4}		9	2.2130
128 × 128	2.08149×10^{-4}	3.91412	7	3.7360
256 × 256	8.06897×10^{-5}	2.57962	9	13.520
512 × 512	1.83151×10^{-5}	4.40563	9	52.034

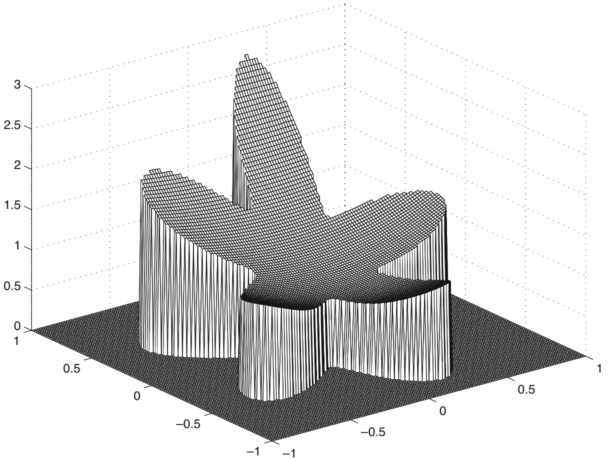


Fig. 4 The computed solution for Example 4.3. The difference between the exact and computed solutions is hardly visible

The cost includes indexing the irregular grid points, finding the projections, and solving a linear system of equations to find the coefficients of the interpolation scheme at each orthogonal projection on the boundary of the interior irregular points. Certainly the CPU time depends on the geometry. In Table 4, we show the CPU time spent in dealing with the boundary and that in solving the entire problem. We can see that the overhead time is only a fraction of the total CPU time especially as the mesh gets finer.

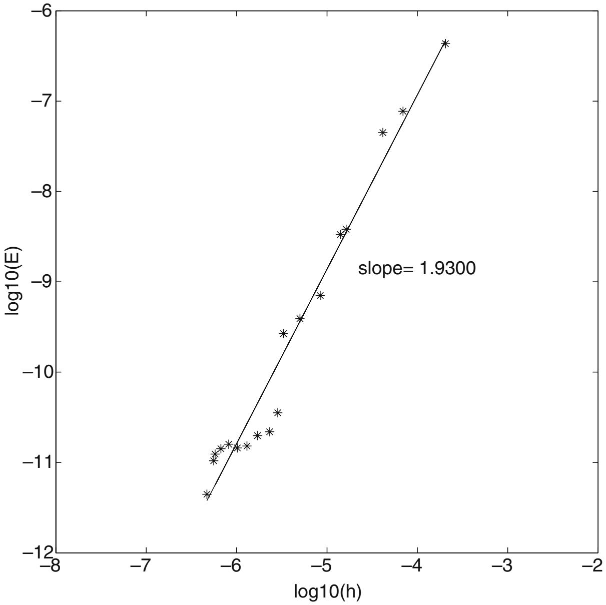


Fig. 5 The linear regression analysis on the errors for Example 4.3

Table 4 The CPU time in dealing with interior irregular grid points (t_{ov}) and the CPU time for the linear solver (t_{solve}) for Example 4.1 and Example 4.3

Mesh size	Example 4.1		Example 4.3	
	t_{ov}	t_{solve}	t_{ov}	t_{solve}
64 × 64	0.0499	0.2510	0.0699	2.2130
128 × 128	0.1100	0.7749	0.2710	3.7360
256 × 256	0.1210	2.3530	0.5410	13.520
512 × 512	0.3999	10.3750	0.6610	52.035

5 An application of the biharmonic solver to the incompressible Stokes flow on an irregular domain

Consider the incompressible Stokes flow in two dimensional spaces

$$\begin{aligned} \mu\Delta u &= p_x - F_1, \qquad \mathbf{x}\in\Omega, \\ \mu\Delta v &= p_y - F_2, \qquad \mathbf{x}\in\Omega, \\ \nabla\cdot\mathbf{u} &= 0, \end{aligned} \tag{5.45}$$

where μ is the fluid viscosity, p is the pressure, $\mathbf{u}=(u,v)$ is the velocity, and $\mathbf{F}=(F_1,F_2)$ is a forcing term. Equations (5.45) are supplemented by the "no-slip" boundary condition

$$\mathbf{u}(\mathbf{x})|_{\partial\Omega} = 0. \tag{5.46}$$

The vorticity function, which is a scalar in two-dimensional case, is defined by

$$\omega(\mathbf{x}) = (\nabla\times\mathbf{u})\cdot\mathbf{k} = -u_y + v_x, \qquad \mathbf{x}\in\Omega, \tag{5.47}$$

where $\mathbf{k}$ is the unit vector in the z-direction. The first two equation of (5.45) can be written in vector form

$$\mu\Delta\mathbf{u} = \nabla p - \mathbf{F}, \qquad \mathbf{x}\in\Omega. \tag{5.48}$$

Taking curl of the above equation we get

$$-\mu\Delta\omega = (\nabla\times\mathbf{F})\cdot\mathbf{k}, \qquad \mathbf{x}\in\Omega. \tag{5.49}$$

The velocity field $\mathbf{u}$ is obtained by using (5.47) in conjunction with (5.46). This can be done via the stream-function formulation as follows. Due to $\nabla\cdot\mathbf{u}=0$, there is a scalar function $\psi(\mathbf{x})$ such that

$$\mathbf{u}(\mathbf{x}) = \nabla^{\perp}\psi = \left(-\frac{\partial\psi}{\partial y}, \frac{\partial\psi}{\partial x}\right), \qquad \mathbf{x}\in\Omega. \tag{5.50}$$

Therefore from (5.47), we get

$$\Delta\psi = \omega. \tag{5.51}$$

Thus, equations (5.49) and (5.51), along with the relation (5.50), serve as the vorticity stream-function formulation of the problem.

We note that the “no-slip” condition (5.46) and the relation (5.50) implies

$$\psi(\mathbf{x})|_{\partial\Omega} = \left.\frac{\partial}{\partial \mathbf{n}}\psi(\mathbf{x})\right|_{\partial\Omega} = 0, \tag{5.52}$$

where $\mathbf{n}$ is the unit normal vector of the boundary $\partial\Omega$ pointing outward. Note that (5.52) follows since ψ is a constant along the boundary and is only determined up to an additive constant.

Plugging in (5.51) into (5.49), and using the relation (5.52), we get

$$\begin{aligned} -\mu\Delta^2\psi &= (\nabla\times\mathbf{F})\cdot\mathbf{k}, \qquad \mathbf{x}\in\Omega,\\ \psi(\mathbf{x})|_{\partial\Omega} &= 0,\\ \psi_{\mathbf{n}}(\mathbf{x})|_{\partial\Omega} &= 0. \end{aligned} \tag{5.53}$$

This is a well defined biharmonic equation with essential boundary condition. The advantage of using the formulation to solve for the velocity field $\mathbf{u}$ is that it eliminates the difficulty of dealing with the improper partition of boundary conditions of the vorticity-stream function formulation. As we stated earlier, we assume that the boundary of the domain $\partial\Omega$ is piecewise smooth.

We use the algorithm developed in previous sections to solve (5.53) numerically. Once ψ is obtained, the standard central difference formula can be used to interpolate (5.50) to get u and v at regular grid points. For irregular interior grid points, we use a similar least squares interpolation technique developed in Section 3 to recover u and v.

We have tested our algorithm for the Stokes flow with Reynolds number being 400. The stream function is

$$\psi(x,y) = \frac{1}{\pi}\sin^2\left(\frac{x^2+y^2}{4}\pi\right) - \frac{1}{\pi}. \tag{5.54}$$

The fluid is confined in a circular region

$$\Omega = \left\{(x,y)|\ x^2+y^2 = 2\right\} \tag{5.55}$$

and the computation region is chosen to be $[-1.6, 1.6]\times[-1.6, 1.6]$. The no-slip boundary condition

$$\begin{aligned} \psi(x,y) &= 0, \qquad (x,y)\in\partial\Omega,\\ \psi_n(x,y) &= 0, \qquad (x,y)\in\partial\Omega. \end{aligned} \tag{5.56}$$

are satisfied.

Table 5 A grid refinement analysis of $\mathbf{u} = (u, v)$ for examples 5.10–5.12

Mesh size	$\|E(u)_n\|_\infty$	r	$\|E(v)_n\|_\infty$	r
64×64	1.07739×10^{-2}		1.08006×10^{-2}	
128×128	2.54015×10^{-3}	4.2415	2.56114×10^{-3}	4.2171
256×256	4.49293×10^{-4}	5.6570	4.49161×10^{-4}	5.7037
512×512	1.09871×10^{-4}	4.0892	1.09842×10^{-4}	4.0907

In Table 5, we show the results of a grid refinement study for the velocity field u and v. We can see clearly second order accuracy for the solution (u, v) in the infinity norm. We also tested other examples and got similar results.

6 Summary

In this paper, a fast iterative method has been developed for biharmonic equations defined on irregular domains with essential boundary conditions. The method takes advantage of the fast Poisson solver on irregular domains available to the public and uses an augmented approach so that the biharmonic equation is decomposed to two Poisson equations. The GMRES method is used to solve the augmented variable which is the Laplacian of the solution along the boundary. An interpolation scheme for approximating the normal derivative of the solution is crucial to the success of the algorithm. Numerical examples show the efficiency of the second order accurate method. Only a few iterations are needed to solve the augmented variable and the entire problem. An application of the fast biharmonic solver to the incompressible Stokes flow is also presented.

Acknowledgements Guo Chen and Zhilin Li are supported in part by USA grants NSF-NIH 0201094, NFS-DMS-0412654, ARO 43751-MA and 49308-MA, and AFSOR-FA9550-06-1-0241. Ping Lin is supported in part by Singapore ARF grant R-146-000-033-112, R-146-000-053-112.

References

1. Adams, J., Swarztrauber, P., Sweet, R.: Fishpack: efficient Fortran subprograms for the solution of separable elliptic partial differential equations. http://www.netlib.org/fishpack/
2. Brenner, S.C.: An optimal-order nonconforming multigrid method for the biharmonic equation. SIAM J. Numer. Anal. **26**, 1124–1138 (1989)
3. Chan, R.H., DeLillo, T.K., Horn, M.A.: The numerical solution of the biharmonic equation by conformal mapping. SIAM J. Sci. Comput. **18**(6), 1571–1582 (1997)
4. Chen, G.: Immersed interface method for biharmonic equations defined on irregular domains and its application to the Stokes flow, Ph.D. thesis, North Carolina State University (2003)
5. Davini, C., Pitacco, I.: An unconstrained mixed method for the biharmonic problem. SIAM J. Numer. Anal. **38**, 820–836 (2000)
6. W.E., Liu, J.: Vorticity boundary condition and related issues for finite difference schemes. J. Comput. Phys. **124**, 368–382 (1996)
7. Ehrlich, L.W.: Solving the biharmonic equation as coupled finite difference equations. SIAM J. Numer. Anal. **8** (1971)
8. Glowinski, R., Pironneau, O.: Numerical methods for the first biharmonic equation and for the two-dimensional stokes problem. SIAM Rev. **21**, 167–212 (1979)
9. Greenbaum, A., Greengard, L., Mayo, A.: On the numerical-solution of the biharmonic equation in the plane. Phy. D **60**, 216–225 (1992)
10. Gupta, M.M.: Spectrum transformation methods for divergent iteration. NASA Techinal Memorandum 103745 (ICOMP-91-02) (1991)
11. Gupta, M.M., Manohar, R.: Direct solution of biharmonic equation using noncoupled approach. J. Comput. Phys. **33**, 236–248 (1979)
12. Hanisch, M.R.: Multigrid preconditioning for the biharmonic Dirichlet problem. SIAM J. Numer. Anal. **30**(1), 184–214 (1993)
13. Hou, T., Li, Z., Osher, S., Zhao, H.: A hybrid method for moving interface problems with application to the Hele-Shaw flow. J. Comput. Phys. **134**, 236–252 (1997)
14. Hunter, J., Li, Z., Zhao, H.: Autophobic spreading of drops. J. Comput. Phys. **183**, 335–366 (2002)
15. Jiang, G., Peng, D.: Weighted ENO schemes for Hamilton–Jacobi equations. SIAM J. Sci. Comput. **21**, 2126–2143 (2000)

16. Kupferman, R.: A central-difference scheme for a pure stream function formulation of incompressible viscous flow. SIAM J. Sci. Statist. Comput. **23** (2001)
17. LeVeque, R.J., Li, Z.: The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J. Numer. Anal. **31**, 1019–1044 (1994)
18. Li, Z.: A fast iterative algorithm for elliptic interface problems. SIAM J. Numer. Anal. **35**, 230–254 (1998)
19. Li, Z.: IIMPACK: a collection of fortran codes for interface problems. Anonymous ftp at ftp.ncsu.edu under the directory: /pub/math/zhilin/Package and http://www4.ncsu.edu/~zhilin/IIM, last updated (2005)
20. Li, Z., Ito, K., Lai., M.-C.: An augmented approach for Stokes equations with a discontinuous viscosity and singular forces. NCSU-CRSC Tech. Report: CRSC-TR04-23, North Carolina State Univeristy (2004), Computers & Fluids, **36**, 622–635 (2007)
21. Li, Z., Wan, X., Ito, K., Lubkin, S.: An augmented pressure boundary condition for a Stokes flow with a non-slip boundary condition. Commun. Comput. Phys. **1**, 874–885 (2006)
22. Li, Z., Zhao, H., Gao, H.: A numerical study of electro-migration voiding by evolving level set functions on a fixed cartesian grid. J. Comput. Phys. **152**, 281–304 (1999)
23. Mayo, A.: The fast solution of Poisson's and the biharmonic equations on irregular regions. SIAM J. Numer. Anal. **21**, 285–299 (1984)
24. Mayo, A., Greenbaum, A.: Fast parallel iterative solution of Poisson's and the biharmonic equations on irregular regions. SIAM J. Sci. Statist. Comput. **13**, 101–118 (1992)
25. McLaurin, J.W.: A general coupled equation approach for solving the biharmonic boundary value problem. SIAM J. Numer. Anal. **11**, 14–33 (1974)
26. Osher, S., Fedkiw, R.: Level Set Methods and Dynamic Implicit Surfaces. Springer, New York (2002)
27. Bjørstad, P.: Fast numerical solution of the biharmonic dirichlet problem on rectangles. SIAM J. Numer. Anal. **20** (1983)
28. Saad, Y.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput. **7**, 856–869 (1986)
29. Sethian, J.A.: Level Set Methods and Fast Marching Methods, 2nd edn. Cambridge University Press, Cambridge (1999)
30. Smith, J.: The coupled equation approach to the numerical solution of the biharmonic equation by finite differences, I. SIAM J. Numer. Anal. **5**, 323–339 (1968)
31. Smith, J.: The coupled equation approach to the numerical solution of the biharmonic equation by finite differences, II. SIAM J. Numer. Anal. **7**, 104–112 (1970)
32. Stephenson, J.W.: Single cell discretizations of order two and four for biharmonic problems. J. Comput. Phys. **55**, 65–80 (1984)
33. Swarztrauber, P.N.: Fast Poisson solver. In: Golub, G.H. (ed.) Studies in Numerical Analysis, vol. 24, pp. 319–370. MAA (1984)
34. Thatcher, R.W.: A least squares method for solving biharmonic problems. SIAM J. Numer. Anal. **38**, 1523–1539 (2000)
35. Timoshenko, S., Goodier, J.: Theory of Elasticity. McGraw-Hill, New York (1970)