

AN IMPROVED NON-TRADITIONAL FINITE ELEMENT FORMULATION FOR SOLVING THE ELLIPTIC INTERFACE PROBLEMS*

Liqun Wang

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

Email: wliqunhmily@gmail.com

Songming Hou Liwei Shi

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

Email: shou@latech.edu sliweihmily@gmail.com

Abstract

We propose a non-traditional finite element method with non-body-fitting grids to solve the matrix coefficient elliptic equations with sharp-edged interfaces. All possible situations that the interface cuts the grid are considered. Both Dirichlet and Neumann boundary conditions are discussed. The coefficient matrix data can be given only on the grids, rather than an analytical function. Extensive numerical experiments show that this method is second order accurate in the L^∞ norm.

Mathematics subject classification: 65N30.

Key words: Elliptic equation, Sharp-edged interface, Jump condition, Matrix coefficient.

1. Introduction and Formulations

Elliptic interface problems are widely used in a variety of disciplines when there are multi-physics and multi-phase materials, such as in electromagnetics, material science, fluid dynamics and so on.

We consider a rectangular domain $\Omega = (x_{min}, x_{max}) \times (y_{min}, y_{max})$. Γ is an interface prescribed by the zero level-set $\{(x, y) \in \Omega \mid \phi(x, y) = 0\}$ of a level-set function $\phi(x, y)$. The advantage of using the level-set function is to represent interface cut locations on the grids without having to parameterize the interface. The unit normal vector of Γ is $n = \frac{\nabla\phi}{|\nabla\phi|}$ pointing from $\Omega^- = \{(x, y) \in \Omega \mid \phi(x, y) \leq 0\}$ to $\Omega^+ = \{(x, y) \in \Omega \mid \phi(x, y) \geq 0\}$. Consider the problem

$$\begin{aligned} -\nabla \cdot (\beta(x)\nabla u(x)) &= f(x), & x \in \Omega^\pm, \\ [u(x)] &= a(x), & x \in \Gamma, \\ [(\beta(x)\nabla u(x)) \cdot n] &= b(x), & x \in \Gamma, \\ u(x) &= g(x), & x \in \partial\Omega, \\ \text{or } \frac{\partial u(x)}{\partial n} &= 0, & x \in \partial\Omega, \end{aligned}$$

where $x = (x_1, \dots, x_d)$ denotes the spatial variables and ∇ is the gradient operator. The coefficient $\beta(x)$ is assumed to be a $d \times d$ matrix that is uniformly elliptic on each disjoint

* Received July 1, 2013 / Revised version received August 17, 2013 / Accepted September 16, 2013 /
Published online January 22, 2014 /

subdomain, Ω^- and Ω^+ , and its components are continuously differentiable on each disjoint subdomain, but they may be discontinuous across the interface Γ . The right-hand side $f(x)$ is assumed to lie in $L^2(\Omega)$. We have the following jump conditions:

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

We introduce the weak solution by the standard procedure of multiplying by a test function and integrating by parts: for the problem with Dirichlet Boundary Condition,

$$\int_{\Omega^+} \beta \nabla u \cdot \nabla \psi + \int_{\Omega^-} \beta \nabla u \cdot \nabla \psi = \int_{\Omega} f \psi - \int_{\Gamma} b \psi; \quad (1.1)$$

and for the problem with Neumann Boundary Condition,

$$\int_{\Omega^+} \beta \nabla u \cdot \nabla \psi + \int_{\Omega^-} \beta \nabla u \cdot \nabla \psi + \int_{\partial\Omega} \frac{\partial u}{\partial n} \psi = \int_{\Omega} f \psi - \int_{\Gamma} b \psi, \quad (1.2)$$

where ψ is in H_0^1 for equation 1.1 and H^1 for (1.2).

The pioneering work on this topic was done by Peskin in 1977. The method he proposed was called ‘‘immersed boundary’’ method [11, 12]. It uses a numerical approximation of the δ -function, which smears out the solution on a thin finite band around the interface Γ . In [13], the ‘‘immersed boundary’’ method was combined with the level set method, resulting in a first order numerical method that is simple to implement, even in multiple spatial dimensions. However, for both methods, the numerical smearing at the interface forces continuity of the solution at the interface, regardless of the interface condition $[u] = a$, where a might not be zero.

To achieve high order accuracy, 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. Using finite difference scheme typically requires taking high order derivatives of jump conditions and interface in Taylor expansion. Also property of the discretized linear system is hard to analyze for interface problem with general jump condition.

The ‘‘immersed interface’’ method presented in [3] can get second-order accuracy. This method incorporates the interface conditions into the finite difference stencil, provided that neither of the two jump conditions are zero. The corresponding linear system is sparse, but not symmetric or positive definite. Various applications and extensions of the ‘‘immersed interface’’ method are discussed in [6].

In [4], on basis of the ‘‘immersed interface’’ method, a fast iterative method was proposed to solve constant coefficient problems with the interface conditions $[u] = 0$ and $[\beta u_n] \neq 0$. Non-body-fitting Cartesian grids are used, and then associated uniform triangulations are added on. Interfaces are not necessarily aligned with cell boundaries. Numerical evidence shows that this method’s conforming version achieves second order accuracy in the L^∞ norm, and higher than first order for its non-conforming version.

Using finite element method developed in [17], elliptic problems with the interface conditions $[u] = 0$ and $[\beta u_n] \neq 0$ can obtain second order accuracy in energy norm and nearly second order accuracy in the L^2 norm. Interfaces are aligned with cell boundaries.

In [9, 10], the solution is extended to a rectangular region by using Fredholm integral equations. The proposed method can deal with interface conditions $[u] \neq 0$ and $[u_n] = 0$ and when Greens function is available. The discrete Laplacian was evaluated using these jump conditions

and a fast Poisson solver can be used to compute the extended solution. It can achieve second or higher-order accuracy.

In order to develop a method that is robust and simple to implement, the boundary condition capturing method [7] uses the Ghost fluid method (GFM) [1] to capture the boundary conditions. This method was sped up by a multi-grid method [14]. The convergence proof for the method is provided in [8]. The boundary condition capturing method can be obtained from discretizing the weak formulation provided in [8]. The convergence proof follows naturally. The method can solve the elliptic equation with interface conditions $[u] \neq 0$ and $[\beta u_n] \neq 0$ in two and three dimensions. However, the original version is only first order accurate. In a recent work [22], the method is improved to second order accuracy for smooth interfaces.

In [2], a non-traditional finite element formulation for solving elliptic equations with smooth or sharp-edged interfaces was proposed with non-body-fitting grids for $[u] \neq 0$ and $[\beta u_n] \neq 0$. It achieved second order accuracy in the L^∞ norm for smooth interfaces and about 0.8th order for sharp-edged interfaces. In [18], the matched interface and boundary (MIB) method was proposed to solve elliptic equations with smooth interfaces. In [16], the MIB 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.

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 [19,20] for two dimensional problems and [21] 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 [15].

In this paper, we further improve the method introduced in [23–27]. The major improvement are as follows: first, we discussed all the possible ways the interface cuts the grid. In [23], for simplicity, if the interface hits one or more grid points exactly, we used a perturbation to move it away. This is not desired, as it is not a natural way to handle the problem. With the improved method, no perturbation is applied. The resulting linear system is still (non-symmetric) positive definite if β is positive definite and lower order terms are not present. Our second improvement is, not only Dirichlet boundary condition but also Neumann boundary condition is considered. Our third improvement is, the coefficient matrix data can only be given at grid points, not as an analytic function. This makes our method more practical than before.

Extensive numerical experiments demonstrate the effectiveness of the above improvement. The condition number of the coefficient matrix is also studied. The growth rate is the same as that without the interface.

2. Numerical Method

For ease of discussion in this section, and for accuracy test in the next section, we assume that a , b are smooth on the closure of Ω , β and f are smooth on Ω^+ and Ω^- , but they may be discontinuous across the interface Γ . However $\partial\Omega$, $\partial\Omega^-$ and $\partial\Omega^+$ are kept to be Lipschitz continuous. 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 $n = \frac{\nabla\phi}{|\nabla\phi|}$ can be obtained on $\bar{\Omega}$, which is a unit normal vector of Γ pointing from Ω^- to Ω^+ .

In this paper, we restrict ourselves to a rectangular domain $\Omega = (x_{min}, x_{max}) \times (y_{min}, y_{max})$ in the plane, β is a 2×2 matrix that is uniformly elliptic in each subdomain. Given positive integers I and J , set $\Delta x = (x_{max} - x_{min})/I$ and $\Delta y = (y_{max} - y_{min})/J$. Define $(x_i, y_j) =$

$(x_{min} + i\Delta x, y_{min} + j\Delta y)$ for $i = 0, \dots, I$ and $j = 0, \dots, J$ as a uniform Cartesian grid. Each (x_i, y_j) is called a grid point. For the case $i = 0, I$ or $j = 0, J$, a grid point is called a boundary point, 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

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

Cut every rectangular region $[x_i, x_{i+1}] \times [y_j, y_{j+1}]$ into two pieces of right triangular regions: one is bounded by $x = x_i, y = y_j$ and $y = \frac{y_{j+1}-y_j}{x_i-x_{i+1}}(x - x_{i+1}) + y_j$, the other is bounded by $x = x_{i+1}, y = y_{j+1}$ and $y = \frac{y_{j+1}-y_j}{x_i-x_{i+1}}(x - x_{i+1}) + y_j$. Collecting all those triangular regions, we obtain a uniform triangulation $T^h : \bigcup_{K \in T^h} K$, see Fig. 2.1. We can also choose the hypotenuse to be $y = \frac{y_{j+1}-y_j}{x_{i+1}-x_i}(x - x_i) + y_j$, and get another uniform triangulation from the same Cartesian grid. There is no conceptual difference on these two triangulations for our method.

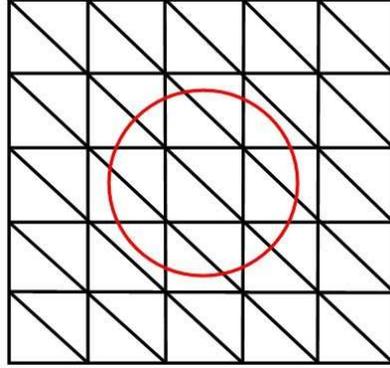


Fig. 2.1. A uniform triangulation.

If $\phi(x_i, y_j) \leq 0$, we count the grid point (x_i, y_j) as in $\overline{\Omega^-}$; otherwise we count it as in Ω^+ . An edge of a triangle in the triangulation is called an interface edge if two of its ends (vertices of triangles in the triangulation) belong to different subdomains; otherwise we call it a regular edge.

A cell K is called an interface cell if its vertices belong to different subdomains. In an interface cell, we write $K = K^+ \cup K^-$. K^+ and K^- are separated by a straight line segment, denoted by Γ_K^h . The two end points of the line segment Γ_K^h are located on interface Γ and their locations can be calculated from the linear interpolations of the discrete level-set functions $\phi^h = \phi(x_i, y_j)$. The vertices of K^+ are located in $\Omega^+ \cup \Gamma$ and the vertices of K^- are located in $\Omega^- \cup \Gamma$. K^+ and K^- are approximations of the regions $K \cap \Omega^+$ and $K \cap \Omega^-$, respectively. A cell K is called a regular cell if all its vertices belong to the same subdomain, either Ω^+ or Ω^- . For a regular cell, we also write $K = K^+ \cup K^-$, where $K^- = \{\}$ (empty set) if all vertices of K are in Ω^+ , and $K^+ = \{\}$ if all vertices of K are in Ω^- . Clearly $\Gamma_K^h = \{\}$ in a regular cell, K^+ and K^- are approximations of the regions $K \cap \Omega^+$ and $K \cap \Omega^-$, respectively. We use $|K^+|$ and $|K^-|$ to represent the areas of K^+ and K^- , respectively.

Two extension operators are needed. The first one is $T^h : H_{\pm}^{1,h} \rightarrow H_0^1(\Omega)$. 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 triangular 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 $u^h \in H_{\pm}^{1,h}$ 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 K^+ and the other is on K^- . The location of its discontinuity in the interface cell is the straight line segment Γ_K^h . Note that two end points of the line segment are located on interface Γ , hence the interface condition $[u] = a$ could be and is enforced exactly at these two end points. In each interface cell, the interface condition $[\beta \nabla u \cdot n] = b$ is enforced with value b at the middle point of Γ_K^h . Similar versions of such extension can be found in literature [5, 7]. In order to use this extension, we need the following theorem.

Theorem 2.1. *For all $u^h \in H_{\pm}^{1,h}$, $U^h(u^h)$ can be constructed uniquely, provided T^h, ϕ, a and b are given.*

Proof. There are five different cases under consideration:

Case 0. The cell $K_{1,2,3}$ is a regular triangle, see Fig. 2.2. When calculating, we assume that all vertices of the triangular belong to Ω^+ or Ω^- .

Suppose vertices 1, 2, 3 in Fig. 2.2 belong to Ω^- , then $K_{1,2,3} \subset \Omega^-$.

$$U^h(u^h) = u(x_1, y_1) + u_x^-(x - x_1) + u_y^-(y - y_1), x, y \in K_{1,2,3},$$

u_x^-, u_y^- can be written in the following form

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

Case 1. The interface comes across one vertex of the cell $K_{1,2,3}$, see Fig. 2.3. When calculating,

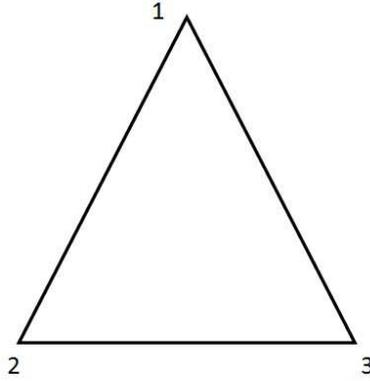


Fig. 2.2. Regular triangle.

we need to check whether one vertex is on the interface or not. If it is on the interface, we assume this vertex belongs to Ω^+ , otherwise it depends on which domain it belongs to.

Suppose vertices 2, 3 in Fig. 2.3 belong to Ω^- , $K_{1,2,3} \subset \Omega^-$.

$$U^h(u^h) = u(x_2, y_2) + u_x^-(x - x_2) + u_y^-(y - y_2), x, y \in K_{1,2,3},$$

u_x^-, u_y^- can be written in the following form

$$\begin{aligned} u_x^- &= c_{x,1}^- u^+(1) + c_{x,2}^- u^-(2) + c_{x,3}^- u^-(3) + c_{x,4}^- a(1), \\ u_y^- &= c_{y,1}^- u^+(1) + c_{y,2}^- u^-(2) + c_{y,3}^- u^-(3) + c_{y,4}^- a(1). \end{aligned}$$

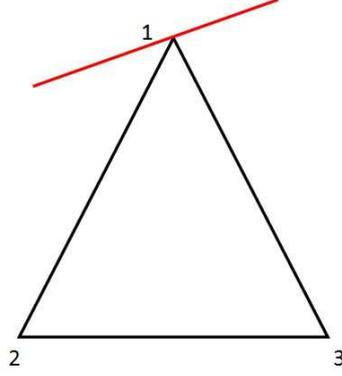


Fig. 2.3. The interface comes across one vertex of the triangle.

Case 2. The interface covers one edge of the cell $K_{1,2,3}$, see Fig. 2.4. When calculating, we also need to check whether one vertex is on the interface or not. If it is on the interface, we assume this vertex belongs to Ω^+ , otherwise it depends on which domain it belongs to. The difference between Case 1 and Case 2 is that when calculating the integration, Case 1 does not need to calculate the line integration, but Case 2 does.

Suppose vertex 2 in Fig. 2.4 belongs to Ω^- , $K_{1,2,3} \subset \Omega^-$.

$$U^h(u^h) = u(x_2, y_2) + u_x^-(x - x_2) + u_y^-(y - y_2), x, y \in K_{1,2,3},$$

u_x^-, u_y^- can be written in the following form

$$\begin{aligned} u_x^- &= c_{x,1}^- u^+(1) + c_{x,2}^- u^-(2) + c_{x,3}^- u^+(3) + c_{x,4}^- a(1) + c_{x,5}^- a(3) + c_{x,6}^- b(6), \\ u_y^- &= c_{y,1}^- u^+(1) + c_{y,2}^- u^-(2) + c_{y,3}^- u^+(3) + c_{y,4}^- a(1) + c_{y,5}^- a(3) + c_{y,6}^- b(6). \end{aligned}$$

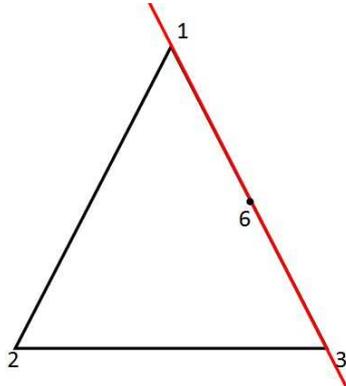


Fig. 2.4. The interface covers one edge of the triangle.

Case 3. The interface comes across one vertex and cuts one edge of the cell $K_{1,2,3} = K^+ \cup K^-$, see Fig. 2.5. When calculating, we also need to check whether one vertex is on the interface or not. If it is on the interface, we assume this vertex belongs to Ω^+ , else are depended on which domain they belong to.

In Fig. 2.5, suppose vertex 1 belongs to Ω^+ , vertex 3 belongs to Ω^- , then $K_{1,2,5} \subset \Omega^+$, $K_{2,3,5} \subset \Omega^-$.

$$U^h(u^h) = \begin{cases} u(x_1, y_1) + u_x^+(x - x_1) + u_y^+(y - y_1), x, y \in K_{1,2,5}, \\ u(x_3, y_3) + u_x^-(x - x_3) + u_y^-(y - y_3), x, y \in K_{2,3,5}. \end{cases}$$

The value $u(5)^\pm$ of vertex 5 can be denoted as a linear function of $u^+(1)$, $u^+(2)$, $u^-(3)$,

$$\begin{aligned} u(5)^+ &= c_1^+ u(1)^+ + c_2^+ u(2)^+ + c_3^+ u^-(3) + c_4^+ a(2) + c_5^+ a(5) + c_6^+ b(6), \\ u(5)^- &= c_1^- u(1)^+ + c_2^- u(2)^+ + c_3^- u^-(3) + c_4^- a(2) + c_5^- a(5) + c_6^- b(6). \end{aligned}$$

Hence u_x^-, u_y^-, u_x^+ and u_y^+ can be written in the following form

$$\begin{aligned} u_x^+ &= c_{x,1}^+ u^+(1) + c_{x,2}^+ u^+(2) + c_{x,3}^+ u^-(3) + c_{x,4}^+ a(2) + c_{x,5}^+ a(5) + c_{x,6}^+ b(6), \\ u_y^+ &= c_{y,1}^+ u^+(1) + c_{y,2}^+ u^+(2) + c_{y,3}^+ u^-(3) + c_{y,4}^+ a(2) + c_{y,5}^+ a(5) + c_{y,6}^+ b(6), \\ u_x^- &= c_{x,1}^- u^+(1) + c_{x,2}^- u^+(2) + c_{x,3}^- u^-(3) + c_{x,4}^- a(2) + c_{x,5}^- a(5) + c_{x,6}^- b(6), \\ u_y^- &= c_{y,1}^- u^+(1) + c_{y,2}^- u^+(2) + c_{y,3}^- u^-(3) + c_{y,4}^- a(2) + c_{y,5}^- a(5) + c_{y,6}^- b(6). \end{aligned}$$

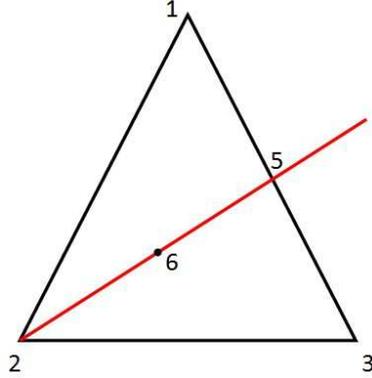


Fig. 2.5. The interface comes across one vertex and one edge of the triangle.

Case 4. The interface cuts two edges of the cell $K_{1,2,3} = K^+ \cup K^-$, see Fig. 2.6. When calculating, we need to check which domain one vertex belongs to.

In Fig. 2.6, suppose vertex 1 belongs to Ω^+ , vertices 2,3 belongs to Ω^- , then $K_{1,4,5} \subset \Omega^+$, $K_{2,3,5} \subset \Omega^-$.

$$U^h(u^h) = \begin{cases} u(x_1, y_1) + u_x^+(x - x_1) + u_y^+(y - y_1), x, y \in K_{1,2,5}, \\ u(x_3, y_3) + u_x^-(x - x_3) + u_y^-(y - y_3), x, y \in K_{2,3,5}. \end{cases}$$

The value $u(4)^\pm$ and $u(5)^\pm$ can be denoted as a linear function of $u^+(1)$, $u^-(2)$, $u^-(3)$,

$$\begin{aligned} u(4)^+ &= c_{4,1}^+ u^+(1) + c_{4,2}^+ u^-(2) + c_{4,3}^+ u^-(3) + c_{4,4}^+ a(4) + c_{4,5}^+ a(5) + c_{4,6}^+ b(6), \\ u(4)^- &= c_{4,1}^- u^+(1) + c_{4,2}^- u^-(2) + c_{4,3}^- u^-(3) + c_{4,4}^- a(4) + c_{4,5}^- a(5) + c_{4,6}^- b(6), \\ u(5)^+ &= c_{5,1}^+ u^+(1) + c_{5,2}^+ u^-(2) + c_{5,3}^+ u^-(3) + c_{5,4}^+ a(4) + c_{5,5}^+ a(5) + c_{5,6}^+ b(6), \\ u(5)^- &= c_{5,1}^- u^+(1) + c_{5,2}^- u^-(2) + c_{5,3}^- u^-(3) + c_{5,4}^- a(4) + c_{5,5}^- a(5) + c_{5,6}^- b(6). \end{aligned}$$

Hence u_x^-, u_y^-, u_x^+ and u_y^+ can be written in the following form

$$\begin{aligned} u_x^+ &= c_{x,1}^+ u^+(1) + c_{x,2}^+ u^-(2) + c_{x,3}^+ u^-(3) + c_{x,4}^+ a(2) + c_{x,5}^+ a(5) + c_{x,6}^+ b(6), \\ u_y^+ &= c_{y,1}^+ u^+(1) + c_{y,2}^+ u^-(2) + c_{y,3}^+ u^-(3) + c_{y,4}^+ a(2) + c_{y,5}^+ a(5) + c_{y,6}^+ b(6), \\ u_x^- &= c_{x,1}^- u^+(1) + c_{x,2}^- u^-(2) + c_{x,3}^- u^-(3) + c_{x,4}^- a(2) + c_{x,5}^- a(5) + c_{x,6}^- b(6), \\ u_y^- &= c_{y,1}^- u^+(1) + c_{y,2}^- u^-(2) + c_{y,3}^- u^-(3) + c_{y,4}^- a(2) + c_{y,5}^- a(5) + c_{y,6}^- b(6). \end{aligned}$$

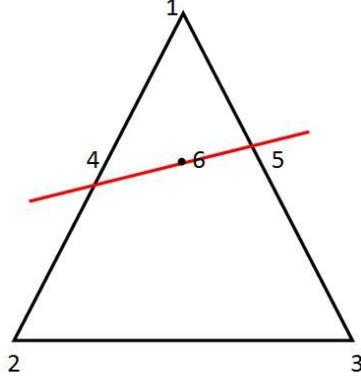


Fig. 2.6. The interface cuts two edges of the triangle.

To complete the proof of Theorem 3.1, we need the following lemma:

Lemma 2.1. *All coefficients c in Theorem 3.1 are finite and independent of u^h, a and b .*

Proof. See the proof of Lemma 3.1 in [23]. □

Based on the above discussion, we propose the following method:

Method 2.1 Find a discrete function $u^h \in H_{\pm}^{1,h}$ such that $u^h = g^h$ on boundary points so that for all $\psi^h \in H_0^{1,h}$, we have

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

To implement the above method, we use the Gaussian quadrature rule for integrals. The idea is illustrated in Fig. 2.7. If T is separated into two pieces by the interface $\overline{u_4 u_5}$, we connect u_3 and u_4 , then we get three triangles: $T_1 = \triangle u_1 u_4 u_5$, $T_2 = \triangle u_2 u_3 u_4$, $T_3 = \triangle u_3 u_4 u_5$. For each triangle, label the center point p_{ij} of each edge $\overline{u_i u_j}$. In numerical computation, apply the average of three $f(p_{ij})$ in each triangle. Numerical results show the improvement over [2], where fewer sample points were used.

When the value of β^{\pm} are only defined on the grid points, we have coefficient matrix instead of analytic function, which means we do not have any value defined on the interface. In this case, we need to calculate the value of β^+ on Ω^+ and β^- on Ω^- . See Fig. 2.8. Since point p is on the interface, we need to use extrapolation. To get the value of point p for β^+ where the interface come across the \triangle_{432} , first choose a point (x, y) , where $x = \min\{x_4, x_3, x_2\} - dx$ and $y = \min\{y_4, y_3, y_2\} - dy$, dx, dy are the grid size, in Fig. 2.8, (x, y) is exactly point 1. Second,

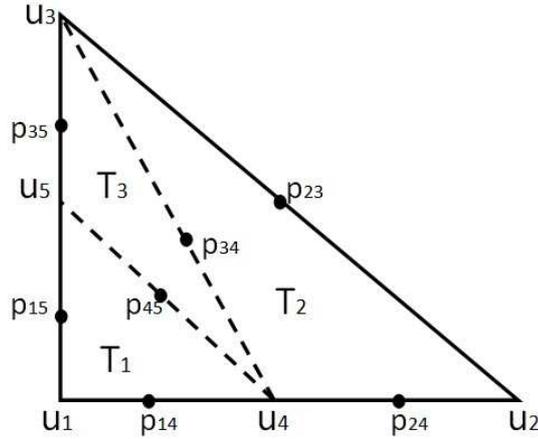


Fig. 2.7. Gaussian quadrature rule for integrals.

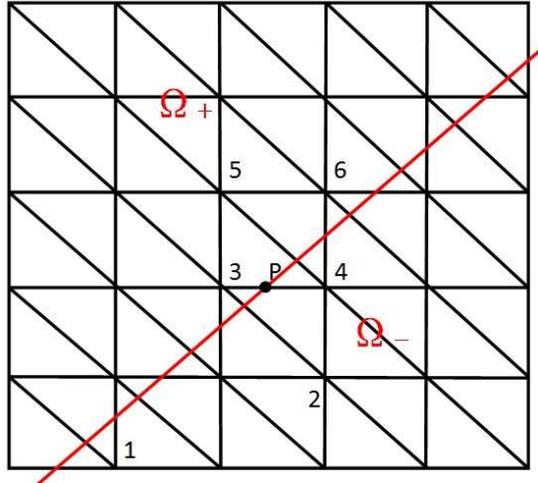


Fig. 2.8. β Extrapolation.

choose three points that are nearest to point p of Ω^+ , which satisfies $x_1 \leq x \leq x_1 + 3dx$ and $y_1 \leq y \leq y_1 + 3dy$ and are not on the same line, for easy implementation, we just check 16 points around it. In Fig. 2.8 they are points 3, 5, 6. At last, use linear extrapolation to get the value of β^+ on point p : $\beta_p = a\beta_3 + b\beta_5 + c\beta_6$.

3. Numerical Experiments

In all numerical experiments below, the level-set function $\phi(x, y)$, the coefficients $\beta^\pm(x, y)$ and the solutions

$$\begin{aligned} u &= u^+(x, y), \text{ in } \Omega^+, \\ u &= u^-(x, y), \text{ in } \Omega^-, \end{aligned}$$

are given. Hence

$$f = -\nabla \cdot (\beta \nabla u),$$

$$\begin{aligned} a &= u^+ - u^-, \\ b &= (\beta^+ \nabla u^+) \cdot n - (\beta^- \nabla u^-) \cdot n, \end{aligned}$$

on the whole domain Ω . g is obtained as a proper Dirichlet boundary condition, since the solutions are given.

All errors in solutions are measured in the L^∞ norm in the whole domain Ω . All errors in the gradients of solutions are measured in the L^∞ norm away from interfaces.

Example 1. We use this simple example, where β is piecewise constant, to demonstrate that the condition number for the fully discretized system depends on $\frac{\beta^-}{\beta^+}$ linearly, and that the condition number grows with the order of $O(h^{-2})$, which is the same as in the case without interface. The level-set function ϕ , the coefficients β^\pm and the solution u^\pm are given as follows:

$$\begin{aligned} \phi(x, y) &= (x + 1)^2 + (y + 1)^2 - 1, \\ \beta^+(x, y) &= 1, \\ \beta^-(x, y) &= 1, 50, 100, 150, 200, 250, 300, \\ u^+(x, y) &= 6 + \sin(6\pi x) \sin(6\pi y), \\ u^-(x, y) &= \exp(x^2 + 1) + y^2. \end{aligned}$$

We plot the condition number VS the ratio between β^- and β^+ for 20-by-20 grid in Fig. 3.1. The correlation coefficient is 0.9999996, which is clearly a linear relation. Table 3.1 shows the error on different grids.

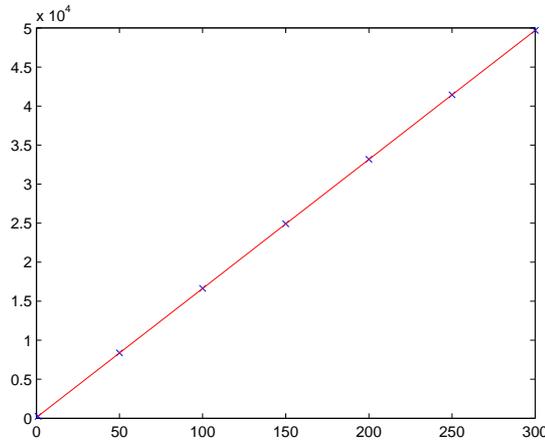


Fig. 3.1. Condition number VS $\frac{\beta^-}{\beta^+}$.

Example 2. The level-set function ϕ , the coefficients β^\pm and the solution u^\pm are given as follows:

$$\begin{aligned} \phi(x, y) &= x + y, & \beta^+(x, y) &= 1, \\ \beta^-(x, y) &= 2 + \sin(x + y), & u^+(x, y) &= 8, \\ u^-(x, y) &= x^2 + y^2 + \sin(x + y). \end{aligned}$$

Table 3.1: Condition Number Growth Pattern.

β^-	1		
$n_x \times n_y$	Err in u^h	Order	Condition Number
20×20	0.40		2.35e+002
40×40	0.092	2.10	9.43e+002
80×80	0.023	2.02	3.77e+003
160×160	0.0057	2.00	1.51e+004
β^-	50		
$n_x \times n_y$	Err in u^h	Order	Condition Number
20×20	0.42		8.38e+003
40×40	0.21	1.02	3.35e+004
80×80	0.068	1.61	1.34e+005
160×160	0.014	2.24	5.37e+005
β^-	100		
$n_x \times n_y$	Err in u^h	Order	Condition Number
20×20	0.57		1.66e+004
40×40	0.33	0.79	6.67e+004
80×80	0.11	1.58	2.27e+005
160×160	0.021	2.39	1.07e+006
β^-	150		
$n_x \times n_y$	Err in u^h	Order	Condition Number
20×20	0.75		2.49e+004
40×40	0.43	0.82	9.97e+004
80×80	0.15	1.55	3.99e+005
160×160	0.028	2.38	1.60e+006
β^-	200		
$n_x \times n_y$	Err in u^h	Order	Condition Number
20×20	0.98		3.32e+004
40×40	0.51	0.94	1.33e+005
80×80	0.18	1.54	5.31e+005
160×160	0.035	2.35	2.13e+006
β^-	250		
$n_x \times n_y$	Err in u^h	Order	Condition Number
20×20	1.19		4.14e+004
40×40	0.59	1.02	1.66e+005
80×80	0.20	1.54	6.64e+005
160×160	0.040	2.32	2.66e+006
β^-	300		
$n_x \times n_y$	Err in u^h	Order	Condition Number
20×20	1.48		4.97e+004
40×40	0.66	1.16	1.99e+005
80×80	0.23	1.54	7.96e+005
160×160	0.046	2.29	3.19e+006

This example illustrates the case when the interface is on the hypotenuse of a triangle. Fig. 3.2 shows the numerical solution with our method using 40 grid points in both x and y directions. Table 3.2 shows the error on different grids.

Example 3. The level-set function ϕ , the coefficients β^\pm and the solution u^\pm are given as

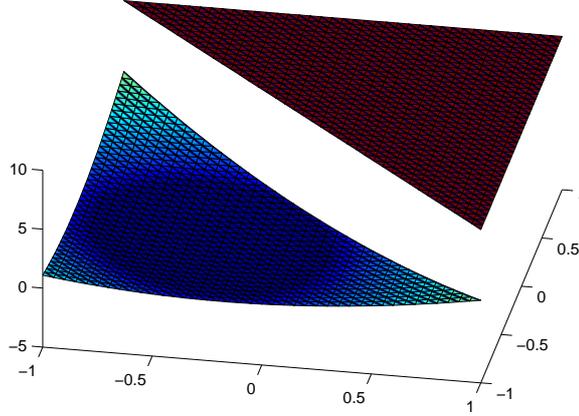


Fig. 3.2. Interface lies on the Hypotenuse of a Triangle.

Table 3.2: Numerical Results when Interface lies on the Hypotenuse of a Triangle.

Method $n_x \times n_y$	Method in [23]		New Method	
	Err in u^h	Order	Err in u^h	Order
20×20	1.13e-4		1.13e-4	
40×40	2.82e-5	2.00	2.82e-5	2.00
80×80	7.04e-6	2.00	7.05e-6	2.00
160×160	1.76e-6	2.00	1.76e-6	2.00

follows:

$$\begin{aligned}
 \phi(x, y) &= x, \quad x + y > 0, \\
 \phi(x, y) &= -y, \quad x + y \leq 0, \\
 \beta^+(x, y) &= 1, \\
 \beta^-(x, y) &= 2 + \sin(x + y), \\
 u^+(x, y) &= 6 + \sin(6\pi x) \sin(6\pi y), \\
 u^-(x, y) &= x^2 + y^2 + \sin(x + y).
 \end{aligned}$$

This example illustrates the case when the interface lies on one leg of a triangle. Fig. 3.3 shows the numerical solution with our method using 40 grid points in both x and y directions. Table 3.3 shows the error on different grids.

Table 3.3: Numerical Results when Interface lies on one Leg of a Triangle.

Method $n_x \times n_y$	Method in [23]		New Method	
	Err in u^h	Order	Err in u^h	Order
20×20	0.42		0.43	
40×40	0.14	1.63	0.14	1.63
80×80	0.044	1.64	0.045	1.63
160×160	0.013	1.71	0.014	1.71

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

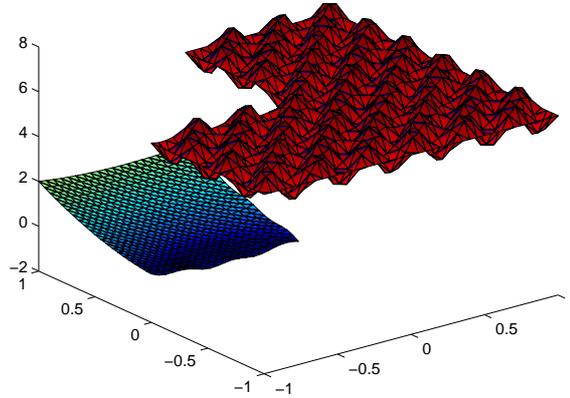


Fig. 3.3. Interface lies on one Leg of a Triangle.

follows:

$$\begin{aligned}\phi(x, y) &= (x + 1)^2 + (y + 1)^2 - 1, \\ \beta^+(x, y) &= 1 + x^2, \quad \beta^-(x, y) = 2 + y^2, \\ u^+(x, y) &= xy^2, \quad u^-(x, y) = \exp(x^2 + 1) + y^3.\end{aligned}$$

This example illustrates the case when the interface cuts one vertex of a triangle. Fig. 3.4 shows the numerical solution with our method using 40 grid points in both x and y directions. Table 3.4 shows the error on different grids.

Table 3.4: Numerical Results when Interface cuts one Vertex of a Triangle.

Method $n_x \times n_y$	Method in [23]		New Method	
	Err in u^h	Order	Err in u^h	Order
20 × 20	0.011		0.011	
40 × 40	0.0027	2.01	0.0027	2.01
80 × 80	6.65e-4	2.00	6.67e-4	2.00
160 × 160	1.63e-4	2.03	1.63e-4	2.03

The following examples are defined on $[0, 1] \times [0, 1]$.

Example 5. This example has the Neumann data $\nabla u \cdot n = 0$ on the boundary $\partial\Omega$. The level-set function ϕ , the coefficients β^\pm and the solution u^\pm are given as follows:

$$\begin{aligned}\phi(x, y) &= x^2 + y^2 - 0.25, \\ \beta^+(x, y) &= 1 + x^2, \quad \beta^-(x, y) = 1 + y^2, \\ u^+(x, y) &= (x^3/3 - x^2/2 + 3)(y^3/3 - y^2/2 - 5), \\ u^-(x, y) &= \cos(\pi x) \cos(\pi y).\end{aligned}$$

The interface intersects with the boundary of the domain. Fig. 3.5 shows the numerical solution with our method using 40 grid points in both x and y directions. Table 3.5 shows the error on different grids.

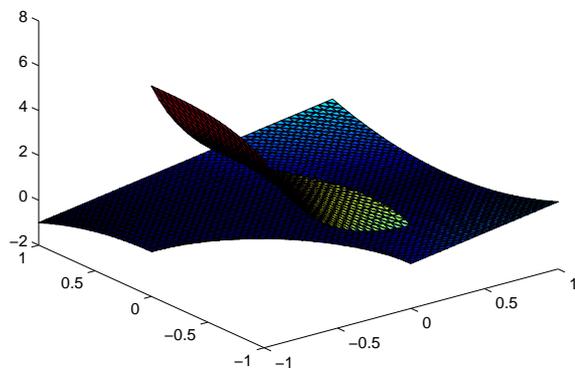


Fig. 3.4. Interface cuts one Vertex of a Triangle.

Table 3.5: Numerical Results when Interface intersects with the Boundary of the Domain with Neumann Boundary Condition.

$n_x \times n_y$	Err in u^h	Order
20×20	0.0090	
40×40	0.0027	1.72
80×80	8.06e-4	1.76
160×160	2.37e-4	1.77

Example 6. This example has the Neumann data $\nabla u \cdot n = 0$ on the boundary $\partial\Omega$. The level-set function ϕ , the coefficients β^\pm and the solution u^\pm are given as follows:

$$\begin{aligned} \phi(x, y) &= 0.15 - ((x - 0.5)^2 + (y - 0.5)^2), \\ \beta^+(x, y) &= 1 + x^2, \quad \beta^-(x, y) = 1 + y^2, \\ u^+(x, y) &= x^3 + 2y^2, \quad u^-(x, y) = \cos(\pi x) \cos(\pi y). \end{aligned}$$

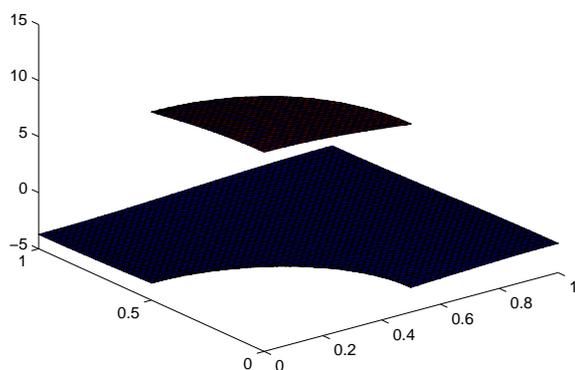


Fig. 3.5. Interface intersects with the Boundary of the Domain with Neumann Boundary Condition.

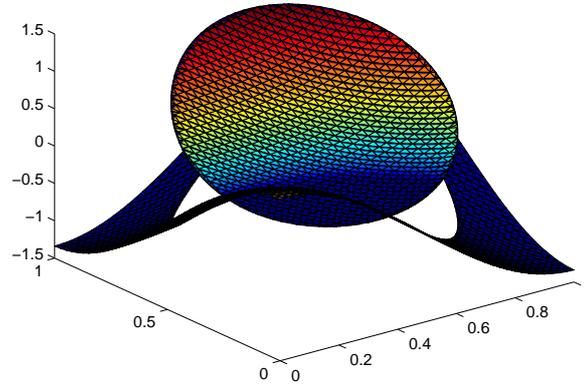


Fig. 3.6. Interface is inside the Domain with Neumann Boundary Condition.

Table 3.6: Numerical Results when Interface is inside the Domain with Neumann Boundary Condition.

$n_x \times n_y$	Err in u^h	Order
20×20	0.0080	
40×40	0.0024	1.71
80×80	7.05e-4	1.78
160×160	2.05e-4	1.78

The interface is inside the domain. Fig. 3.6 shows the numerical solution with our method using 40 grid points in both x and y directions. Table 3.6 shows the error on different grids.

In physical world, it often happens that we only have the value of β^\pm and ϕ on the grid points instead of the real function. In this situation, we use distance matrix to denote ϕ , use linear extrapolation to calculate the value of β^\pm on the interface. The following examples will show the numerical result of such cases.

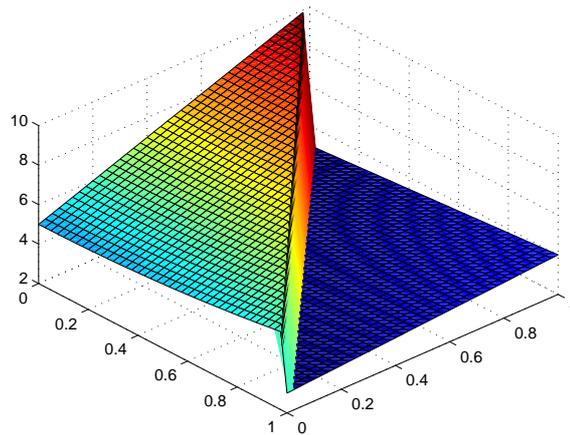


Fig. 3.7. The Coefficients β^\pm (Discrete Data) when Interface is a Line.

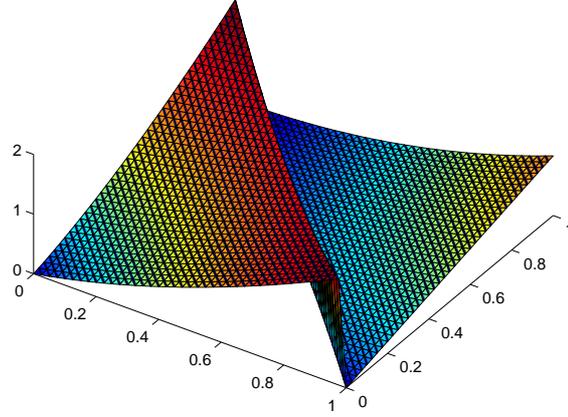


Fig. 3.8. The Numerical Results when Interface is a Line with Discrete Coefficient Data.

Example 7. The solution u^\pm are given as follows:

$$\begin{aligned} u^+(x, y) &= x^2 y, \\ u^-(x, y) &= x^2 + y^2 + \sin(x + y). \end{aligned}$$

Case 1. When the coefficients β^\pm is shown in Fig. 3.7 and the distance matrix is obtained from the level-set function: $\phi(x, y) = x + y - 1$, Fig. 3.8 shows the numerical solution with our method using 40 grid points in both x and y directions. Table 3.7 shows the error on different grids.

Table 3.7: The Numerical Results when Interface is a Line with Discrete Coefficient Data.

$n_x \times n_y$	Err in u^h	Order
20×20	4.22e-5	
40×40	1.06e-5	2.00
80×80	2.64e-6	2.00
160×160	6.60e-7	2.00

Case 2. When the coefficients β^\pm on the domain is shown in Fig. 3.9 and the distance matrix is obtained from the level-set function:

$$\begin{aligned} \phi(r, \theta) &= \frac{R \sin(\theta_t/2)}{\sin(\theta_t/2 + \theta - \theta_r - 2\pi(i-1)/5)} - r, \\ \theta_r + \pi(2i-2)/5 &\leq \theta < \theta_r + \pi(2i-1)/5, \end{aligned}$$

Table 3.8: Numerical Results when Interface is a corner of a Star with Discrete Coefficient Data.

$n_x \times n_y$	Err in u^h	Order
20×20	0.0011	
40×40	2.77e-4	2.02
80×80	7.68e-5	1.85
160×160	2.15e-5	1.83

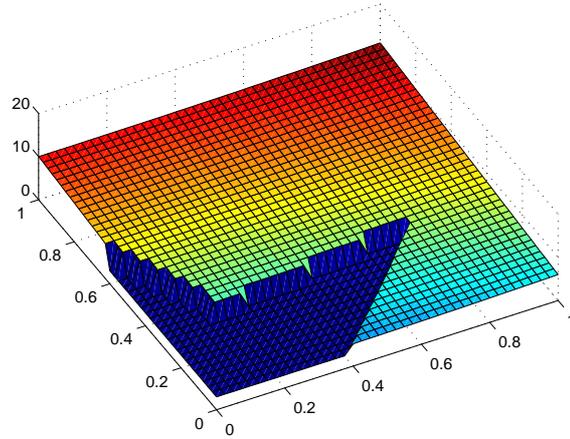


Fig. 3.9. The Coefficients β^\pm (Discrete Data) when Interface is a corner of a Star.

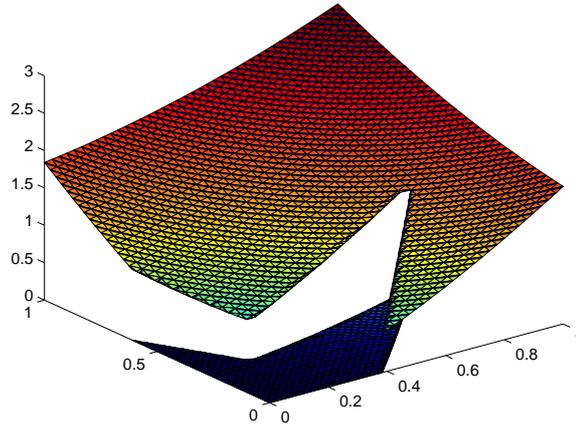


Fig. 3.10. The Numerical Results when Interface is a corner of a Star with Discrete Coefficient Data.

$$\phi(r, \theta) = \frac{R \sin(\theta_t/2)}{\sin(\theta_t/2 - \theta + \theta_r - 2\pi(i-1)/5)} - r,$$

$$\theta_r + \pi(2i-3)/5 \leq \theta < \theta_r + \pi(2i-2)/5,$$

with

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

Fig. 3.10 shows the numerical solution with our method using 40 grid points in both x and y directions. Table 3.8 shows the error on different grids.

4. Conclusion

We proposed a non-traditional finite element method for solving the elliptic interface problem with Dirichlet boundary condition or Neumann boundary condition. All the possible ways

the interface cuts the grid are considered. If β is positive definite and lower order terms are not present, the resulting linear system is (non-symmetric) positive definite. The coefficient matrix data can only be given at grid points, not as an analytic function, making our method more practical. We used 7 examples to show that our method has second order accuracy in L^∞ norm, and the condition number of the coefficient matrix grows with order $O(h^{-2})$, which is the same as the case without interface.

Acknowledgments. L. Wang's research is supported by Science Foundation of China University of Petroleum-Beijing (No.YJRC-2013-48). S. Hou's research is supported by NSF grant DMS-1317994.

References

- [1] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher, A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.*, **152**:2 (1999), 457-492.
- [2] S. Hou and X. Liu, A numerical method for solving variable coefficient elliptic equations with interfaces. *J. Comput. Phys.*, **202** (2005), 411-445.
- [3] R.J. LeVeque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.*, **31** (1994), 1019-1044.
- [4] Z. Li, A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, **35**:1 (1998), 230-254.
- [5] Z. Li, T. Lin, and X. Wu, New cartesian grid methods for interface problems using the finite element formulation. *Numer. Math.*, **96**:1 (2003), 61-98.
- [6] Z. Li and K. Ito, The immersed interface method: Numerical solutions of pdes involving interfaces and irregular domains, SIAM, Philadelphia, 2006.
- [7] X.-D. Liu, R. P. Fedkiw, and Myungjoo Kang, A boundary condition capturing method for Poisson's equation on irregular domains. *J. Comput. Phys.*, **160**:1 (2000), 151-178.
- [8] X.-D. Liu and T. Sideris, Convergence of the ghost fluid method for elliptic equations with interfaces. *Math. Comp.*, **72** (2003).
- [9] A. Mayo, The fast solution of Poisson's and biharmonic equations in irregular domains. *SIAM J. Numer. Anal.*, **21**:2 (1984), 285-299.
- [10] A. Mayo, Fast high order accurate solutions of Laplace's equation on irregular domains. *SIAM J. Sci. Stat. Comput.*, **6**:1 (1985), 144-157.
- [11] C. Peskin, Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, **25** (1977), 220-252.
- [12] C. Peskin and B. Printz, Improved volume conservation in the computation of flows with immersed elastic boundaries. *J. Comput. Phys.*, **105** (1993), 33-46.
- [13] M. Sussman, P. Smereka, and S. Osher, A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, **114** (1994), 146-154.
- [14] J. W.L. Wan and X.-D. Liu, A boundary condition capturing multigrid approach to irregular boundary problems. *SIAM J. Sci. Comput.*, **25**:6 (2004), 1982-2003.
- [15] W.-J. Ying and C.S. Henriquez, A kernel-free boundary integral method for elliptic boundary value problems. *J. Comput. Phys.*, **227**:2 (2007), 1046-1074.
- [16] S. Yu, Y. Zhou, and G.W. Wei, Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces. *J. Comput. Phys.*, **224** (2007), 729-756.
- [17] Z.-M. Chen and J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems. *Numer. Math.*, **79** (1998), 175-202.
- [18] 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. *J. Comput. Phys.*, **213** (2006), 1-30.

- [19] P. Colella and H. Johansen, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comput. Phys.*, **60** (1998), 85-147.
- [20] M. Oevermann, and R. Klein, A Cartesian grid finite volume method for elliptic equations with variable coefficients and embedded interfaces. *J. Comput. Phys.*, **219** (2006), 749-769.
- [21] M. Oevermann, C. Scharfenberg, and R. Klein, A sharp interface finite volume method for elliptic equations on Cartesian grids. *J. Comput. Phys.*, **228** (2009), 5184-5206.
- [22] P. Macklin and J. S. Lowengrub, A New Ghost Cell / Level Set Method for Moving Boundary Problems: Application to Tumor Growth. *J. Sci. Comput.*, **35** (2008), 266-299.
- [23] S. Hou, W. Wang, L. Wang, Numerical method for solving matrix coefficient elliptic equation with sharp-edged interfaces. *J. Comput. Phys.*, **229** (2010), 7162-7179.
- [24] S. Hou, Z. Li, L. Wang, and W. Wang, A numerical method for solving elasticity equations with interfaces. *Comm. Comput. Phys.*, (2012), 595-612.
- [25] S. Hou, L. Wang, and W. Wang, A numerical method for solving the elliptic interface problems with multi-domains and triple junction points. *J. Comput. Math.*, **30**:5 (2012), 504-516.
- [26] S. Hou, P. Song, L. Wang, and H. Zhao, A weak formulation for solving elliptic interface problems without body fitted grid. *J. Comput. Phys.*, **249** (2013), 80-95.
- [27] 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.