

Research Article

Numerical Method for Solving Matrix Coefficient Elliptic Equation on Irregular Domains with Sharp-Edged Boundaries

Liquan Wang¹ and Liwei Shi²

¹ Department of Mathematics, College of Science, China University of Petroleum, Beijing 102249, China

² Department of Mathematics and Statistics, Louisiana Tech University, Ruston, LA 71272, USA

Correspondence should be addressed to Liwei Shi; sliweihmily@gmail.com

Received 2 June 2013; Accepted 21 November 2013

Academic Editor: Yanping Lin

Copyright © 2013 L. Wang and L. Shi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a new second-order accurate numerical method for solving matrix coefficient elliptic equation on irregular domains with sharp-edged boundaries. Nontraditional finite element method with non-body-fitting grids is implemented on a fictitious domain in which the irregular domains are embedded. First we set the function and coefficient in the fictitious part, and the nonsmooth boundary is then treated as an interface. The emphasis is on the construction of jump conditions on the interface; a special position for the ghost point is chosen so that the method is more accurate. The test function basis is chosen to be the standard finite element basis independent of the interface, and the solution basis is chosen to be piecewise linear satisfying the jump conditions across the interface. This is an efficient method for dealing with elliptic equations in irregular domains with non-smooth boundaries, and it is able to treat the general case of matrix coefficient. The complexity and computational expense in mesh generation is highly decreased, especially for moving boundaries, while robustness, efficiency, and accuracy are promised. Extensive numerical experiments indicate that this method is second-order accurate in the L^∞ norm in both two and three dimensions and numerically very stable.

1. Introduction

Let $\Omega^- \subset \mathbb{R}^d$ ($d = 2, 3$) be an open-bounded domain with a Lipschitz continuous boundary Γ .

We consider the variable coefficient elliptic equation

$$-\nabla \cdot (\beta^-(x) \nabla u^-(x)) = f(x) \quad \text{in } \Omega^-, \quad (1)$$

where $x = (x_1, \dots, x_d)$ refers to the spatial variable, ∇ is the gradient operator, and the right-hand side $f(x)$ is assumed to lie in $L^2(\Omega^-)$. The coefficient $\beta^-(x)$ is a $d \times d$ matrix that is uniformly elliptic, and its entries are continuously differentiable on Ω^- .

For a given function $g(x)$ on the boundary Γ , the Dirichlet boundary condition is prescribed as

$$u^-(x) = g(x) \quad \text{on } \Gamma. \quad (2)$$

Elliptic partial differential equations are often used to construct models of the most basic theories underlying physics and engineering, such as electromagnetism, material

science, and fluid dynamics. Different kinds of boundary conditions arise with the wide range of applications, such as Dirichlet boundary condition, Neumann boundary condition, and Robin boundary condition.

Elliptic equation on irregular domains has been studied by many researchers and several techniques have been developed. Finite element methods use a mesh triangulation to capture the boundary [1–4]. However, in many situations, such as when the boundary is moving, the mesh generation may be both computational expensive and challenging. A more preferred method is to combine the Cartesian grid method with level-set approach [5–7] to capture the boundary. Higher-order accuracy can be achieved by modifying standard difference formulas; examples are Shortley-Weller discretization [8] and the fast solution proposed by Mayo in [9] for solving Poisson or biharmonic equation on irregular regions with smooth boundary.

Since the pioneer work of Peskin [10] in 1977, much attention has been paid to the numerical solution of elliptic equations with discontinuous coefficients and singular

sources on regular Cartesian grids. A large class of finite difference methods have been proposed. The main idea is to use difference scheme and stencils carefully near the interface to incorporate jump conditions and achieve high-order local truncation error using Taylor expansion. The immersed interface method [10, 11] incorporates the interface conditions into the finite difference scheme near the interface to achieve second-order accuracy based on a Taylor expansion in a local coordinate system. The boundary condition capturing method [12] uses the ghost fluid method [13] to capture the boundary conditions. The method extends the solution from one side across the interface using the jump conditions. In [14], the matched interface and boundary method was proposed to solve elliptic equations with smooth interfaces. In [15], the matched interface and boundary method was generalized to treat sharp-edged interfaces. With an elegant treatment, second-order accuracy was achieved in the L^∞ norm. However, for oscillatory solutions, the errors degenerate.

The existing finite element schemes on non-body-fitted meshes are usually designed by modifying the finite element basis near the interface. The extended finite element method [16–18] was proposed by Ted Belytschko and collaborators to ease difficulties in solving problems with localized features that are not efficiently resolved by mesh refinement. In immersed finite element method [19], a Lagrangian solid mesh moves on top of a background Eulerian fluid mesh which spans the entire computational domain. The penalty finite element method [20, 21] modifies the bilinear form near the interface by penalizing the jump of the solution value (with no general flux jump) across the interface.

Also, there has been a large body of work from the finite volume perspective for developing high-order methods for elliptic equations in complex domains, such as [22, 23] for two-dimensional problems and [24] for three-dimensional problems. Another recent work in this area is a class of kernel-free boundary integral (KFBI) methods for solving elliptic BVPs, presented in [25].

In this paper, we implemented the nontraditional finite element method into matrix coefficient elliptic equation on irregular domains. The idea of applying interface problem solvers into noninterface problems is completely new. The nontraditional finite element method is originally developed in [26–32] for solving elliptic or elasticity equations with sharp-edged interfaces. This method uses non-body-fitting Cartesian grids and uses different basis for the solution and test function; its linear system is independent of jump condition. In this way, it is easy to implement, especially for moving interface. It is capable of treating the more general case of variable matrix coefficient and can deliver (almost) second-order accuracy in L^∞ norm. Since the method is based on the assumption of discontinuous solution, in this work, we embed the research domain with irregular boundary into a fictitious rectangular/cube. After setting the function and coefficient in the fictitious part, we construct the jump conditions and calculate the integration on the boundary of the research domain (we call it interface in the following discussion). By implementing the nontraditional finite element method on the fictitious domain containing the irregular domains which are under research, we overcome

the difficulty of capturing the sharp-edged boundary with high accuracy. Both two- and three-dimensional models are studied. Extensive numerical experiments show the efficiency of this method.

The rest of the paper is organized as follows. The next section presents the discretization of two- and three-dimensional irregular domains. In Section 3, we construct the jump conditions and develop the weak formulation of the problem. In Section 4, six extensive numerical experiments are presented to show the second-order accuracy of our method. Finally some conclusions are drawn in Section 5.

2. Discretization of the Domain

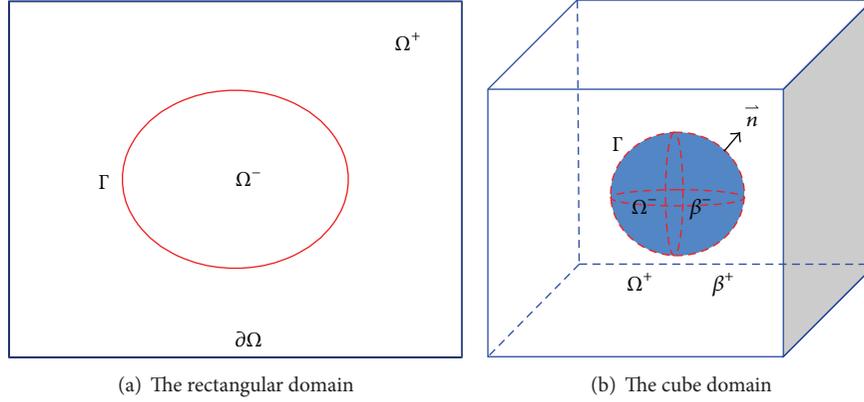
For simplicity, we first embed the irregular domain Ω^- into a fictitious rectangular/cube Ω ; the boundary of Ω is denoted by $\partial\Omega$, the domain between Γ and $\partial\Omega$ is denoted by Ω^+ , and let $\Omega = \partial\Omega \cup \Omega^+ \cup \Gamma \cup \Omega^-$. For ease of discussion, from now on we call Γ the interface and $\partial\Omega$ the boundary of Ω . In this paper, we restrict ourselves to a fictitious rectangular domain $\Omega = (x_{\min}, x_{\max}) \times (y_{\min}, y_{\max})$ in two-dimensional space and a fictitious cube domain $\Omega = (x_{\min}, x_{\max}) \times (y_{\min}, y_{\max}) \times (z_{\min}, z_{\max})$ in three-dimensional space; see Figure 1. We assume that there is a Lipschitz continuous and piecewise smooth level-set function ϕ on Ω , where $\Gamma = \{\phi = 0\}$, $\Omega^- = \{\phi < 0\}$, and $\Omega^+ = \{\phi > 0\}$. A unit vector $n = \nabla\phi/|\nabla\phi|$ can be obtained on Ω^- , which is a unit normal vector of Γ at Γ pointing from Ω^- to Ω^+ .

2.1. Discretization of Two-Dimensional Problems. In the two-dimensional case, β is a 2×2 matrix coefficient that is uniformly elliptic in Ω^- . Given two positive integers I and J , set $\Delta x = (x_{\max} - x_{\min})/I$ and $\Delta y = (y_{\max} - y_{\min})/J$. We define a uniform Cartesian grid with grid points given by $(x_i, y_j) = (x_{\min} + i\Delta x, y_{\min} + j\Delta y)$ for $i = 0, \dots, I$ and $j = 0, \dots, J$. A grid point is called a boundary point if $i = 0, I$ or $j = 0, J$; otherwise, it is called an interior point. The grid size is defined as $h = \max(\Delta x, \Delta y) > 0$.

Two sets of grid functions are needed and they are denoted by

$$\begin{aligned} H^{1,h} &= \{\omega^h = (\omega_{i,j}) : 0 \leq i \leq I, 0 \leq j \leq J\}, \\ H_0^{1,h} &= \{\omega^h = (\omega_{i,j}) \in H^{1,h} \\ &: \omega_{i,j} = 0 \text{ if } i = 0, I \text{ or } j = 0, J\}. \end{aligned} \quad (3)$$

To triangulate the domain Ω into a set of uniform triangular cells, we divide each rectangular region $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$ into two right triangular cells: one is bounded by $x = x_i, y = y_j$, and $y = ((y_{j+1} - y_j)/(x_i - x_{i+1}))(x - x_{i+1}) + y_j$; and the other by $x = x_{i+1}, y = y_{j+1}$, and $y = ((y_{j+1} - y_j)/(x_i - x_{i+1}))(x - x_{i+1}) + y_j$. Collecting all these triangular cells, we obtain a uniform triangulation of Ω ; see Figure 2(a). Note that we can also choose the hypotenuse to be $y = ((y_{j+1} - y_j)/(x_{i+1} - x_i))(x - x_i) + y_j$ and get a different uniform triangulation from the same Cartesian grid. There is no conceptual difference for our method on these two triangulations.

FIGURE 1: The domain of $\Omega = \partial\Omega \cup \Omega^+ \cup \Omega^- \cup \Gamma$.

If $\phi(x_i, y_j) \leq 0$, the grid point (x_i, y_j) is counted as in $\overline{\Omega^-}$; otherwise, it is counted as in Ω^+ .

2.2. Discretization of Three-Dimensional Problems. For the three-dimensional problem, β is a 3×3 matrix coefficient that is uniformly elliptic in Ω^- . Given positive integers I, J , and K , set $\Delta x = (x_{\max} - x_{\min})/I$, $\Delta y = (y_{\max} - y_{\min})/J$, and $\Delta z = (z_{\max} - z_{\min})/K$.

We define a uniform Cartesian grid with grid points given by $(x_i, y_j, z_k) = (x_{\min} + i\Delta x, y_{\min} + j\Delta y, z_{\min} + k\Delta z)$ for $i = 0, \dots, I, j = 0, \dots, J$, and $k = 0, \dots, K$. A grid point is called a boundary point if $i = 0, I, j = 0, J$, or $k = 0, K$; otherwise it is called an interior point. The grid size is defined as $h = \max(\Delta x, \Delta y, \Delta z) > 0$.

The following two sets of grid functions are needed:

$$\begin{aligned} H^{1,h} &= \{\omega^h = (\omega_{i,j,k}) : 0 \leq i \leq I, 0 \leq j \leq J, 0 \leq k \leq K\}, \\ H_0^{1,h} &= \{\omega^h = (\omega_{i,j,k}) \in H^{1,h} \\ &: \omega_{i,j,k} = 0 \text{ if } i = 0, I, \text{ or } j = 0, J, \text{ or } k = 0, K\}. \end{aligned} \quad (4)$$

Each cube cell region $[x_i, x_{i+1}] \times [y_j, y_{j+1}] \times [z_k, z_{k+1}]$ is cut into six tetrahedron cells. Upon collecting all these cells, we obtain a uniform territorialization of the cube cell region Ω ; see Figure 3.

If $\phi(x_i, y_j, z_k) \leq 0$, the grid point (x_i, y_j, z_k) is counted as in $\overline{\Omega^-}$; otherwise it is counted as in Ω^+ .

In both two- and three-dimensional cases, a cell Δ_L belongs to one of the following two different sets:

$$\begin{aligned} \Lambda_1 &= \{\Delta_L \subset \Omega : \text{All the vertexes of cell } \Delta_L \\ &\text{are in the same domain } (\Omega^+ \text{ or } \Omega^-)\}, \end{aligned} \quad (5)$$

$$\begin{aligned} \Lambda_2 &= \{\Delta_L \subset \Omega : \text{The vertexes of cell } \Delta_L \\ &\text{are in different domains } (\Omega^+, \Omega^-)\}. \end{aligned} \quad (6)$$

If a cell belongs to Λ_1 , we call it a regular cell; otherwise we call it an interface cell, written as $\Delta_L = \Delta_L^+ \cup \Delta_L^-$, Δ_L^+ and

Δ_L^- are separated by the interface segment, denoted by $\Gamma_{\Delta_L}^h$; see Figures 2(b) and 4.

Two extension operators are needed. The first one is $T^h : H^{1,h} \rightarrow H_0^1(\Omega)$. For any $\psi^h \in H_0^{1,h}$, $T^h(\psi^h)$ is a standard continuous piecewise linear function, which is a linear function in every cell and $T^h(\psi^h)$ matches ψ^h on grid points. Clearly such a function set, denoted by $H_0^{1,h}$, is a finite dimensional subspace of $H_0^1(\Omega)$. The second extension operator U^h is constructed as follows. For any $u^h \in H^{1,h}$ with $u^h = g^h$ at boundary points, $U^h(u^h)$ is a piecewise linear function and matches u^h on grid points. It is a linear function in each regular cell, just like the first extension operator $U^h(u^h) = T^h(u^h)$ in a regular cell. In each interface cell, it consists of two pieces of linear functions: one is on Δ_L^+ and the other is on Δ_L^- .

3. Construction of Jump Conditions and Weak Formulation

In [26–32], jump conditions a and b are known and are used to derive the linear combination of the values of the points across the interface Γ . However, in this paper, we need to construct a and b for each interface cell $\Delta_L \in \Lambda_2$ from the points with known value across the interface Γ and u^- on Ω^- and then calculate the linear integral

$$\begin{aligned} [u]_{\Gamma}(x) &\equiv u^+(x) - u^-(x) = a(x), \\ [(\beta \nabla u) \cdot n]_{\Gamma}(x) &\equiv n \cdot (\beta^+(x) \nabla u^+(x)) \\ &\quad - n \cdot (\beta^-(x) \nabla u^-(x)) = b(x). \end{aligned} \quad (7)$$

Here β is smooth on Ω and Γ is kept to be Lipschitz continuous. n is the unit vector defined in Section 2.

For different u^+ and β^+ we have different a and b . For simplicity, set u^+ as 0 and $\beta^+ = \beta^-$ on Ω^+ ; then $a(x) = 0 - u^-|_{\Gamma}(x) = -g(x)$. For $b(x)$, since $\nabla u^+ = 0$, the difficult part is to generate ∇u^- . For accuracy consideration, we choose the position of the ghost point $G = (x_G, y_G) \in \Delta_L$ as the midpoint of the grid points on $\Delta_L \cap \Omega^-$. Let u_G be the value of point G ;

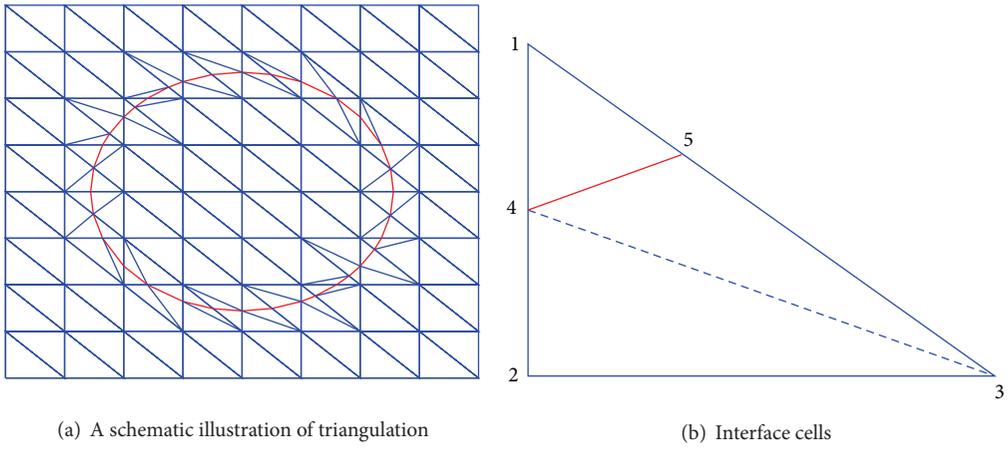


FIGURE 2: Discretization of two-dimensional problems.

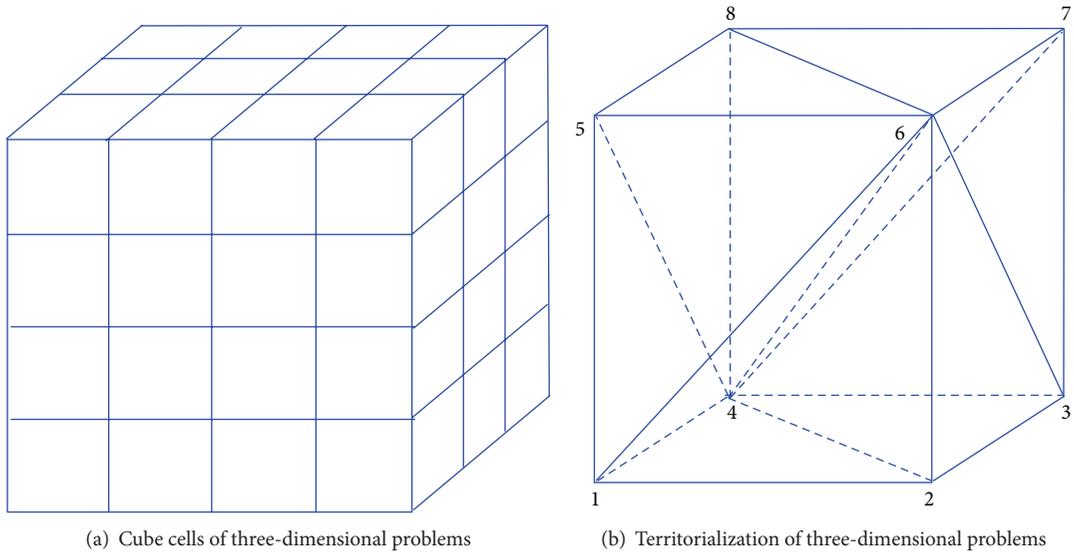


FIGURE 3: Discretization of three-dimensional problems.

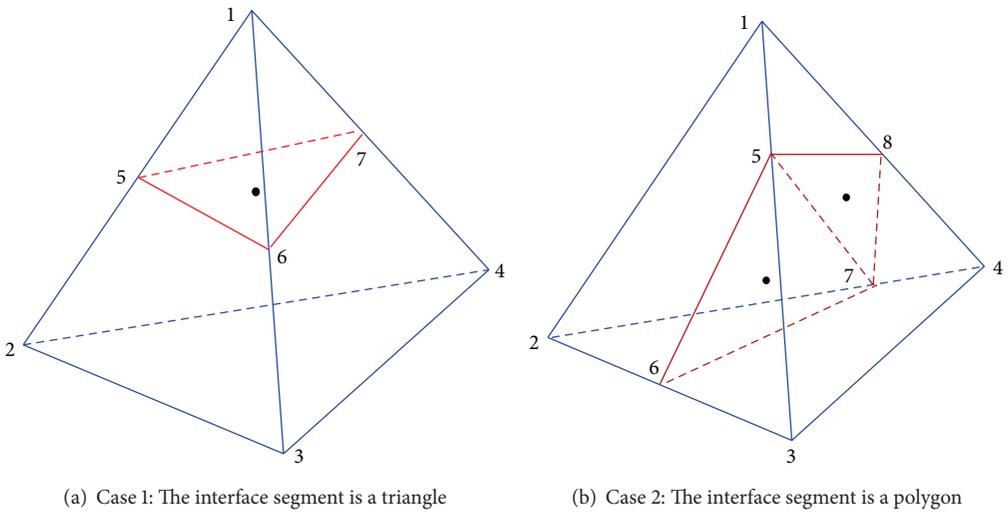


FIGURE 4: The interface segment of three-dimensional problems.

then $u_G^- = (1/m) \sum_{i=1}^m u_i^-$, where u_i^- is the value of the grid point $(x_i, y_i)/(x_i, y_i, z_i)$ on $\Delta_L \cap \Omega^-$.

For two-dimensional problem, cell Δ_{123} is cut into two parts, P_{145} and P_{4235} ; see Figure 2(b). If part $P_{145} \in \Omega^-$, then G is chosen to be point 1, $u_G^- = u_1^-$; if part $P_{4235} \in \Omega^-$, then G is chosen to be the midpoint of points 2 and 3, $u_G^- = (u_2^- + u_3^-)/2$. Since the values of points 4 and 5 are known as $u_4^- = g(x_4, y_4)$ and $u_5^- = g(x_5, y_5)$, the value of ∇u^- can be determined by

$$\begin{aligned} u_x^- &= c_{1,x} u_4^- + c_{2,x} u_5^- + c_{3,x} u_G^-, \\ u_y^- &= c_{1,y} u_4^- + c_{2,y} u_5^- + c_{3,y} u_G^-. \end{aligned} \quad (8)$$

Let

$$n = \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}, \quad \beta^- = \begin{pmatrix} \beta_{11}^- & \beta_{12}^- \\ \beta_{21}^- & \beta_{22}^- \end{pmatrix}. \quad (9)$$

Then

$$\begin{aligned} b &= \beta^+ \nabla u^+ \cdot n - \beta^- \nabla u^- \cdot n \\ &= - \left(\beta_{11}^- u_x^- n_1 + \beta_{12}^- u_y^- n_1 \right. \\ &\quad \left. + \beta_{21}^- u_x^- n_2 + \beta_{22}^- u_y^- n_2 \right). \end{aligned} \quad (10)$$

For three-dimensional problem, there are two kinds of interface cells.

Case 1. The interface cell Δ_{1234} is cut into a triangular pyramid P_{1567} and a pentahedron P_{567234} ; see Figure 4(a). If part $P_{1567} \in \Omega^-$, then G is chosen to be point 1, $u_G^- = u_1^-$; if part $P_{567234} \in \Omega^-$, then G is chosen to be the midpoint of points 2, 3, and 4, $u_G^- = (u_2^- + u_3^- + u_4^-)/3$.

Case 2. The interface cell Δ_{1234} is cut into a tetrahedron P_{125678} and a pentahedron P_{567834} . We can always rotate the cell to make $P_{567834} \in \Omega^-$, such that G is the midpoint of points 3 and 4, $u_G^- = (u_3^- + u_4^-)/2$.

In Figure 4(a), the values of points 5, 6, and 7 are known as $u_5^- = g(x_5, y_5, z_5)$, $u_6^- = g(x_6, y_6, z_6)$, and $u_7^- = g(x_7, y_7, z_7)$; therefore, the value of ∇u^- can be calculated by

$$\begin{aligned} u_x^- &= c_{1,x} u_5^- + c_{2,x} u_6^- + c_{3,x} u_7^- + c_{4,x} u_G^-, \\ u_y^- &= c_{1,y} u_5^- + c_{2,y} u_6^- + c_{3,y} u_7^- + c_{4,y} u_G^-, \\ u_z^- &= c_{1,z} u_5^- + c_{2,z} u_6^- + c_{3,z} u_7^- + c_{4,z} u_G^-. \end{aligned} \quad (11)$$

In Figure 4(b), the values of points 5, 6, 7, and 8 are known as $u_5^- = g(x_5, y_5, z_5)$, $u_6^- = g(x_6, y_6, z_6)$, $u_7^- = g(x_7, y_7, z_7)$, and $u_8^- = g(x_8, y_8, z_8)$; therefore, the value of ∇u^- can be calculated by

$$\begin{aligned} u_x^- &= c_{1,x} u_5^- + c_{2,x} u_6^- + c_{3,x} u_7^- + c_{4,x} u_8^- + c_{5,x} u_G^-, \\ u_y^- &= c_{1,y} u_5^- + c_{2,y} u_6^- + c_{3,y} u_7^- + c_{4,y} u_8^- + c_{5,y} u_G^-, \\ u_z^- &= c_{1,z} u_5^- + c_{2,z} u_6^- + c_{3,z} u_7^- + c_{4,z} u_8^- + c_{5,z} u_G^-. \end{aligned} \quad (12)$$

Let

$$n = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}, \quad \beta^- = \begin{pmatrix} \beta_{11}^- & \beta_{12}^- & \beta_{13}^- \\ \beta_{21}^- & \beta_{22}^- & \beta_{23}^- \\ \beta_{31}^- & \beta_{32}^- & \beta_{33}^- \end{pmatrix}. \quad (13)$$

Then

$$\begin{aligned} b &= \beta^+ \nabla u^+ \cdot n - \beta^- \nabla u^- \cdot n \\ &= - \left(\beta_{11}^- u_x^- n_1 + \beta_{12}^- u_y^- n_1 + \beta_{13}^- u_z^- n_1 + \beta_{21}^- u_x^- n_2 + \beta_{22}^- u_y^- n_2 \right. \\ &\quad \left. + \beta_{23}^- u_z^- n_2 + \beta_{31}^- u_x^- n_3 + \beta_{32}^- u_y^- n_3 + \beta_{33}^- u_z^- n_3 \right). \end{aligned} \quad (14)$$

Lemma 1. All coefficients c in equations (8)–(12) are finite and independent of u^h and a .

Proof. See Lemma 3.1 in [26]. \square

Upon above discussion, the system described in (1)–(2) is equivalent to the following system:

$$\begin{aligned} -\nabla \cdot (\beta \nabla u(x)) &= f(x), \quad x \in \Omega \setminus \Gamma, \\ a(x) &= [u](x) = -u^-(x) = -g(x), \quad x \in \Gamma, \\ b(x) &= [(\beta \nabla u) \cdot n](x) \\ &= -n \cdot \beta^-(x) \nabla u^-(x), \quad x \in \Gamma, \\ u(x) &= 0, \quad x \in \partial\Omega. \end{aligned} \quad (15)$$

Since $u^+ = 0$ is known, we only care about the value on Ω^- . We generalize the weak formulation for the elliptic equation with matrix coefficient. The usual Sobolev space $H^1(\Omega)$ is used. For $H_0^1(\Omega)$, instead of the usual inner product, we choose one which is better suited to our problem:

$$\int_{\Omega^-} \beta \nabla u \cdot \nabla \psi + \int_{\Gamma} b \psi = \int_{\Omega^-} f \psi. \quad (16)$$

In this way, we have the following definition.

Definition 2. A function u with $u|_{\Omega^-} \in H^1(\Omega^-)$ and $u = g$ on Γ is a weak solution of system (1)–(2) if it satisfies

$$\int_{\Omega^-} \beta \nabla u \cdot \nabla \psi + \int_{\Gamma} b \psi = \int_{\Omega^-} f \psi, \quad \forall \psi \in H_0^1(\Omega). \quad (17)$$

A classical solution of system (1)–(2), $u|_{\Omega^-} \in C^2(\overline{\Omega^-})$, is necessarily a weak solution.

Theorem 3. If $f \in L^2(\Omega)$, then there exists a unique weak solution of equations (1)–(2).

Proof. See Theorem 2.1 in [33]. \square

Theorem 4. For all $u^h \in H^{1,h}$, $U^h(u^h)$ can be constructed uniquely, provided T^h and ϕ are given.

Proof. See Theorem 3.1 in [26]. \square

Method 1. Find a discrete function $u^h \in H^{1,h}$ such that $u^h = g^h$ on boundary points and so that for all $\psi^h \in H_0^{1,h}$, we have

$$\begin{aligned} & \sum_{K \in T^h} \left(\int_{K^-} \beta \nabla U^h(u^h) \cdot \nabla T^h(\psi^h) + \int_{T_K^h} b T^h(\psi^h) \right) \\ &= \sum_{K \in T^h} \int_{K^-} f T^h(\psi^h). \end{aligned} \quad (18)$$

Theorem 5. *If β^- is positive definite, then the matrix A for the linear system generated by Method 1 is positive definite.*

Proof. For any vector $c \in R^n$,

$$c^T A c = \sum_{i,j=1}^n a_{ij} c_i c_j = B \left[\sum_{i=1}^n c_i u^i, \sum_{i=1}^n c_i \psi^i \right], \quad (19)$$

where u^i and ψ^i are basis functions for the solution and the test functions, respectively. According to Lemma 1, matrix A is independent of a ; choose a such that $u^i = \psi^i$ everywhere. Let

$$\sum_{i=1}^n c_i u^i = \sum_{i=1}^n c_i \psi^i = w. \quad (20)$$

Considering the positive definiteness of β^- , we have

$$c^T A c = B[w, w] = \int_{\Omega^-} (\beta(x) \nabla w(x))^T \nabla w(x) dx > 0. \quad (21)$$

Therefore, A is positive definite. \square

In our implementation, the integrals are computed with Gaussian quadrature rule. For each cell, we denote the midpoint of each edge $u_i u_j$ as p_{ij} . In numerical computation, we apply the average of all the $f(p_{ij})$ in each cell.

4. Numerical Experiments

Of all the numerical experiments below, in $\overline{\Omega^-}$, the level-set function ϕ , the coefficients β^- , and the solutions u^- are given. Hence the source term f and the Dirichlet boundary condition g can be directly evaluated and the jump conditions a and b can be constructed. The L^∞ norm of the error in solution is measured over the domain Ω^- . The uniform Cartesian grid is restricted to the whole domain Ω .

4.1. Numerical Examples of Two-Dimensional Problems

Example 1. This example has a “star” interface. The level-set function ϕ , the coefficients β^- , and the solution u^- are given as follows:

$$\phi(r, \theta) = \begin{cases} -\frac{R \sin(\theta_i/2)}{\sin((\theta_i/2) + \theta - \theta_r - (2\pi(i-1)/5))} + r, \\ \theta_r + \frac{\pi(2i-2)}{5} \leq \theta < \theta_r + \frac{\pi(2i-1)}{5}, \\ -\frac{R \sin(\theta_i/2)}{\sin((\theta_i/2) - \theta + \theta_r - (2\pi(i-1)/5))} + r, \\ \theta_r + \frac{\pi(2i-3)}{5} \leq \theta < \theta_r + \frac{\pi(2i-2)}{5}, \end{cases} \quad (22)$$

TABLE 1: Numerical error of the problem with “star” shape.

| $n_x \times n_y$ | Error in u^- | Order |
|------------------|----------------|-------|
| 20×20 | 0.51 | |
| 40×40 | 0.20 | 1.32 |
| 80×80 | 0.061 | 1.73 |
| 160×160 | 0.016 | 1.90 |
| 320×320 | 0.0042 | 1.96 |

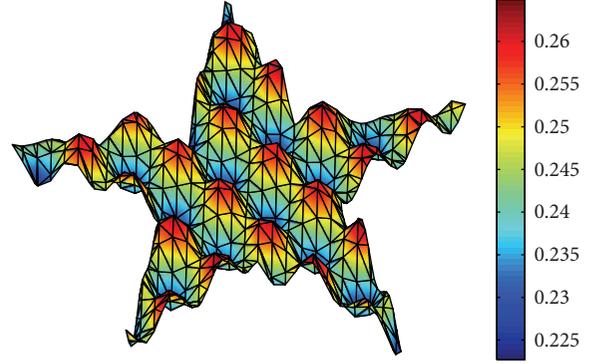


FIGURE 5: Numerical result of the problem with “star” shape.

with $\theta_t = \pi/5$, $\theta_r = \pi/7$, $R = 6/7$, and $i = 1, 2, 3, 4, 5$

$$\beta^-(x, y) = \begin{pmatrix} x^2 + 5 & xy + 1 \\ x + 2 & y^2 + 3 \end{pmatrix}, \quad (23)$$

$$u^-(x, y) = 6 + \sin(6\pi x) \sin(6\pi y).$$

The numerical result with our method using a 40×40 uniform Cartesian grid is shown in Figure 5. Table 1 shows the error on different grids.

Example 2. This example has a “happy face” interface. The level-set function $\phi(x, y)$, the coefficients β^- , and the solution u^- are given as follows:

$$\begin{aligned} \phi(x, y) = & \max(\min(\phi_1, \phi_2, \phi_3), \phi_4, \\ & \phi_5, \phi_6, \min(\phi_7, \phi_8)), \end{aligned}$$

$$\phi_1(x, y) = x^2 + y^2 - 0.75^2 - 0.15^2,$$

$$\phi_2(x, y) = (x - 0.75)^2 + y^2 - 0.15^2,$$

$$\phi_3(x, y) = (x + 0.75)^2 + y^2 - 0.15^2,$$

$$\begin{aligned} \phi_4(x, y) = & -\frac{0.1}{0.12}(x - 0.2)^2 \\ & -\frac{0.12}{0.1}(y - 0.22)^2 + 0.12 \cdot 0.1, \end{aligned}$$

TABLE 2: Numerical error of the problem with “face” shape.

| $n_x \times n_y$ | Error in u^- | Order |
|------------------|----------------|-------|
| 20×20 | 0.016 | |
| 40×40 | 0.0041 | 1.97 |
| 80×80 | 0.0010 | 1.97 |
| 160×160 | $2.58e - 004$ | 2.02 |

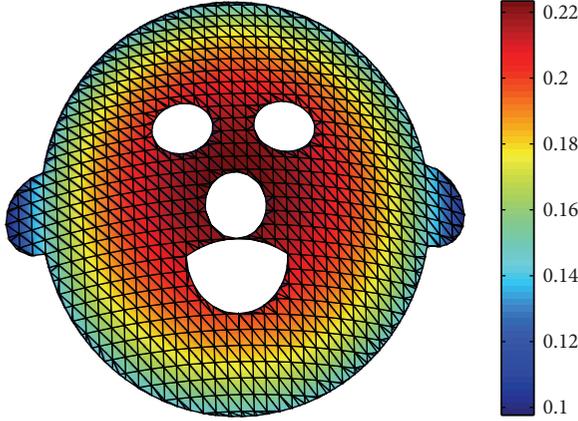


FIGURE 6: Numerical result of the problem with “face” shape.

$$\begin{aligned}
\phi_5(x, y) &= -\frac{0.1}{0.12}(x+0.2)^2 \\
&\quad -\frac{0.12}{0.1}(y-0.22)^2 + 0.12 \cdot 0.1, \\
\phi_6(x, y) &= -x^2 - (y+0.08)^2 + 0.12^2, \\
\phi_7(x, y) &= -x^2 - (y+0.625)^2 + 0.425^2, \\
\phi_8(x, y) &= -x^2 - (y+0.25)^2 + 0.2^2, \\
\beta^-(x, y) &= \begin{pmatrix} x^2 - y^2 + 3 & x^2 - y^2 + 1 \\ x^2 - y^2 + 1 & x^2 - y^2 + 4 \end{pmatrix}, \\
u^-(x, y) &= 5 - 5x^2 - 5y^2.
\end{aligned} \tag{24}$$

The numerical result with our method using a 40×40 uniform Cartesian grid is shown in Figure 6. Table 2 shows the error on different grids.

Example 3. This example has a “target” interface. The level-set function ϕ , the coefficients β^- , and the solution u^- are given as follows:

$$\begin{aligned}
\phi(x, y) &= \min(\phi_1, \min(\min(\min_1, \min_2), \\
&\quad \min(\min_3, \min_4))), \\
\phi_1(x, y) &= x^2 + y^2 - 0.02, \\
\phi_2(x, y) &= x^2 + y^2 - 0.08, \\
\phi_3(x, y) &= x^2 + y^2 - 0.15,
\end{aligned}$$

$$\phi_4(x, y) = x^2 + y^2 - 0.25,$$

$$\phi_5(x, y) = x^2 + y^2 - 0.35,$$

$$\phi_6(x, y) = x^2 + y^2 - 0.45,$$

$$\phi_7(x, y) = x^2 + y^2 - 0.55,$$

$$\phi_8(x, y) = x^2 + y^2 - 0.80,$$

$$\phi_9(x, y) = x^2 + y^2 - 1,$$

$$\min_1 = -\min(\phi_2, -\phi_3),$$

$$\min_2 = -\min(\phi_4, -\phi_5),$$

$$\min_3 = -\min(\phi_6, -\phi_7),$$

$$\min_4 = -\min(\phi_8, -\phi_9),$$

$$\beta^-(x, y) = \begin{pmatrix} 2 + \sin(x+y) & 0 \\ 0 & 2 + \sin(x+y) \end{pmatrix},$$

$$u^-(x, y) = x^2 + y^2 + \sin(x+y).$$

(25)

The numerical result with our method using a 40×40 uniform Cartesian grid is shown in Figure 7. Table 3 shows the error on different grids.

4.2. Numerical Examples of Three-Dimensional Problems

Example 4. The level-set function ϕ , the coefficients β^- , and the solution u^- are given as follows:

$$\phi(x, y, z) = \max(\min_3, \phi_6),$$

$$\phi_1(x, y, z) = x^2 + y^2 - 0.02,$$

$$\phi_2(x, y, z) = x^2 + y^2 - 0.1,$$

$$\phi_3(x, y, z) = x^2 + y^2 - 0.3,$$

$$\phi_4(x, y, z) = x^2 + y^2 - 0.6,$$

$$\phi_5(x, y, z) = x^2 + y^2 - 0.89,$$

$$\phi_6(x, y, z) = x^2 + y^2 + z^2 - 0.9,$$

$$\min_1 = -\min(\phi_2, -\phi_3),$$

$$\min_2 = -\min(\phi_4, -\phi_5),$$

$$\min_3 = \min(\phi_1, \min(\min_1, \min_2)),$$

$$\beta^-(x, y, z)$$

$$= \begin{pmatrix} 4x^2 + 6 & \sin(x+y) & x \\ \sin(x+y) & 2z^2 + 3 & 0.5 \sin(x) \\ x & 0.5 \sin(x) & \cos(z) + 5 \end{pmatrix},$$

$$u^-(x, y, z) = 5 - \sin(x^2) + 3y^2 + z.$$

(26)

TABLE 3: Numerical error of the problem with “target” shape.

| $n_x \times n_y$ | Error in u^- | Order |
|------------------|----------------|-------|
| 20×20 | 0.0034 | |
| 40×40 | $8.44e - 004$ | 1.99 |
| 80×80 | $2.14e - 004$ | 1.98 |
| 160×160 | $5.35e - 005$ | 2.00 |

TABLE 4: Numerical error of Example 4.

| $n_x \times n_y \times n_z$ | Error in u^- | Order |
|-----------------------------|----------------|-------|
| $10 \times 10 \times 10$ | 0.016 | |
| $20 \times 20 \times 20$ | 0.0050 | 1.70 |
| $40 \times 40 \times 40$ | 0.0013 | 1.96 |
| $80 \times 80 \times 80$ | 0.00032 | 2.02 |

TABLE 5: Numerical error of Example 5.

| $n_x \times n_y \times n_z$ | Error in u^- | Order |
|-----------------------------|----------------|-------|
| $10 \times 10 \times 10$ | 0.00090 | |
| $20 \times 20 \times 20$ | 0.00025 | 1.86 |
| $40 \times 40 \times 40$ | 0.000062 | 2.01 |
| $80 \times 80 \times 80$ | 0.000015 | 2.00 |

Figure 8 shows the interface and boundary condition using a $40 \times 40 \times 40$ uniform Cartesian grid. Table 4 shows the error on different grids.

Example 5. The level-set function ϕ , the coefficients β^- , and the solution u^- are given as follows:

$$\begin{aligned}
 \phi(x, y, z) &= \min(\min(\phi_1, \phi_2), \min(\phi_3, \phi_4)), \\
 \phi_1(x, y, z) &= -0.25 + x^2 + y^2 + (z + 0.5)^2, \\
 \phi_2(x, y, z) &= -0.25 + (x - 0.4)^2 + y^2 + z^2, \\
 \phi_3(x, y, z) &= -0.25 + (x + 0.3)^2 + y^2 + z^2, \\
 \phi_4(x, y, z) &= -0.25 + x^2 + (y + 0.5)^2 + z^2, \\
 \beta^-(x, y, z) &= \begin{pmatrix} \cos(x+y)^2 + 3 & z & 0.2 \sin(z-x) \\ z & z^2 + 5 & y \\ 0.2 \sin(z-x) & y & \sin(z)^2 + 2 \end{pmatrix}, \\
 u^-(x, y, z) &= \exp\left(-\frac{(x^2 + y^2 + z^2)}{20}\right).
 \end{aligned} \tag{27}$$

Figure 9 shows the interface and boundary condition using a $40 \times 40 \times 40$ uniform Cartesian grid. Table 5 shows the error on different grids.

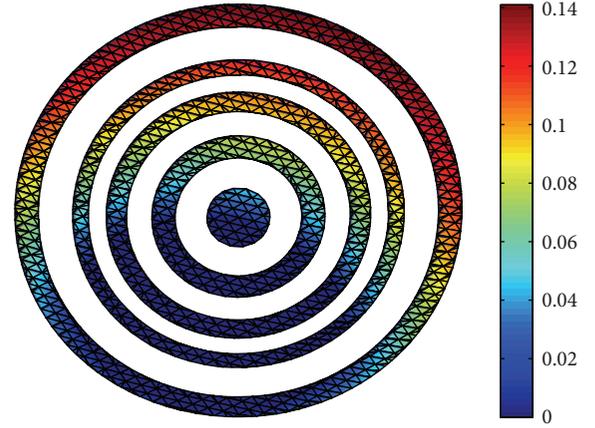


FIGURE 7: Numerical result of the problem with “target” shape.

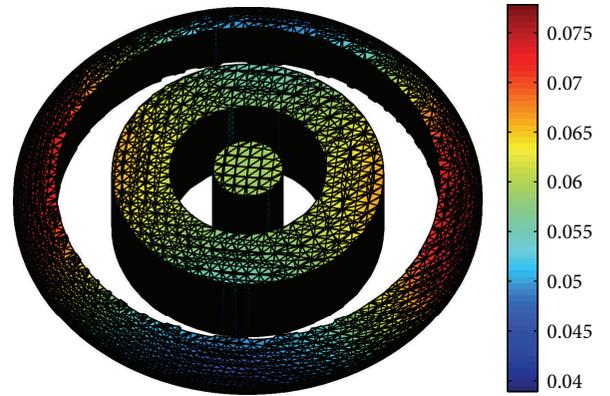


FIGURE 8: Numerical result of Example 4.

Example 6. The level-set function ϕ , the coefficients β^- , and the solution u^- are given as follows:

$$\begin{aligned}
 \phi(x, y, z) &= \min\left((x - 0.39)^2 + y^2 + z^2, \right. \\
 &\quad \left. -0.25, (x + 0.39)^2 + y^2 + z^2 - 0.25\right), \\
 \beta^-(x, y, z) &= \begin{pmatrix} 4x^2 + 6 & \sin(x+y) & xy \\ \sin(x+y) & 2z^2 + 3 & 0.5 \sin(x) \\ xy & 0.5 \sin(x) & \cos(xy+z)^2 + 5 \end{pmatrix}, \\
 u^-(x, y, z) &= 10 - 2x^3 + 3y^2 + \sin(z - y).
 \end{aligned} \tag{28}$$

Figure 10 shows the interface and boundary condition using a $40 \times 40 \times 40$ uniform Cartesian grid. Table 6 shows the error on different grids.

5. Conclusion

We developed a novel numerical method for solving matrix valued coefficient elliptic equation on irregular domains. The research domain with sharp-edged boundary is embedded

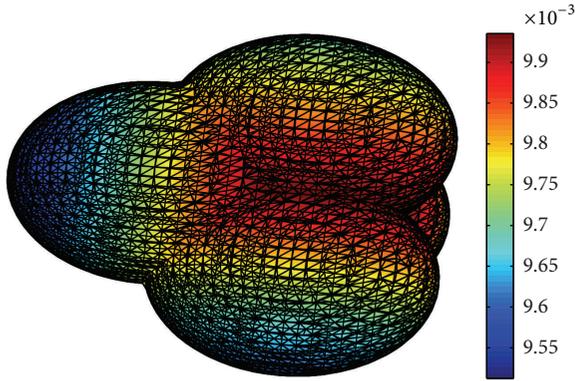


FIGURE 9: Numerical result of Example 5.

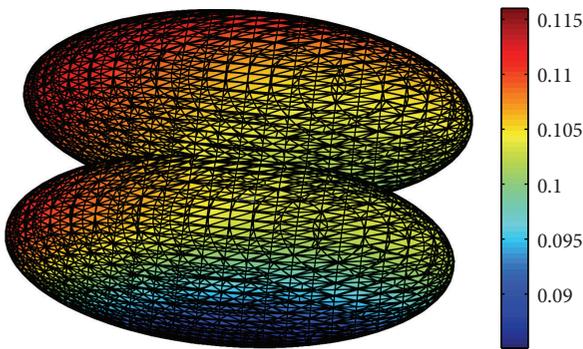


FIGURE 10: Numerical result of Example 6.

TABLE 6: Numerical error of Example 6.

| $n_x \times n_y \times n_z$ | Error in u^- | Order |
|-----------------------------|----------------|-------|
| $10 \times 10 \times 10$ | 0.041 | |
| $20 \times 20 \times 20$ | 0.011 | 1.92 |
| $40 \times 40 \times 40$ | 0.0029 | 1.88 |
| $80 \times 80 \times 80$ | 0.00076 | 1.95 |

into a fictitious rectangular/cube and ghost point is used to construct the jump conditions on the interface. Nontraditional finite element method, which employs different basis for the solution and the test function, is implemented in the whole domain. Both two- and three-dimensional problems are discussed in this paper. Extensive numerical experiments show that this method is second-order accurate in the L^∞ norm in two and three dimensions. The work shows the robustness and efficiency of the method and indicates that this method can be used as an alternative way to solve elliptic equations; it lays a foundation for interface problems on irregular domains with various kinds of boundary conditions; as such, the variety of problems that are solvable is largely expanded.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to the editor and referees for their valuable comments and suggestions. L. Wang’s research is supported by Science Foundation of China University of Petroleum Beijing (no. YJRC-2013-48).

References

- [1] A. Quarteroni, *Numerical Models for Differential Problems*, vol. 2, Springer, Milan, Italy, 2009.
- [2] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, vol. 23, Springer, Berlin, Germany, 1994.
- [3] R. H. Nochetto, M. Paolini, and C. Verdi, “An adaptive finite element method for two-phase Stefan problems in two space dimensions. II. Implementation and numerical experiments,” *SIAM Journal on Scientific and Statistical Computing*, vol. 12, no. 5, pp. 1207–1244, 1991.
- [4] A. Schmidt, “Computation of three dimensional dendrites with finite elements,” *Journal of Computational Physics*, vol. 125, pp. 293–312, 1996.
- [5] M. Sussman, P. Smereka, and S. Osher, “A level set approach for computing solutions to incompressible 2-phase flow,” *Journal of Computational Physics*, vol. 114, pp. 146–159, 1994.
- [6] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, vol. 153, Springer, New York, NY, USA, 2003.
- [7] A. du Chéné, C. Min, and F. Gibou, “Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes,” *Journal of Scientific Computing*, vol. 35, no. 2-3, pp. 114–131, 2008.
- [8] G. H. Shortley and R. Weller, “The numerical solution of Laplaces equation,” *Journal of Applied Physics*, vol. 9, pp. 334–348, 1938.
- [9] A. Mayo, “The fast solution of Poisson’s and the biharmonic equations on irregular regions,” *SIAM Journal on Numerical Analysis*, vol. 21, no. 2, pp. 285–299, 1984.
- [10] C. S. Peskin, “Numerical analysis of blood flow in the heart,” *Journal of Computational Physics*, vol. 25, no. 3, pp. 220–252, 1977.
- [11] C. S. Peskin and B. F. Printz, “Improved volume conservation in the computation of flows with immersed elastic boundaries,” *Journal of Computational Physics*, vol. 105, no. 1, pp. 33–46, 1993.
- [12] X.-D. Liu, R. P. Fedkiw, and M. Kang, “A boundary condition capturing method for Poisson’s equation on irregular domains,” *Journal of Computational Physics*, vol. 160, no. 1, pp. 151–178, 2000.
- [13] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, “A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method),” *Journal of Computational Physics*, vol. 152, no. 2, pp. 457–492, 1999.
- [14] S. Yu, Y. Zhou, and G. W. Wei, “Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces,” *Journal of Computational Physics*, vol. 224, no. 2, pp. 729–756, 2007.
- [15] Y. C. Zhou, S. Zhao, M. Feig, and G. W. Wei, “High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources,” *Journal of Computational Physics*, vol. 213, no. 1, pp. 1–30, 2006.
- [16] N. Moes, J. Dolbow, and T. Belytschko, “A finite element method for crack growth without remeshing,” *International Journal for*

- Numerical Methods in Engineering*, vol. 46, no. 1, pp. 131–150, 1999.
- [17] G. J. Wagner, N. Moës, W. K. Liu, and T. Belytschko, “The extended finite element method for rigid particles in Stokes flow,” *International Journal for Numerical Methods in Engineering*, vol. 51, no. 3, pp. 293–313, 2001.
- [18] J. Chessa and T. Belytschko, “An extended finite element method for two-phase fluids: flow simulation and modeling,” *Journal of Applied Mechanics*, vol. 70, no. 1, pp. 10–17, 2003.
- [19] Y. Gong, B. Li, and Z. Li, “Immersed-interface finite-element methods for elliptic interface problems with nonhomogeneous jump conditions,” *SIAM Journal on Numerical Analysis*, vol. 46, no. 1, pp. 472–495, 2008.
- [20] I. Babuška, “The finite element method for elliptic equations with discontinuous coefficients,” *Computing*, vol. 5, pp. 207–213, 1970.
- [21] A. Hansbo and P. Hansbo, “An unfitted finite element method, based on Nitsche’s method, for elliptic interface problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 47–48, pp. 5537–5552, 2002.
- [22] P. Colella and H. Johansen, “A Cartesian grid embedded boundary method for Poissons equation on irregular domains,” *Journal of Computational Physics*, vol. 60, pp. 85–147, 1998.
- [23] M. Oevermann and R. Klein, “A Cartesian grid finite volume method for elliptic equations with variable coefficients and embedded interfaces,” *Journal of Computational Physics*, vol. 219, no. 2, pp. 749–769, 2006.
- [24] M. Oevermann, C. Scharfenberg, and R. Klein, “A sharp interface finite volume method for elliptic equations on Cartesian grids,” *Journal of Computational Physics*, vol. 228, no. 14, pp. 5184–5206, 2009.
- [25] W. Ying and C. S. Henriquez, “A kernel-free boundary integral method for elliptic boundary value problems,” *Journal of Computational Physics*, vol. 227, no. 2, pp. 1046–1074, 2007.
- [26] S. Hou, W. Wang, and L. Wang, “Numerical method for solving matrix coefficient elliptic equation with sharp-edged interfaces,” *Journal of Computational Physics*, vol. 229, no. 19, pp. 7162–7179, 2010.
- [27] S. Hou, Z. Li, L. Wang, and W. Wang, “A numerical method for solving elasticity equations with interfaces,” *Communications in Computational Physics*, vol. 12, no. 2, pp. 595–612, 2012.
- [28] S. Hou, L. Wang, and W. Wang, “A numerical method for solving the elliptic interface problems with multi-domains and triple junction points,” *Journal of Computational Mathematics*, vol. 30, no. 5, pp. 504–516, 2012.
- [29] L. Wang, S. Hou, and L. Shi, “A numerical method for solving 3D elasticity equations with sharp-edged interfaces,” *International Journal of Partial Differential Equations*, vol. 2013, Article ID 476873, 13 pages, 2013.
- [30] L. Wang, S. Hou, and L. Shi, “An improved non-traditional finite element formulation for solving the elliptic interface problems,” *Journal of Computational Mathematics*. In press.
- [31] S. Hou, P. Song, L. Wang, and H. Zhao, “A weak formulation for solving elliptic interface problems without body fitted grid,” *Journal of Computational Physics*, vol. 249, pp. 80–95, 2013.
- [32] L. Wang, *A numerical method for solving the elliptic and elasticity problems [Ph.D. Dissertation]*, Louisiana Tech University, 2011.
- [33] S. Hou and X.-D. Liu, “A numerical method for solving variable coefficient elliptic equation with interfaces,” *Journal of Computational Physics*, vol. 202, no. 2, pp. 411–445, 2005.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

