Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2016-0155

Vol. **22**, No. 4, pp. 965-996
October 2017

A Second Order Ghost Fluid Method for an Interface Problem of the Poisson Equation

Cheng Liu* and Changhong Hu

Research Institute for Applied Mechanics, Kyushu University, Fukuoka 816-0811, Japan.

Received 14 September 2016; Accepted (in revised version) 7 February 2017

Abstract. A second order Ghost Fluid method is proposed for the treatment of interface problems of elliptic equations with discontinuous coefficients. By appropriate use of auxiliary virtual points, physical jump conditions are enforced at the interface. The signed distance function is used for the implicit description of irregular domain. With the additional unknowns, high order approximation considering the discontinuity can be built. To avoid the ill-conditioned matrix, the interpolation stencils are selected adaptively to balance the accuracy and the numerical stability. Additional equations containing the jump restrictions are assembled with the original discretized algebraic equations to form a new sparse linear system. Several Krylov iterative solvers are tested for the newly derived linear system. The results of a series of 1-D, 2-D tests show that the proposed method possesses second order accuracy in L^∞ norm. Besides, the method can be extended to the 3-D problems straightforwardly. Numerical results reveal the present method is highly efficient and robust in dealing with the interface problems of elliptic equations.

AMS subject classifications: 65Z05, 76M20

Key words: Interface problem, ghost fluid method, Poisson equation, jump conditions.

1 Introduction

The elliptic equation with a discontinuous physical field across the irregular interface appears in many applications such as diffusion phenomenon, heat transfer, crystal growth and many others. For fluid dynamic problems, the method used for treating irregular interface can be extended in solving the incompressible NavierStokes equations. For instance, without adding source terms, the effects of surface tension can be considered in the pressure Poisson equation straightforwardly.

*Corresponding author. *Email addresses:* liu@riam.kyushu-u.ac.jp (C. Liu), hu@riam.kyushu-u.ac.jp (C. Hu)

 ©2017 Global-Science Press

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

To solve an elliptic equation with irregular interfaces, a body-fitted grid can be applied [1]. Unfortunately, generating mesh that fits the boundaries of the computational domain with complex internal geometries is time-consuming, often requiring manual intervention to modify and cleaning-up the geometry. On the other hand, we can use the finite element method [2] with more flexible unstructured mesh to model the complex boundary. However, it may require a huge amount of time for the regeneration or deformation of the computational grid when the corresponding interfaces are changing. The Cartesian grid method can be generated automatically and efficient in handling the complex geometries with simplified data structure and formulations. One difficulty in using the Cartesian grid method exists in how to impose the jump conditions implicitly at the grid points adjacent to the interface without losing accuracy.

The IB (immersed boundary) method is a type of Cartesian grid method first proposed by Peskin [3] for the simulation of human heart. The direct forcing approach was latter proposed by Mohd-Yusof [4]. However, high order immersed boundary methods are restricted to certain type of boundary conditions. To ensure numerical stability, the traditional IB method uses Heaviside function to smooth the jumps of the diffusive coefficient which may bring out unexpected smearing around the interface. With the application of generalized Taylor expansions, the original IIM (Immersed Interface Method) [5,6] adaptively modifies the stencil to obtain the $\mathcal{O}(h)$ truncation error along the interface. For smooth coefficients, this reduces to the standard 5-point finite difference stencil. IIM is a second order numerical method to preserve the jump conditions at the interface. Compared with the original second-order IIM [5], a newly developed IIM [7] achieves arbitrarily high-order accuracy with a wider set of grid stencils. However, this algorithm is fairly complex and result in a non-symmetric and not diagonally dominated system.

The Ghost Fluid Method (GFM) was proposed by R. Fedkiw et al. [8] to properly treat the boundary conditions across the interface. The GFM creates an artificial fluid to implicitly enforce proper conditions. Motivated by the original GFM, Liu et al. [9] introduce fictitious points along coordinates to enforce the jump conditions properly. Although a symmetric positive definite linear system is derived, the tangential flux $[\beta\nabla u\cdot\tau]$ in determining the fictitious contribution is neglected, which results in only first order accuracy. The MIB (Matched Interface and Boundary) method [10] was then proposed to account for a non-zero $[\beta\nabla u\cdot\tau]$ by differentiating the given jump conditions using one-sided interpolations. This treatment widens the stencil in several directions that depend on the local geometry, and results in a non-symmetric discretization. The MIB method was extended by Zhou et al. [11] to handle high curvature geometry, and by Yu et al. [12] to provide a 3D version MIB method.

Some researchers use the fictitious points to enforce the Dirichlet or Neumann type boundary conditions as well as the jump conditions for the immersed interface boundary. The method presented by Johansen et al. [13] achieves second order accuracy and preserves jumps at the interface. However, this method only handling the Dirichlet type boundary conditions. In the method proposed by Cisternino et al. [14], additional unknowns are introduced to allow straightforward expression of the interface transmission

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

conditions. Gibou et al. [15] develop a second-order Cartesian grid method for imposing the Dirichlet type boundary conditions on an irregular domain. The value at the ghost nodes is assigned by linear extrapolation, and the whole discretization leads to a symmetric linear system, which can be solved by a preconditioned conjugate gradient method. Recently, the Lagrange multiplier approach [16,17] is applied to enforce jump conditions, numerical experiments indicate second order accuracy in L^∞ can be obtained.

We propose an improved GFM for approximation of the discontinuous physical field across the interfaces. In previous GFM method [9], only first order accuracy is preserved, while numerical tests reveals the present method reaches second order overall accuracy. Another feature of the present method is the additional unknowns are introduced for the enforcement of the jump conditions. Additional equations that contain the jump restrictions are assembled to form a new sparse linear systems of algebraic equations. Although the newly derived matrix is unsymmetrical and standard fast linear solvers cannot be utilized directly, the solution procedure can be highly efficient with a preconditioned Krylov iterative solver. It should be noted that the symmetric positive definite matrix can only be preserved when the elliptic equations are discretized in uniform grid manner. For the more general cases, in which the stretched mesh are used or solving a multi-phase flow problem, the matrix will lose the symmetric property even without introducing any jump condition. Although the matrix derived from the present method cannot satisfy the symmetrical and positive features, all the iterative solvers we tested always yield the solution with favorable stopping error.

The rest of this paper is organized as follows. In Section 2, the main ideas of the proposed second order method are described by using a one dimensional example. The scheme for a general two dimensional problem is developed based on the same ideas in Section 3. The strategies for linear and quadratic interpolation as well as the method of enforcing the jump conditions are described in Sections 3.1 and 3.2, respectively. The three dimensional problem is briefly discussed in Section 4. Section 5 gives extensive numerical tests and validations of the proposed method in solving one, two and three dimensional elliptic interface problems with regular or irregular boundaries. A conclusion is given at the end Section.

2 One dimensional problem

Consider an elliptic problem with variable coefficients in one dimension

$$(\beta u_x)_x = f(x) \tag{2.1}$$

with Dirichlet boundary conditions defined on $\partial\Omega$. The second order finite difference discretization of Eq. (2.1) follows

$$\left[\beta_{i+\frac{1}{2}}\left(\frac{u_{i+1}-u_i}{\Delta x}\right)-\beta_{i-\frac{1}{2}}\left(\frac{u_i-u_{i-1}}{\Delta x}\right)\right]\Big/\Delta x = f_i, \quad (i=1,2,\cdots,n). \tag{2.2}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

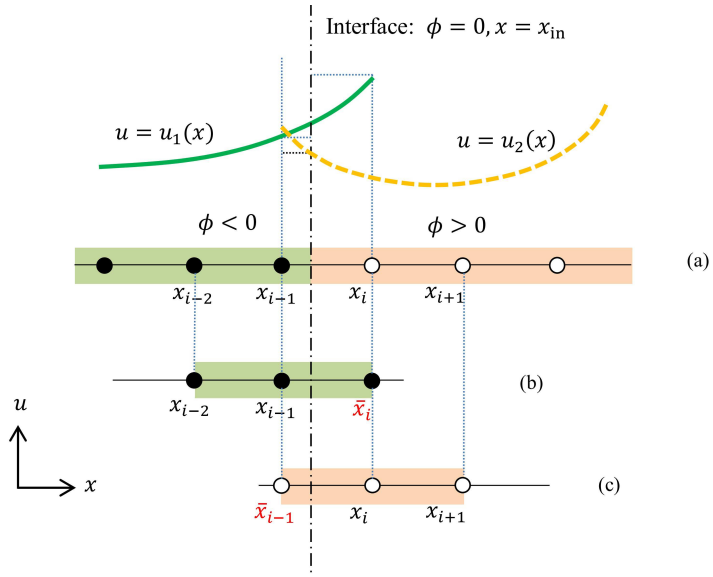


Figure 1: One dimensional interpolation scheme for jump conditions.

If no discontinuous boundary conditions are considered, a simple linear sparse system can be generated from Eq. (2.2). The $n \times n$ tri-diagonal matrices can be solved by Thomas algorithm efficiently. A second order convergence rate for spatial accuracy is expected. For a problem with irregular interfaces, however, it is necessary to consider an efficient sharp interface method to enforce the jump conditions of $[u]$ and $[\beta u_n]$ with additional equations. Here $u_n = \nabla u \cdot \mathbf{N}$ is the normal derivative of u. To maintain the second order overall accuracy of the discretized formulation, a new treatment is proposed in which the jump conditions are enforced with the additional unknowns. The discontinuity of $\beta(x)$ exists at the interface and the smearing of coefficient function $\beta(x)$ can be avoided by applying the Ghost Fluid Method [9].

Fig. 1 shows the stencils used in discretization of one dimensional elliptic problem. In present study, the signed distance function $\phi = sign(\min(\|x - x_{in}\|))$ is defined on the grid and $\phi = 0$ is correspondent to the immersed interface. As shown in Fig. 1(a), the points that belong to the left side $(\phi < 0)$ of the interface are indicated by the solid circles, while the hollow circles are used to represent the right side points $(\phi > 0)$. The profiles of scalar field u is shown in Fig. 1(a) with jump conditions $[u] = \delta$ and $[\beta u_n] = \varepsilon$ defined at the interface $\phi = 0$, respectively. In the original GFM, the interpolation stencils in each subdomain ($\phi < 0$ and $\phi > 0$) are not overlapped and only first order accuracy can be obtained. To enforce the two types of jump conditions with second order accuracy, we follow the basic idea in GFM to extend the stencil points to involve one of the points in its opposite's domain, as described in Figs. 1(b) and (c). This field extension creates two additional points, $\bar{\mathbf{x}}_i$ in the original position of $\mathbf{x}_i$ and $\bar{\mathbf{x}}_{i-1}$ in the position of $\mathbf{x}_{i-1}$. The two additional stencils $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_{i-1}$ will be utilized for the enforcement of jump conditions.

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

Finally, u_{i-2}, u_{i-1} and $\bar{u}_i$ can be used for constructing a quadratic formulation of $u=u_1(x)$ and $\bar{u}_{i-1}$, u_i, u_{i+1} can be used for constructing $u=u_2(x)$. The second order approximations for $u_1(x)$, $u_2(x)$ are as follows:

$$u_k(x)=a_kx^2+b_kx+c_k,\quad k=1,2. \tag{2.3}$$

For example, if the uniform grid is used, the undetermined coefficients of $u_1(x)$ can be directly calculated as

$$a_1=\frac{\bar{u}_i-2u_{i-1}+u_{i-2}}{2dx^2},\quad b_1=\frac{\bar{u}_i-u_{i-2}}{2dx},\quad c_1=u_{i-1},$$

where dx is the grid space. Similarly, coefficients for $u_2(x)$ are obtained as

$$a_2=\frac{u_{i+1}-2u_i+\bar{u}_{i-1}}{2dx^2},\quad b_2=\frac{4u_i-u_{i+1}-3\bar{u}_{i-1}}{2dx},\quad c_2=\bar{u}_{i-1}.$$

Considering $x_{in}=(1-\theta)dx$, $\theta=((x_i-x_{in}))/((x_i-x_{i-1}))$, the jump condition for u can be written as

$$\begin{aligned}[u]=&u_1(x_{in})-u_2(x_{in})\\ =&(a_1-a_2){x_{in}}^2+(b_1-b_2)x_{in}+(c_1-c_2)\\ =&\frac{\theta(\theta-1)}{2}u_{i-2}+\frac{-\theta^2+2\theta+1}{2}u_{i-1}+\frac{\theta^2+2\theta-3}{2}u_i\\ &+\frac{\theta(1-\theta)}{2}u_{i+1}+\frac{\theta(\theta-1)}{2}\bar{u}_{i-1}+\frac{\theta^2-3\theta+2}{2}\bar{u}_i.\end{aligned} \tag{2.4}$$

The jump condition for u_n leads to

$$\begin{aligned}[\beta u_n]=&\beta_1\partial u_1x(x_{in})/\partial x-\beta_2\partial u_2x(x_{in})/\partial x\\ =&\frac{2\beta_1(1-\theta)-\beta_2}{2dx}u_{i-2}+\frac{2\beta_1(\theta-1)}{dx}u_{i-1}+\frac{2\beta_1(1-\theta)-2\beta_2}{dx}u_i\\ &+\frac{2\beta_1(\theta-1)+\beta_2}{2dx}u_{i+1}+\frac{2\beta_1(\theta-1)+3\beta_2}{2dx}\bar{u}_{i-1}+\frac{2\beta_1(\theta-1)+\beta_2}{2dx}\bar{u}_i.\end{aligned} \tag{2.5}$$

With the two additional equations (Eq. (2.4) and Eq. (2.5)), the new matrices can be obtained which contains two additional unknowns $\bar{u}_{i-1}$ and $\bar{u}_i$.

We notice that for point x_{i-1}, the discretization of Poisson equation should be written as

$$\left[\beta_{i-\frac{1}{2}}\left(\frac{\bar{u}_i-u_{i-1}}{\Delta x}\right)-\beta_{i-\frac{3}{2}}\left(\frac{u_i-u_{i-2}}{\Delta x}\right)\right]\Big/\Delta x=f_{i-1}.$$

Similarly, for point x_i, the equation follows

$$\left[\beta_{i+\frac{1}{2}}\left(\frac{u_{i+1}-u_i}{\Delta x}\right)-\beta_{i-\frac{1}{2}}\left(\frac{u_i-\bar{u}_{i-1}}{\Delta x}\right)\right]\Big/\Delta x=f_i.$$

Finally, the $(n+2)\times(n+2)$ sparse matrix systems are assembled, which can be solved by a Krylov subspace iteration method, such as preconditioned BICGSTAB or GMRES method.

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

3 Two dimensional problem

The two dimensional problem considered here is a Cartesian grid based elliptic problem as follows,

$$\begin{aligned} &\nabla\cdot(\beta\nabla u)=f \qquad \text{on } \Omega=\Omega_1\cup\Omega_2,\\ &[u]=\delta \qquad\qquad \text{on } \Gamma,\\ &[\beta u_n]=\varepsilon \qquad\qquad \text{on } \Gamma. \end{aligned}$$

Ω contains two separated sub domains Ω_1 and Ω_2 that is divided by the irregular interface Γ. The discretization of the elliptic operator at ordinary grid point (i,j) follows

$$\begin{aligned} &\left[\beta_{i+\frac{1}{2},j}\frac{u_{i+1,j}-u_{i,j}}{\Delta x}-\beta_{i-\frac{1}{2},j}\frac{u_{i,j}-u_{i-1,j}}{\Delta x}\right]\Big/\Delta x+\left[\beta_{i,j+\frac{1}{2}}\frac{u_{i,j+1}-u_{i,j}}{\Delta y}-\beta_{i,j-\frac{1}{2}}\frac{u_{i,j}-u_{i,j-1}}{\Delta y}\right]\Big/\Delta y\\ &=f_{i,j},\quad (i=1,2,\cdots nx;\quad j=1,2,\cdots ny). \end{aligned} \tag{3.1}$$

For the domain without irregular interface, a five-diagonal sparse matrix will be generated by Eq. (3.1). By introducing appropriate boundary conditions, the derived linear system can be solved easily.

We follow the basic idea described in Section 2 in dealing with jump conditions for present 2-D problem. By introducing additional unknowns, the discontinuous conditions can be enforced at the points adjacent to the interface. Although the approach in Section 2 can be extended to the higher dimensional case, there are still significant differences in details, for instance the strategy in determining the interpolation stencils etc.

3.1 Linear interpolation method

The classification of the grid points has to be done at the beginning. As shown in Fig. 2, the irregular interface indicated by the brown solid line divides the whole domain into two regions, domain-I ($\phi>0$) and II ($\phi<0$). The virtual points marked with the hollow circles and squares are defined at the position that belongs to one domain and have at least one neighbor in another domain. These points will be used for imposing the constraint equations of $[u]$ and $[\beta u_n]$. We refer to the set of additional virtual vertices that belong to domain-I as $\aleph^I$, and the additional vertices that belong to domain-II as $\aleph^{II}$. Once the virtual points are determined, the additional unknowns can be defined at the corresponding position. Fig. 3(a) shows the points $\mathbf{x}_{i,j}$ and the adjacent interface. With the definition of the signed distance function $\phi_{i,j}$, many geometrical computations can be simplified. The normal vector of the body surface in a forcing point is derived by

$$\mathbf{n}_{i,j}=\frac{\nabla\phi_{i,j}}{\|\nabla\phi_{i,j}\|}. \tag{3.2}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

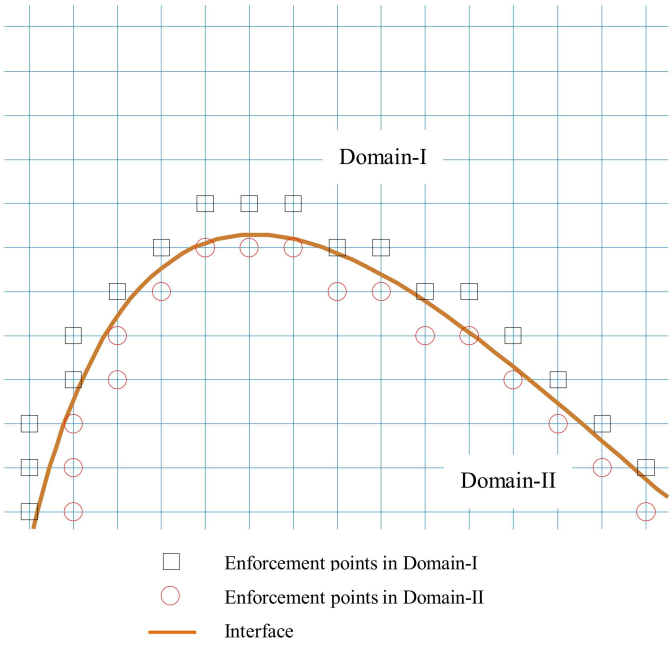


Figure 2: Virtual points around the interface.

A central difference method is used to calculate $\mathbf{n}_{i,j}$. The points $\mathbf{x}_{IN1}$,$\mathbf{x}_{IN2}$ on the interface with minimum distance to $\mathbf{x}_{i,j}$ are then obtained by

$$\mathbf{x}_{IN1}=\mathbf{x}_{i,j}-\mathbf{n}_{i,j}\phi_{i,j},\quad \mathbf{x}_{IN2}=\mathbf{x}_{i,j-1}-\mathbf{n}_{i,j-1}\phi_{i,j-1}. \tag{3.3}$$

If there is no discontinuous variation across the interface, the $u_{i,j}$, $u_{i-1,j}$, $u_{i,j-1}$ can be utilized for the interpolation of u at $\mathbf{x}_{IN1}$. For present interface problems, in order to reconstruct the jump conditions at the interface, additional virtual points are required. As shown in Fig. 3(b), take the points $\mathbf{x}_{i,j}$, $\mathbf{x}_{i,j-1}$ for instance, we follow the basic idea mentioned in Section 2. The field extension strategy is applied to extend the physical field u from domain-I to II at position $(i,j-1)$ with the additional virtual point $\bar{\mathbf{x}}_{i,j-1}$. Similarly, u field can also be extended from domain-II to domain-I with the additional virtual points $\bar{\mathbf{x}}_{i,j}$ as shown in Fig. 3(c). Once the additional unknowns are defined, the construction of the jump conditions is straightforward. The jump constraint equation can be written as:

$$[u]|_{\mathbf{x}_{IN}}=F(\bar{\mathbf{x}}_{i,j}),\qquad \bar{\mathbf{x}}_{i,j}\in\aleph^I, \tag{3.4a}$$

$$[\beta u_n]|_{\mathbf{x}_{IN}}=G(\bar{\mathbf{x}}_{i,j}),\qquad \bar{\mathbf{x}}_{i,j}\in\aleph^{II}. \tag{3.4b}$$

Here F, G can be built from interpolation. In present method, a linear variation of the scalar field is assumed in domain-I, therefore,

$$u_I=f_I(u_{i,j},u_{i-1,j},\bar{u}_{i,j-1}), \tag{3.5}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

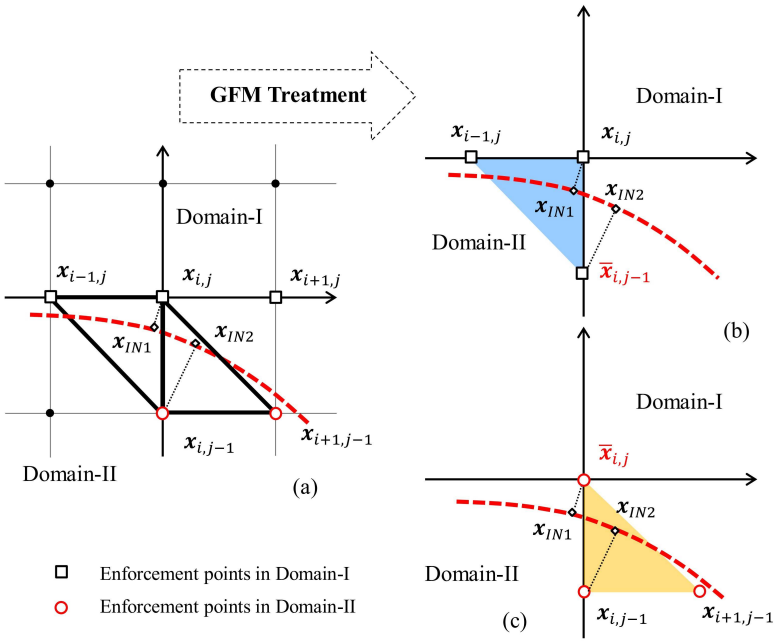


Figure 3: Classification of the points for construction of jump conditions. (a) Original enforcing points of the two domains, (b) Interpolation stencils for domain-I, (c) interpolation stencils for domain-II.

and in domain-II

$$u_{II} = f_{II}(u_{i,j-1}, \bar{u}_{i-1,j}, \bar{u}_{i,j}). \tag{3.6}$$

The jump conditions of $[u]$ can be enforced at the virtual points in domain-I (marked with hollow squares in Fig. 3(b)), while $[\beta u_n]$ can be enforced at the virtual points in domain-II (marked with hollow circles in Fig. 3(a)) and vice versa. As it has been validated in Section 5.2, the accuracy is not influenced by the sequence significantly. However, if the diffusive coefficient β is discontinuous around the interface, the effect of sequence should be considered. For simplicity, only the treatments for two virtual points $\bar{\mathbf{x}}_{i,j}$, $\bar{\mathbf{x}}_{i,j-1}$ are used for demonstration. Since two virtual points are involved, two additional constrain equations should be introduced to close the linear system. For $\bar{\mathbf{x}}_{i,j}$ (Fig. 3(b)), the jump conditions for u can be written as

$$[u]|_{\mathbf{x}_{IN1}} = (u_I - u_{II})|_{\mathbf{x}_{IN1}} = (f_I(u_{i,j}, u_{i-1,j}, \bar{u}_{i,j-1}) - f_{II}(\bar{u}_{i,j}, \bar{u}_{i-1,j}, u_{i,j-1}))|_{\mathbf{x}_{IN1}}. \tag{3.7}$$

For $\bar{\mathbf{x}}_{i,j-1}$ (Fig. 3(c)), the jump conditions for u_n is given

$$\begin{aligned}
&[\beta u_n]|_{\mathbf{x}_{IN2}} = (\beta_I u_{n_I} - \beta_{II} u_{n_{II}})|_{\mathbf{x}_{IN2}} \\
&= \Big(\frac{\partial}{\partial x} (\beta_I f_I(u_{i,j}, \bar{u}_{i,j-1}, \bar{u}_{i+1,j-1}) - \beta_{II} f_{II}(\bar{u}_{i,j}, u_{i,j-1}, u_{i+1,j-1})) N_x \\
&\quad + \frac{\partial}{\partial y} (\beta_I f_I(u_{i,j}, \bar{u}_{i,j-1}, \bar{u}_{i+1,j-1}) - \beta_{II} f_{II}(\bar{u}_{i,j}, u_{i,j-1}, u_{i+1,j-1})) N_y \Big) \Big|_{\mathbf{x}_{IN2}}.
\end{aligned} \tag{3.8}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

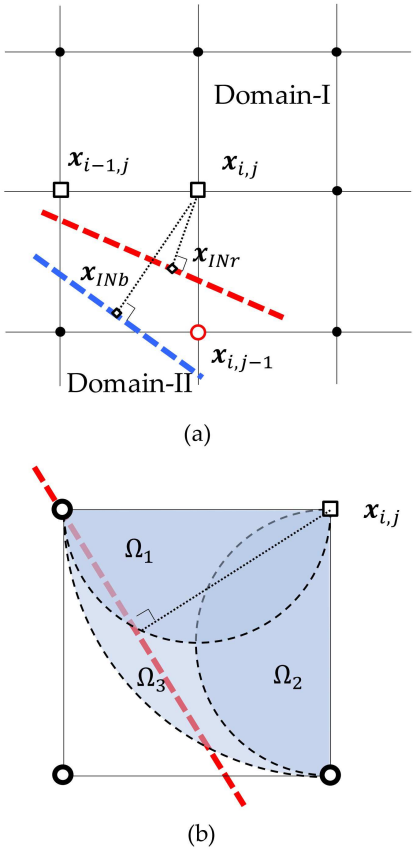


Figure 4: Interface approximated by the piecewise segments. (b). Possible region of $\mathbf{x}_{IN}$.

Eq. (3.7) and Eq. (3.8) together with Eq. (3.1) are assembled to formulate a linear system for the final solution.

Since the position of the crossing point $\mathbf{x}_{IN}$ depends on the shape of the interface, the matrix can be ill-conditioned for some extreme cases. We carry out numerical analysis and propose a modified interpolation scheme to avoid such ill-conditioned matrix. We start from the assumption that the interface can be approximated by piecewise segments as shown in Fig. 4(a). Then we need to find out the possible locations of $\mathbf{x}_{IN}$. Take the virtual point in $\mathbf{x}_{i,j}$ for instance, if the red dotted line represent the interface, the crossing point $\mathbf{x}_{INr}$ can be determined easily (Eq. (3.3)). If the interface is indicated by the blue line, since the $\mathbf{x}_{i,j}$ is no longer virtual points, the $\mathbf{x}_{INb}$ is out of consideration in this case. Generally, the possible region $\Omega = \Omega_1 \cup \Omega_2$ of $\mathbf{x}_{IN}$ is shown in Fig. 4(b). Even without piecewise assumption, considering the radius of curvature $\rho_\kappa \geq h$, (h: grid interval), the possible region of $\mathbf{x}_{IN}$ will be restricted in $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$. In the following analysis, we need to ensure that the interpolation scheme will not cause ill-conditioned matrix in region Ω (shadow in Fig. 4(b)).

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

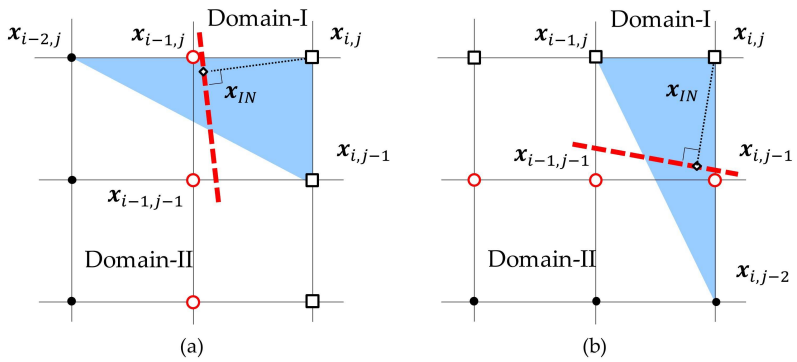


Figure 5: Two cases for the modified linear interpolation stencils.

Further analysis are required to make the linear system solvable and in well-conditioned manner. Since the constraint equations for $[u]$ and $[\beta u_n]$ are assembled to form the matrix, the property of the linear system could be easily affected by different interpolation schemes. We now discuss the constraint equations for $[u]$. As shown in Fig. 5(a), for the constraint equation at $\mathbf{x}_{i,j}$, the jump conditions at $\mathbf{x}_{IN}$ can be interpolated by the linear approximation of the common and virtual points $\bar{\mathbf{x}}_{i-1,j}$, $\mathbf{x}_{i,j-1}$, $\mathbf{x}_{i,j}$, $\mathbf{x}_{i-1,j}$, $\bar{\mathbf{x}}_{i,j-1}$ and $\bar{\mathbf{x}}_{i,j}$. Therefore,

$$\begin{aligned}[u]|_{\mathbf{x}_{IN}} =& (u_I - u_{II})|_{\mathbf{x}_{IN}} = (f_I(\bar{u}_{i-1,j}, u_{i,j-1}, u_{i,j}) - f_{II}(u_{i-1,j}, \bar{u}_{i,j-1}, \bar{u}_{i,j}))|_{\mathbf{x}_{IN}} \\ =& C_{diag}\bar{u}_{i,j} + \sum_{neighbours} C_{ix,jx}u_{ix,jx} + \sum_{neighbours} C_{ix,jx}\bar{u}_{ix,jx},\end{aligned} \tag{3.9}$$

where ix, jx indicate the subscripts of the neighbors for $\mathbf{x}_{i,j}$. The first term in Eq. (3.9) means the diagonal part of the matrix, the second and the third term stand for the contributions of the ordinary and the additional ghost fluid neighbors, respectively. If $\mathbf{x}_{IN}$ is very close to $\mathbf{x}_{i-1,j}$, as shown in Fig. 5(a), the weight of $u_{i-1,j}$ in Eq. (3.9) will be much larger than the weight of diagonal variable $\bar{u}_{i,j}$. An ill matrix will be generated by assembling a constraint equation of this case. Similar situation will happen when $\mathbf{x}_{IN}$ getting close to $\mathbf{x}_{i,j-1}$, see Fig. 5(b). However, this problem can be addressed by leaving $\mathbf{x}_{i-1,j}$ and substitute it by the point $\mathbf{x}_{i-2,j}$ to build the interpolation scheme, as shown in the blue triangle of Fig. 5(a). In this case,

$$u_{II} = f_{II}(u_{i-2,j}, \bar{u}_{i,j-1}, \bar{u}_{i,j}). \tag{3.10}$$

By this modification, the ill-conditioned matrix can be avoided. Similar strategy can be applied for the case of Fig. 5(b) in which

$$u_{II} = f_{II}(u_{i,j-2}, \bar{u}_{i-1,j}, \bar{u}_{i,j}). \tag{3.11}$$

However, this treatment causes a question on how to select the stencils for interpolation. Facilitating discussion, we named the scheme mentioned in Eq. (3.6) as $Stencil-1$,

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

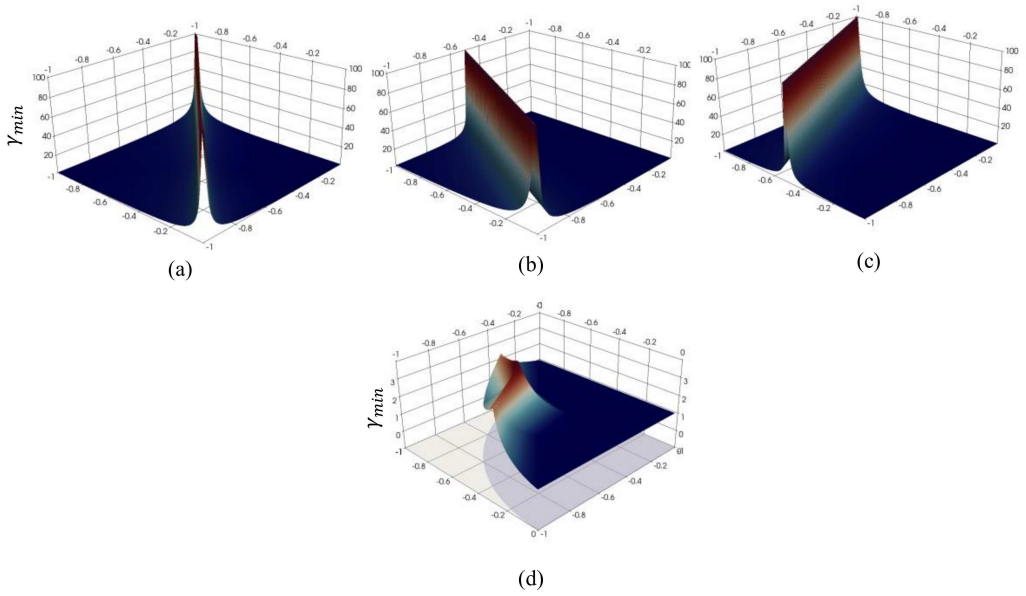


Figure 6: Profile of $\gamma_{Stencil1}$ (a), $\gamma_{Stencil2}$ (b), $\gamma_{Stencil3}$ (c), $\gamma_{\min}$ (d).

Eq. (3.10) as $Stencil-2$ and Eq. (3.11) as $Stencil-3$. Define γ for the assessment of the three schemes:

$$\gamma_{Stencil-n}=\max\left(\frac{\max(|C_{neigbours}|)}{|C_{diag}|},1\right),\quad n=1,2,3, \tag{3.12a}$$

$$\gamma_{\min}=\min(\gamma_{Stencil-1},\gamma_{Stencil-2},\gamma_{Stencil-3}). \tag{3.12b}$$

Figs. 6(a)-(c) shows the distribution of $\gamma_{Stencil-1,2,3}$ for $x,y\in[-1,0]$. For $Stencil-1$, the $|C_{neighbours}|/|C_{diag}|\gg 1$ when the crossing point $\mathbf{x}_{IN}$ is getting close to the line of $(\mathbf{x}_{i-1,j},\mathbf{x}_{i,j-1})$, see Figs. 6(a). Similar situation can be observed for $Stencil-2,3$. The additional constraint equation of $[u]$ will make the solutions unstable with respect to small changes in data. However, from Figs. 6(d), it is seen $\gamma_{\min}<4.0$ when $\mathbf{x}_{IN}\in\Omega$. Therefore, we deduced that the ill-conditioned matrix can be avoided by selecting the three types of schemes according to the position of $\mathbf{x}_{IN}$.

For the constraint equations of $[\beta u_n]$ imposed at $\bar{\mathbf{x}}_{i,j}\in\aleph^I$ as shown in Fig. 5(a), the jump conditions in $\mathbf{x}_{IN}$ are given

$$\begin{aligned}&[\beta u_n]|_{\mathbf{x}_{IN}}=(\beta_I u_{n_I}-\beta_{II}u_{n_{II}})|_{\mathbf{x}_{IN}}\\&=\Big(\frac{\partial}{\partial x}(\beta_I f_I(\bar{u}_{i-1,j},u_{i,j-1},u_{i,j})-\beta_{II}f_{II}(u_{i-1,j},\bar{u}_{i,j-1},\bar{u}_{i,j}))N_x\\&\quad+\frac{\partial}{\partial y}(\beta_I f_I(\bar{u}_{i-1,j},u_{i,j-1},u_{i,j})-\beta_{II}f_{II}(u_{i-1,j},\bar{u}_{i,j-1},\bar{u}_{i,j}))N_y\Big)\Big|_{\mathbf{x}_{IN}}\end{aligned}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

$$=\beta_{II}C_{diag}\bar{u}_{i,j}+\sum_{neighbours}\beta_{I,II}C_{ix,jx}u_{ix,jx}+\sum_{neighbours}\beta_{I,II}C_{ix,jx}\bar{u}_{ix,jx}, \tag{3.13a}$$

$$\gamma_{Stencil-n}=\max\Big(\frac{\max(|\beta_{I,II}C_{neigbours}|)}{|\beta_{II}C_{diag}|},1\Big), \qquad n=1,2,3. \tag{3.13b}$$

The definition of γ is shown in Eq. (3.13b). Without considering the variation of β, we have the distribution of $\gamma_{Stencil-1}=1$ for $\mathbf{x}_{IN}\in\Omega$. When $\beta_{II}\gg\beta_I$, the diagonal coefficient $|\beta_{II}C_{diag}|$ is also dominated which is beneficial for solving the linear system. On the other hand, for $\beta_{II}\ll\beta_I$, to avoid the ill-conditioned matrix, the Eq. (3.4) will be substituted by Eq. (3.14)

$$[u]|_{\mathbf{x}_{IN}}=F(\bar{\mathbf{x}}_{i,j}), \qquad \bar{\mathbf{x}}_{i,j}\in\aleph^{II}, \tag{3.14a}$$

$$[\beta u_n]|_{\mathbf{x}_{IN}}=G(\bar{\mathbf{x}}_{i,j}), \qquad \bar{\mathbf{x}}_{i,j}\in\aleph^{I}. \tag{3.14b}$$

In general, the interpolation scheme is selected adaptively to balance the accuracy of the interpolation and the numerical stability. Numerical tests in Section 5 reveals the robustness and accuracy of the present adaptive approach. However, the linear interpolation possesses only first order overall accuracy. It is mainly used for demonstrating the implementation procedure for two dimensional problems, details of the quadratic method are presented as follows.

3.2 Quadratic interpolation method

In terms of the second order method, the classification of the points is the same as in Section 3.1. According to the numerical tests, the overall accuracy and the matrix property depends heavily on the approximation stencils. For constructing the equations containing the discontinuity, an appropriate quadratic interpolation scheme should possess three features. First, the scheme should be built by using as few virtual points as possible. Second, the newly derived linear systems can be closed even additional unknowns are introduced. Third, the condition number associated with the linear system should not be too large. To achieve the goals, a compact scheme that uses an overlapped region with the minimal vertices requirement is applied, as shown in Figs. 7(a)-(b). If we take $\mathbf{x}_{i,j}$ as the origin, $\mathbf{x}_{IN}$ is in the third quadrant, in which two cases (Case-1, Case-2) are involved according to the shape and the position of the interface. In Fig. 7(a), three vertices are virtual points are used while in Fig. 7(b) four virtual points are involved. Other cases can be obtained by rotating the frame in Fig. 7.

Take Case-1 (Fig. 7 (a)) for instance, the u in polygon A is approximated by the second order polynomial:

$$u(x,y)=C_1x^2+C_2y^2+C_3xy+C_4x+C_5y+C_6. \tag{3.15}$$

The undetermined coefficients C_n can be expressed by the linear combination of $\mathbf{u}$. Eq. (3.15) can be written as

$$u(\mathbf{x}_{ix,jx})=f_I(\bar{u}_{i,j},u_{i-1,j},u_{i,j-1},u_{i-1,j-1},u_{i-2,j-1},u_{i-1,j-2}). \tag{3.16}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

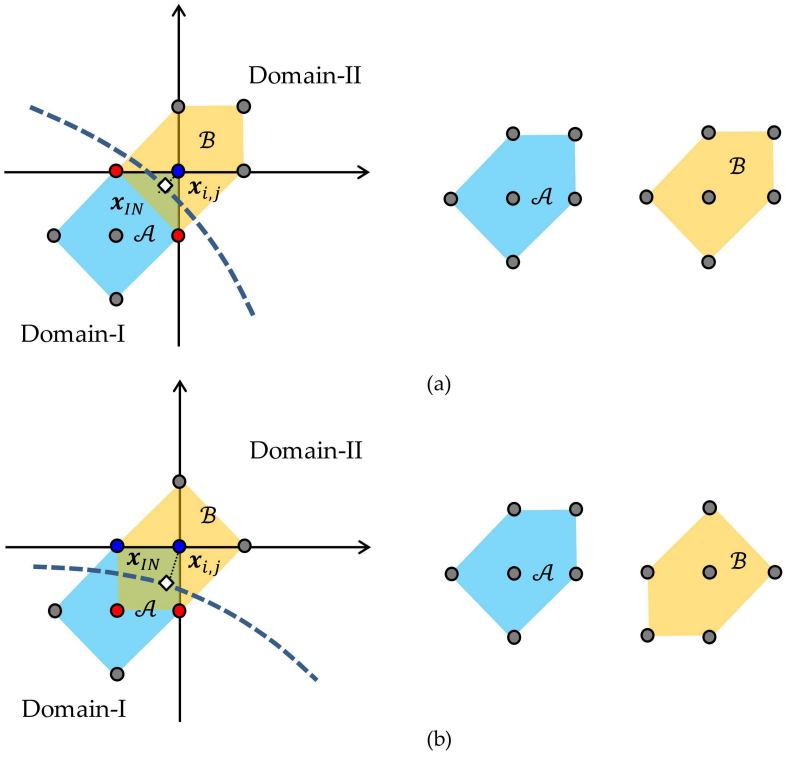


Figure 7: Two cases for the quadratic method (a) Case-1. (b) Case-2.

The following equation is derived with Eq. (3.16),

$$\mathbf{u}=\mathbf{A}\mathbf{c},$$

where

$$\mathbf{A}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ (-h)^2 & 0 & 0 & -h & 0 & 1 \\ 0 & (-h)^2 & 0 & 0 & -h & 1 \\ (-h)^2 & (-h)^2 & (-h)^2 & -h & -h & 1 \\ (-2h)^2 & (-h)^2 & 2h^2 & -2h & -h & 1 \\ (-h)^2 & (-2h)^2 & 2h^2 & -h & -2h & 1 \end{bmatrix},$$

$$\mathbf{c}=\begin{bmatrix} C_1 & C_2 & C_3 & C_4 & C_5 & C_6 \end{bmatrix}^T,$$

$$\mathbf{u}=\begin{bmatrix} \bar{u}_{i,j} & u_{i-1,j} & u_{i,j-1} & u_{i-1,j-1} & u_{i-2,j-1} & u_{i-1,j-2}, \end{bmatrix}^T,$$

where h is the grid intervals in $x-$ and $y-$ directions. The coefficients vector $\mathbf{c}$ can be derived by

$$\mathbf{c}=\mathbf{A}^{-1}\mathbf{u}, \tag{3.17}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

here $\mathbf{c}$ is a symbolic vector. For $\mathbf{x}_{IN}=(x_0,y_0)$, with the normal vectors $\mathbf{N}=(N_x,N_y)$ defined at $\mathbf{x}_{IN}$, the interpolated u and u_n at $\mathbf{x}_{IN}$ can be calculated by

$$u|_{x=x_0,y=y_0}=C_1{x_0}^2+C_2{y_0}^2+C_3x_0y_0+C_4x_0+C_5y_0+C_6, \tag{3.18a}$$

$$\left[\beta_I\frac{\partial u}{\partial \mathbf{n}}\right]\Big|_{x=x_0,y=y_0}=\beta_I\left(\frac{\partial u}{\partial x}N_x+\frac{\partial u}{\partial y}N_y\right)\Big|_{x=x_0,y=y_0}$$
$$=\beta_I(2C_1x_0+C_3y_0+C_4)N_x+\beta_I(2C_2y_0+C_3x_0+C_5)N_y. \tag{3.18b}$$

Similarly, the u in polygon B is approximated by the second order polynomial:

$$u(\mathbf{x}_{ix,jx})=f_{II}(u_{i,j},\bar{u}_{i-1,j},\bar{u}_{i,j-1},u_{i,j+1},u_{i+1,j},u_{i+1,j+1}). \tag{3.19}$$

The following equation can be derived

$$\mathbf{u}=\mathbf{B}\mathbf{d}, \tag{3.20}$$

where

$$\mathbf{B}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1\\ (-h)^2 & 0 & 0 & -h & 0 & 1\\ 0 & (-h)^2 & 0 & 0 & -h & 1\\ 0 & h^2 & 0 & 0 & h & 1\\ h^2 & 0 & 0 & h & 0 & 1\\ h^2 & h^2 & h^2 & h & h & 1 \end{bmatrix},$$

$$\mathbf{d}=\begin{bmatrix} D_1 & D_2 & D_3 & D_4 & D_5 & D_6 \end{bmatrix}^T,$$

$$\mathbf{u}=\begin{bmatrix} u_{i,j} & \bar{u}_{i-1,j} & \bar{u}_{i,j-1} & u_{i+1,j} & u_{i,j+1} & u_{i+1,j+1} \end{bmatrix}^T.$$

The coefficient vectors $\mathbf{d}$ can be derived as

$$\mathbf{d}=\mathbf{B}^{-1}\mathbf{u}. \tag{3.21}$$

The interpolated u and u_n at $\mathbf{x}_{IN}$ can be calculated by

$$u|_{x=x_0,y=y_0}=D_1{x_0}^2+D_2{y_0}^2+D_3x_0y_0+D_4x_0+D_5y_0+D_6, \tag{3.22a}$$

$$\beta_{II}\frac{\partial u}{\partial \mathbf{n}}\Big|_{x=x_0,y=y_0}=\beta_{II}(2D_1x_0+D_3y_0+D_4)N_x+\beta_{II}(2D_2y_0+D_3x_0+D_5)N_y. \tag{3.22b}$$

From Eq. (3.18a) and Eq. (3.22a), the jump conditions for $[u]$ can be written as

$$[u]|_{x=x_0,y=y_0}$$
$$=(C_1-D_1){x_0}^2+(C_2-D_2){y_0}^2+(C_3-D_3)x_0y_0$$
$$+(C_4-D_4)x_0+(C_5-D_5)y_0+(C_6-D_6). \tag{3.23}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

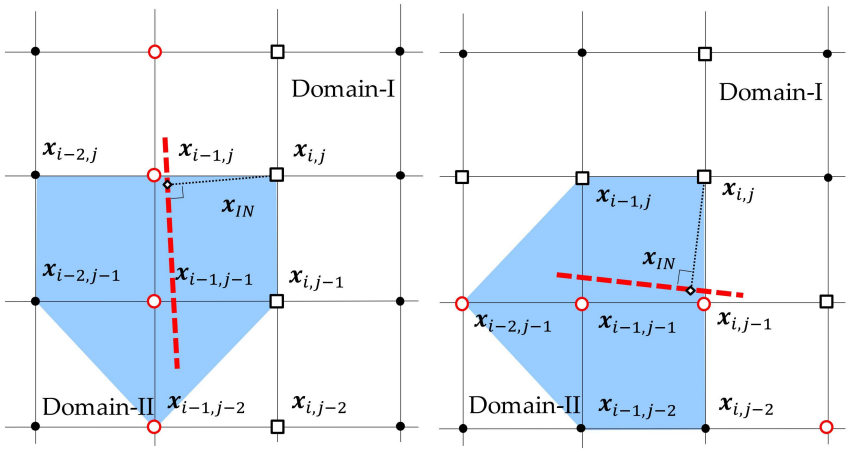


Figure 8: Two cases for the modified linear interpolation stencils.

Considering Eq. (3.17) and (3.21), the above equation can be expressed as

$$[u]|_{x=x_0,y=y_0} = F_1\bar{u}_{i,j} + F_2u_{i-1,j} + F_3u_{i,j-1} + F_4u_{i-1,j-1} + F_5u_{i-2,j-1} + F_6u_{i-1,j-2} + F_7u_{i,j} + F_8\bar{u}_{i-1,j} + F_9\bar{u}_{i,j-1} + F_{10}u_{i+1,j} + F_{11}u_{i,j+1} + F_{12}u_{i+1,j+1}. \quad (3.24)$$

Similarly, jump conditions for $[\beta u_n]$ can be written as

$$\left[\beta\frac{\partial u}{\partial \mathbf{n}}\right]\Big|_{x=x_0,y=y_0} = [2(C_1-D_1)x_0 + (C_3-D_3)y_0 + (C_4-D_4)]N_x + (2(C_2-D_2)y_0 + (C_3-D_3)x_0 + (C_5-D_5))N_y. \quad (3.25)$$

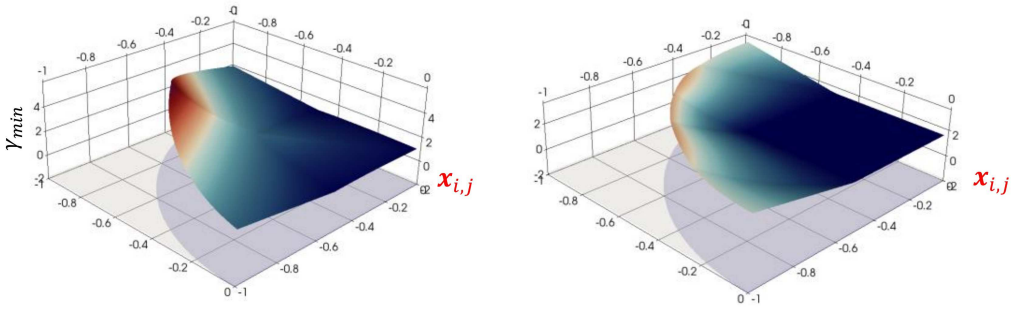
Using Eq. (3.17) and (3.21), the above equation can be expressed as

$$\left[\beta\frac{\partial u}{\partial \mathbf{n}}\right]\Big|_{x=x_0,y=y_0} = E_1\bar{u}_{i,j} + E_2u_{i-1,j} + E_3u_{i,j-1} + E_4u_{i-1,j-1} + E_5u_{i-2,j-1} + E_6u_{i-1,j-2} + E_7u_{i,j} + E_8\bar{u}_{i-1,j} + E_9\bar{u}_{i,j-1} + E_{10}u_{i+1,j} + E_{11}u_{i,j+1} + E_{12}u_{i+1,j+1}. \quad (3.26)$$

Finally, Eq. (3.24) and Eq. (3.26) should be assembled with Eq. (3.1) to form a new linear sparse system. In order to make the number of unknowns identical to the number of equations, we can enforce the jump conditions u only at the virtual points in one subdomain, and enforce the jump conditions βu_n in another subdomain and vice versa.

The property of the second order constraint equation for imposing the jump conditions are discussed following the analytical process used in Section 3. The possible active region $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ of $\mathbf{x}_{IN}$ is shown in Fig. 4(b). For instance, if the $\mathbf{x}_{IN}$ in Fig. 8(a)

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press



(a) (b)

Figure 9: Profile of $\gamma_{\min}$ (a) and $\gamma_{Stencil-1}$ (b).

is getting very close to the point $\mathbf{x}_{i-1,j}$, $\mathbf{x}_{i-1,j}$ will be excluded from the interpolation template and Eq. (3.16) should be substitute by

$$u(\mathbf{x}_{ix,jx}) = f_I(u_{i,j}, \bar{u}_{i-2,j}, u_{i,j-1}, \bar{u}_{i-2,j-1}, \bar{u}_{i-1,j-1}, \bar{u}_{i-1,j-2}). \tag{3.27}$$

Similarly, if $\mathbf{x}_{IN}$ is very closed to $\mathbf{x}_{i,j-1}$, Eq. (3.27) should be modified as

$$u(u_{ix,jx}) = f_I(u_{i,j}, u_{i-1,j}, \bar{u}_{i,j-2}, \bar{u}_{i-1,j-1}, \bar{u}_{i-2,j-1}, \bar{u}_{i-1,j-2}). \tag{3.28}$$

We named the scheme mentioned in Eq. (3.16) as $Stencil-1$, Eq. (3.27) as $Stencil-2$ and Eq. (3.28) as $Stencil-3$. Using the same strategy as in Section 3.1, the ill-conditioned matrix can be avoided by selecting among the three types of schemes according to the location of $\mathbf{x}_{IN}$. Fig. 9(a) shows $\gamma_{\min}$ derived from Eq. (3.24) and Fig. 9(b) shows $\gamma_{Stencil-1}$ for Eq. (3.18a). Note the discontinuity of diffusive coefficient β is not considered. For the case of $\beta_{II} \ll \beta_I$ or $\beta_{II} \gg \beta_I$, the same treatment given in Section 3.1 is adopted to ensure a well-conditioned matrix can be formulated finally.

4 Three dimensional problem

The approach presented in Section 3 can be extended to 3-dimensional elliptic problems without additional difficulty. The implementation of the method is organized as following

(1) Initializing the signed distance field ϕ related to the interface Γ.

(2) Classify the vertices into three groups, the normal points set $\aleph$, the virtual points set $\aleph^I$ in $domain-I$ and $\aleph^{II}$ in $domain-II$. If the linear method is applied, the scalar field u can be approximated by 4-points, for instance, for $domain-I$ in Fig. 10(a),

$$u_I = f_I(\bar{u}_{i,j,k}, u_{i-1,j,k}, u_{i,j-1,k}, u_{i,j,k-1}) = C_1 x + C_2 y + C_3 z + C_4. \tag{4.1}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

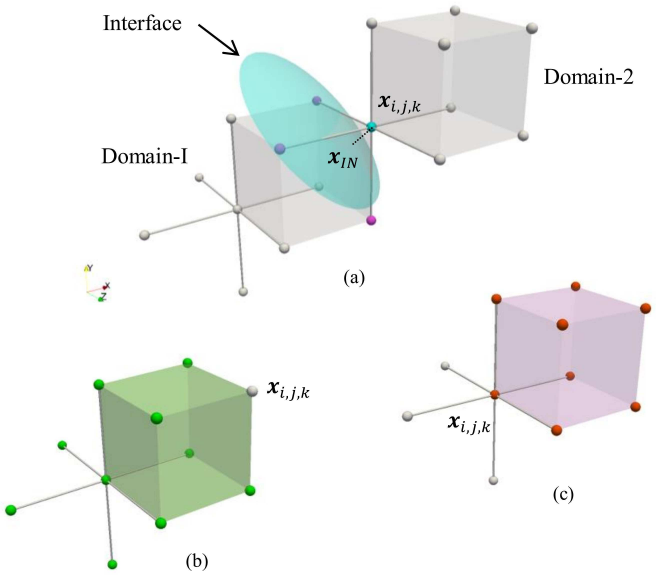


Figure 10: (a). Stencils for 3D case. (b). Interpolation template for u in $domain-I$. (c). Interpolation template for u in $domain-II$.

In terms of the second order method, 11-points approximation for u is written as

$$\begin{aligned} u_I =& f_I(\bar{u}_{i,j,k}, u_{i-1,j,k}, u_{i,j-1,k}, u_{i,j,k-1}, u_{i-1,j,k-1}, u_{i-1,j-1,k}, u_{i,j-1,k-1}, u_{i-1,j-1,k-1}, \\ & u_{i-2,j-1,k-1}, u_{i-1,j-2,k-1}, u_{i-1,j-1,k-2}) \\ =& C_1x^2+C_2y^2+C_3z^2+C_4xyz+C_5xy+C_6xz+C_7yz+C_8x+C_9y+C_{10}z+C_{11}. \end{aligned} \tag{4.2}$$

For simplification, only the implementation of the second order method is given in the following.

(3). Determine the stencils for the quadratic interpolation. The normal vector $\mathbf{N}=(N_x,N_y,N_z)$ on $\mathbf{x}_{IN}$ is utilized for searching the stencils (ix,jx,kx). As shown in Fig. 10(a), for approximating u in $domain-I$,

$$(ix,jx,kx)=\begin{cases} (i,j,k), \\ (i+\Delta_i,j,k), \\ (i,j,k+\Delta_k), \\ (i+\Delta_i,j,k+\Delta_k), \\ (i+\Delta_i,j+\Delta_j,k), \\ (i,j+\Delta_j,k+\Delta_k), \\ (i+\Delta_i,j+\Delta_j,k+\Delta_k), \\ (i+2\Delta_i,j+\Delta_j,k+\Delta_k), \\ (i+\Delta_i,j+2\Delta_j,k+\Delta_k), \\ (i+\Delta_i,j+\Delta_j,k+2\Delta_k), \end{cases} \qquad \begin{cases} \Delta_i=-sign(N_x), \\ \Delta_j=-sign(N_y), \\ \Delta_k=-sign(N_z). \end{cases} \tag{4.3}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

The first element in the vector stands for virtual point. To avoid the ill-conditioned system, a threshold of the distance between $\mathbf{x}_{IN}$ and $\mathbf{x}_{ix^n,jx^n,kx^n}$, $(n=2,\cdots,11)$ is set up. For the distance below the limitation, the point will be excluded from the interpolation template, the closed neighbor point of $\mathbf{x}_{ix^n,jx^n,kx^n}$ lies in the opposite direction of the normal vector will be adopted to build the complete quadratic approximation.

For a regular interface such as a circle or a square, the signed distance field can be initialized by the formula. For complex interface, however, direct calculation is infeasible. The method first demonstrated by J. Choi [18] and improved by Liu et al. [19] is applied for the initialization of the signed distance field ϕ. Initially, the $|\phi|$ is computed only for the cells cut by the interface. To obtain the signed distance fields in the region far from the interface, the Fast Marching Method is applied. Since the accuracy of the present method depends largely on the accuracy of the normal vectors $\mathbf{N}$, we use a large number of triangle panels to approximate the curved surface.

5 Numerical examples

5.1 Example 1

Consider $\nabla\cdot(\beta\nabla u)=f(x)$ in one dimension with $x\in[0,2]$, the inner interface is defined in $x_0=1.0$. We consider an exact solution, $u=e^x\cos(x)$ on the left side of interface $(x<1.0)$ and $u=\sin^2(x)-\cos^2(x)$ on the right side $(x\geq 1.0)$ of the interface with appropriate Dirichlet boundary conditions defined in $x=0.0$ and $x=2.0$. Besides, $\beta=1$ with $f(x)=-2e^x\sin(x)$ is defined when $x<1.0$ and $\beta=1$ with $f(x)=4(\cos^2(x)-\sin^2(x))$ is defined for $x\geq 1.0$. The jump conditions are

$$[u]=e^x x-\sin^2(x)+\cos^2(x) \quad \text{and} \quad [u_n]=e^x(\cos(x)-\sin(x))-4\sin(x)\cos(x).$$

Fig. 11 shows the numerical solution using the method in Section 2 with 161 grids. Table 1 shows the convergence rates of spatial accuracy using the first order GFM method [9] and the present method. It can be seen that the present method reaches second order accuracy both locally and globally for 1-D problems.

Table 1: Average error L^2 and maximum error L^∞.

Δx	Present GFM method				Previous GFM method [9]			
	L^∞ error	order	L^2 error	order	L^∞ error	order	L^2 error	order
1/20	2.2534E-3	-	1.0847E-3	-	5.3112E-3	-	1.9732E-3	-
1/40	5.7171E-4	1.985	2.6573E-4	2.020	2.6889E-3	0.988	9.8561E-4	1.001
1/80	1.4380E-4	1.994	6.5911E-5	2.008	1.3635E-3	0.986	4.8552E-4	1.015
1/160	3.6052E-5	1.997	1.6424E-5	2.003	6.8512E-4	0.995	2.4228E-4	1.002
1/320	9.0252E-6	1.999	4.1000E-6	2.001	3.4567E-4	0.991	1.2030E-4	1.007

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

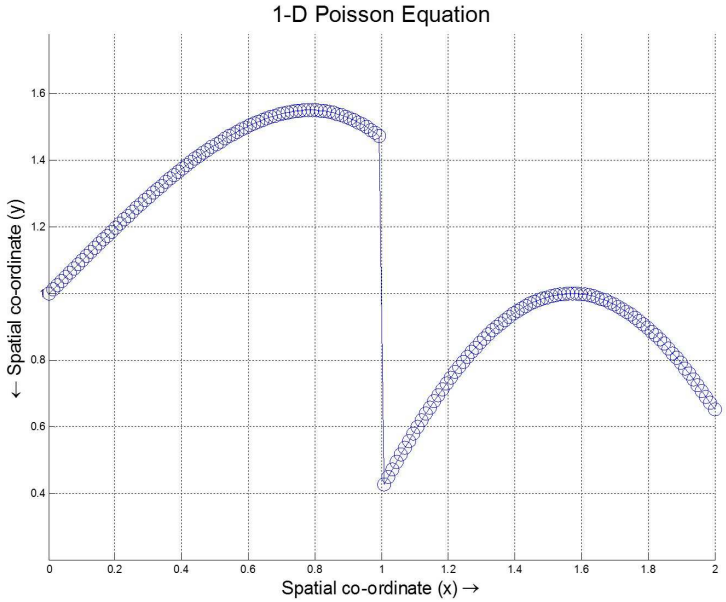


Figure 11: Solution of one dimensional elliptic problem with 80 grids (solid line indicates numerical solutions, hollow circles represent the analytical solutions).

5.2 Example 2

For the first 2-D example, the Laplace equation with an irregular interface is calculated using the present method. We consider the Laplace equation $\nabla\cdot(\nabla u)=0.0$ in two dimensions with $x\in[-1,1]$ and $y\in[-1,1]$. The shape of the interface is described by $x^2+y^2=r^2$ with outer pointed normal vectors $\mathbf{N}=(2x,2y)$ and $r=0.5$. With the definition of Dirichlet boundary conditions, the exact solution outside the interface is $u=1+\ln(2(x^2+y^2)^{0.5})$, while for the rest of domain $u=1.0$. The jump conditions at the interface are $[u]=0.0$ and $[u_n]=2.0$.

Fig. 12(a) show the local convergence rates L^∞ of the spatial accuracy. Obviously, the quadratic method shows higher accuracy in both global and local spatial tests. Besides, the two types of quadratic method discussed in Section 3.2 reach second order accuracy with the convergence rate of the order $r1=1.96$ for $Quad-1$ and $r2=2.03$ for $Quad-2$. Notice that from Fig. 12(b), the absolute differences of the two quadratic methods are small. It can be concluded that exchanging of enforcing points sequence in outer and inner domains does not affect the accuracy significantly. Moreover, the error of present second order method reduces to only 1% of the first order GFM method with the increasing of the grid resolution. Fig. 13 shows the numerical solution with 96 grids in two directions. The Localization of the error for the linear and quadratic methods is given in Fig. 14. As seen in Table 2, the present method shows much less local and global error than the previous GFM method [9].

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

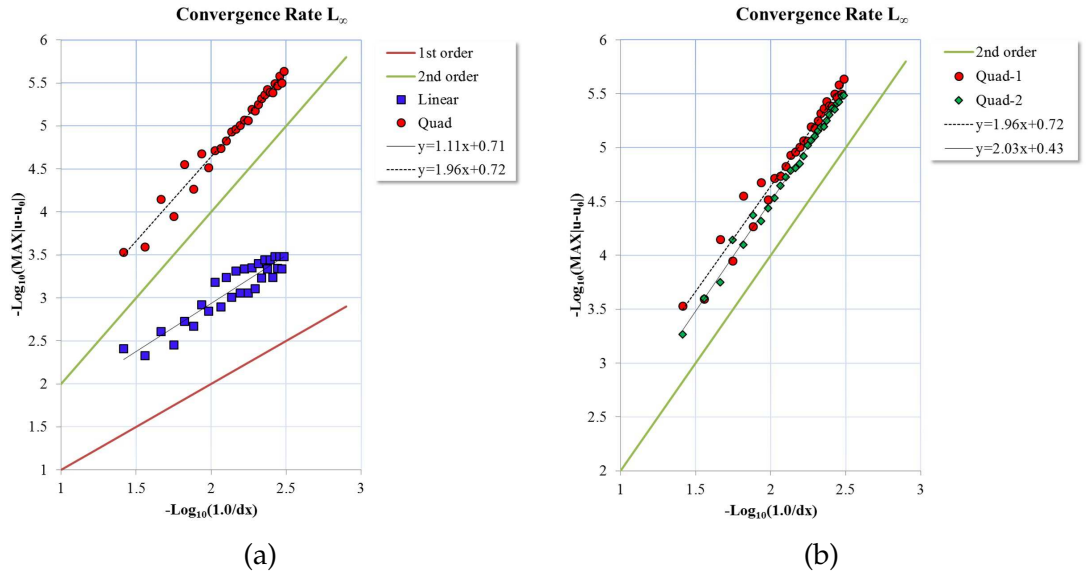


Figure 12: Numerical accuracy tests for Example 2. (a) The local convergence rates L^∞, (b) differences of the two quadratic methods.

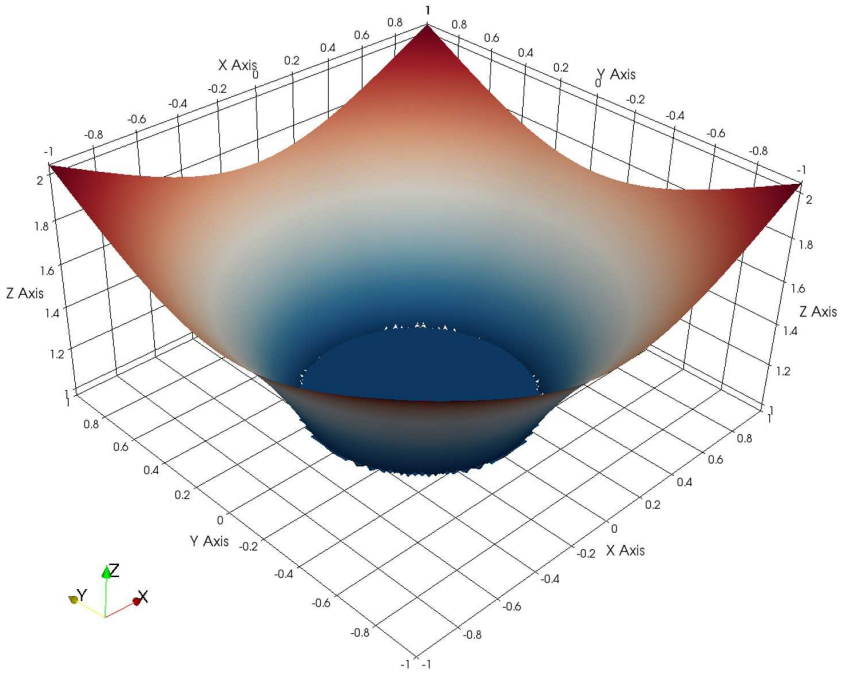
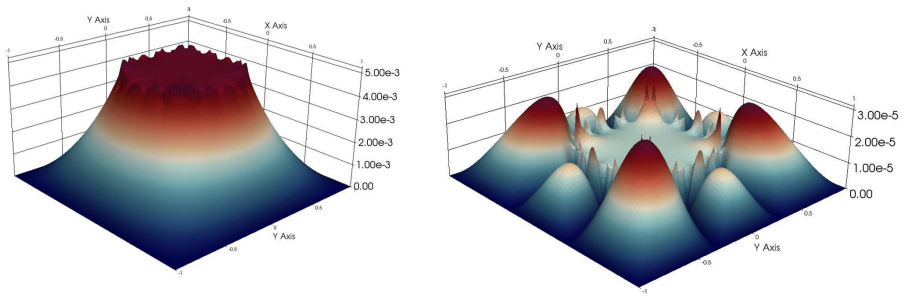


Figure 13: Solution of Laplace equation (Example 2).

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press



(a) (b)

Figure 14: Localization of the error for Example 2. (a) Linear method. (b) Quadratic method.

Table 2: Average error L^2 and maximum error L^∞.

Δx	Present GFM method				Previous GFM method [9]			
	L^∞ error	order	L^2 error	order	L^∞ error	order	L^2 error	order
1/10	9.7912E-3	-	7.2351E-3	-	3.2600E-2	-	2.9900E-2	-
1/20	2.8513E-3	1.853	1.7899E-3	2.010	1.3000E-2	1.330	1.1100E-2	1.430
1/40	7.8394E-4	1.907	4.7494E-4	1.941	5.0000E-3	1.380	4.0000E-3	1.470
1/80	2.2724E-4	1.857	8.7737E-5	2.327	1.9000E-3	1.400	1.4000E-3	1.510

5.3 Example 3

In previous 2-D example, an irregular interface problem with constant exact solution for the inner domain is studied. Now we study the Laplace equation $\nabla\cdot(\nabla u)=0.0$ with $x\in[-1,1]$ and $y\in[-1,1]$. As an exact solution, a constant field $u=0$ outside and a variable physical field $u=e^x\cos(y)$ inside the circle domain are defined with appropriate Dirichlet boundary conditions. The circle interface is defined by $x^2+y^2=r^2$ with outer pointed normal vectors $\mathbf{N}=(2x,2y)$ and $r=0.5$. The jump conditions are $[u]=e^x\cos(y)$ and $[u_n]=2e^x(y\sin(y)-x\cos(y))$. Fig. 15 indicates that present method reaches second order convergence rate in both local and global error tests. Fig. 16 shows the numerical solution with 96 grids in each direction and Table 3 gives a comparison of the results derived from the two GFM based methods.

Table 3: Average error L^2 and maximum error L^∞.

Δx	Present GFM method				Previous GFM method [9]			
	L^∞ error	order	L^2 error	order	L^∞ error	order	L^2 error	order
1/10	2.8790E-3	-	5.8066E-4	-	1.5300E-2	-	5.4000E-3	-
1/20	5.5882E-4	1.932	1.4639E-4	1.992	8.1000E-3	0.920	2.2000E-3	1.300
1/40	1.3707E-4	2.019	4.0393E-5	1.904	4.4000E-3	0.880	9.0000E-4	1.290
1/80	4.0929E-5	1.830	1.0532E-5	1.958	2.3000E-3	0.940	3.0000E-4	1.590

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

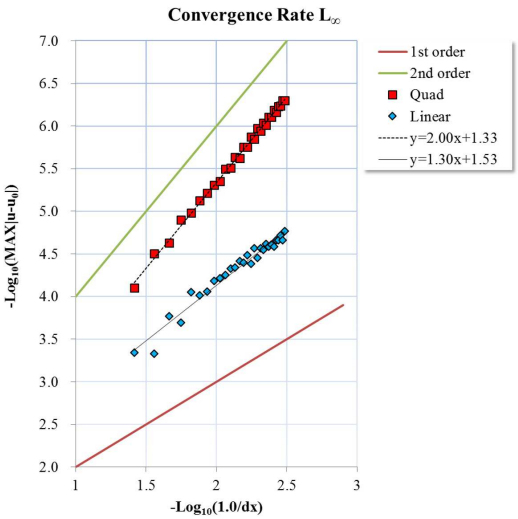


Figure 15: Numerical accuracy tests for Example 3.

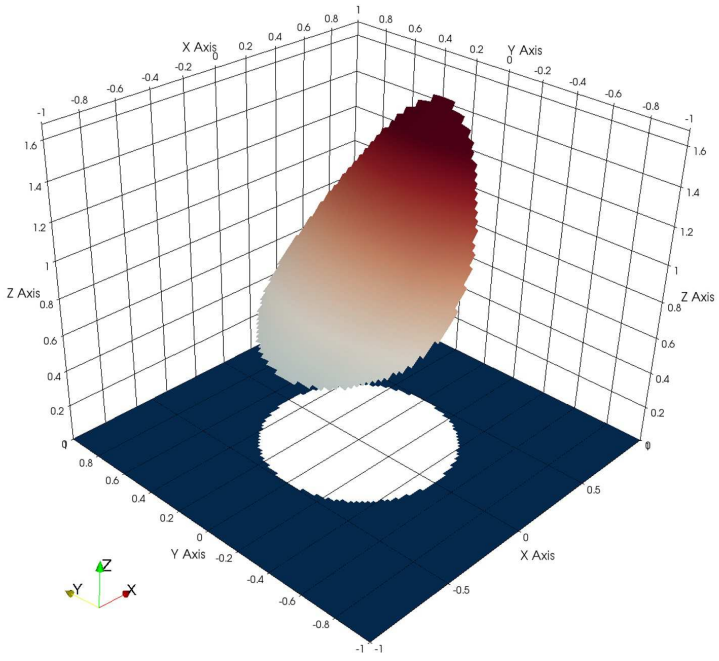


Figure 16: Solution of Laplace equation for Example 3.

5.4 Example 4

Previous 2-D examples (Example 2, Example 3) only consider the Laplace equation with accurate solution u varying in one subdomain and constant u in the rest area. The

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

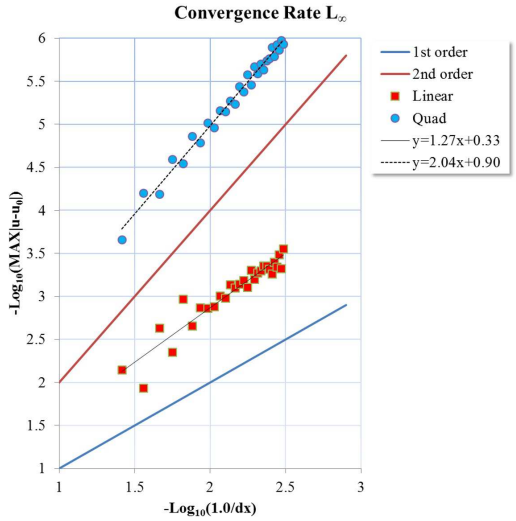


Figure 17: Numerical accuracy tests for Example 4.

present case consider an elliptic problem $\nabla\cdot(\beta\nabla u)=f(x,y)$ in two dimensions with $x\in[-1,1]$ and $y\in[-1,1]$. The interface is defined by $x^2+y^2=0$ with outer pointed normal vectors $\mathbf{N}=(N_x,N_y)=(2x,2y)$. The exact solution is $u=x^2+y^2$ in the interior region and $u=0.1(x^2+y^2)^2-0.01\ln(2(x^2+y^2)^{0.5})$ in the exterior region with appropriate Dirichlet boundary conditions. $\beta=1$ with $f(x,y)=4$ is defined in the interior region and $\beta=10$ with $f(x,y)=16(x^2+y^2)$ is in the exterior region. The jump conditions are $[u]=0.1(x^2+y^2)^2-0.01\ln(2(x^2+y^2)^{0.5})-(x^2+y^2)$ and $[\beta u_n]=(4(x^2+y^2)-0.1(x^2+y^2)^{-1}-2)(xN_x+yN_y)$. Fig. 17 shows the convergence rates of the spatial accuracy tests. The second order convergence rate of the spatial accuracy is obtained as expected. Fig. 18 shows the numerical solution with 96 grids in each direction.

5.5 Example 5

The previous numerical examples are tested based on a regular circle interface. From now we consider two irregular interface cases. The irregular interface (I) is described by a collection of points $(x(\theta),y(\theta))$ as follows,

$$\begin{cases} x(\theta)=0.02\sqrt{5}+(0.5+0.2\sin(5\theta))\cos(\theta), \\ y(\theta)=0.02\sqrt{5}+(0.5+0.2\sin(5\theta))\sin(\theta), \end{cases} \quad \theta\in[0,2\pi]. \tag{5.1}$$

The second irregular interface (II) is defined as

$$\begin{cases} x(\theta)=0.6\cos(\theta)-0.3\cos(3\theta), \\ y(\theta)=1.5+0.7\sin(\theta)-0.07\sin(3\theta)+0.2\sin(7\theta), \end{cases} \quad \theta\in[0,2\pi]. \tag{5.2}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

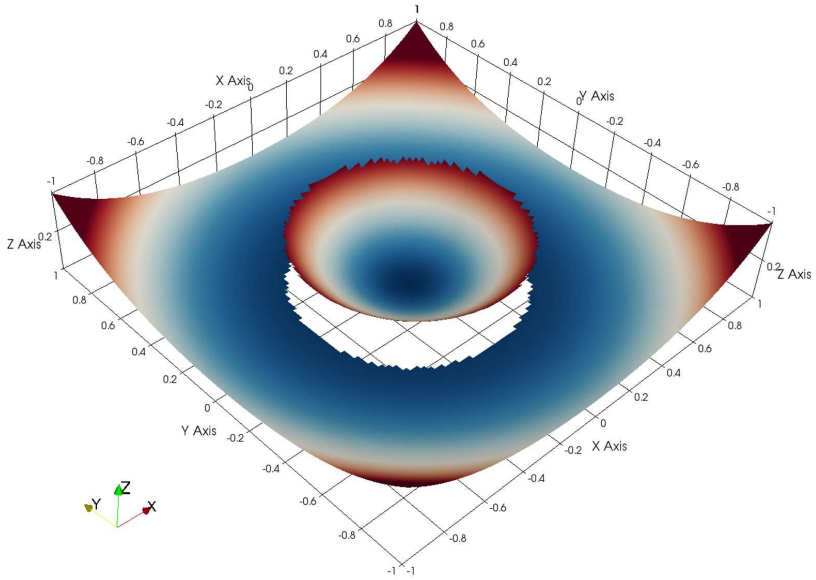


Figure 18: Solution of elliptic problem for Example 4.

The unit normal vectors $\mathbf{N}=(N_x,N_y)$ pointing from interior region to exterior region can be calculated using the method described in Section 3.1. In present interface problem, the same elliptic problem with equivalent jump conditions in $[u]$ and $[\beta u_n]$ as demonstrated in Section 5.4 are solved. The only difference exists in the shape of the interface. Fig. 19 shows the geometry of irregular interface (I) and the numerical solution with 96 grid points in each direction. Fig. 20 shows the geometry of irregular interface (II) and numerical solution with 96 grid points in two dimensions and Fig. 21 shows the results of numerical accuracy tests with regular circular interface, irregular interface (I) and (II). With the adoption of the modified quadratic method described in Section 3.2, the second order convergence rate is achieved.

5.6 Example 6

In this example an elliptic problem $\nabla\cdot(\beta\nabla u)=f(x,y)$ in two dimensions with $x\in[-1,1]$ and $y\in[-1,1]$ is considered. The discontinuous of the diffusive coefficient β is involved in this case. The exact solution is

$$u=\frac{x(\rho+1)-x(\rho-1){r_0}^2/r^2}{\rho+1+{r_0}^2(\rho-1)} \tag{5.3}$$

in the interior region and

$$u=\frac{2x}{\rho+1+{r_0}^2(\rho-1)} \tag{5.4}$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

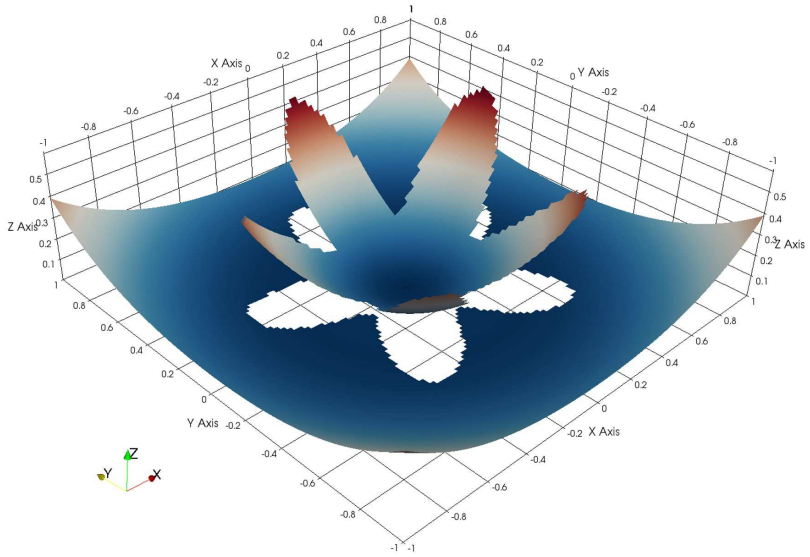


Figure 19: Solution of elliptic problem in Example 5 with irregular interface (I).

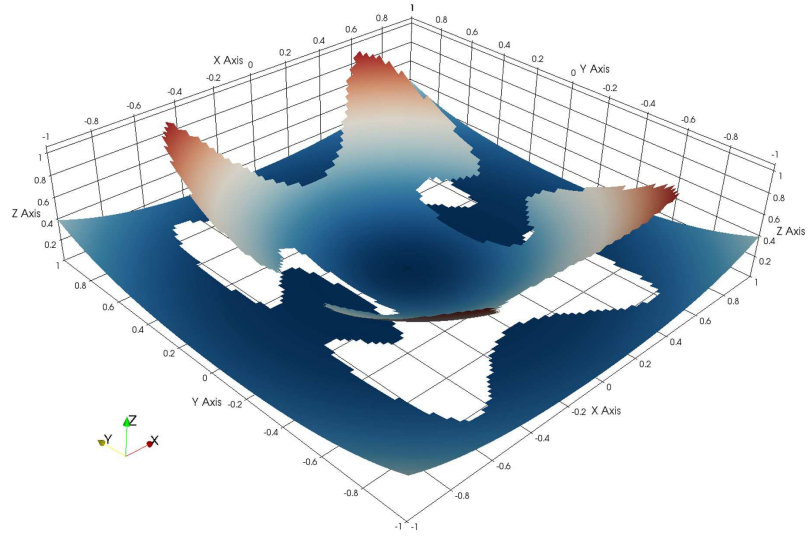


Figure 20: Solution of elliptic problem in Example 5 with irregular interface (II).

in the exterior region with appropriate Dirichlet boundary conditions. Here $r_0=0.5$ and $r=((x^2+y^2)^{0.5})$. $\rho=\beta_I/\beta_{II}$ is defined to describe the discontinuous coefficient β. $f(x,y)=0.0$ with $\beta=\beta_I$ is defined in the interior region and $\beta=\beta_{II}$ with

$$f(x,y)=\frac{\beta_{II}r_0{}^2(\rho-1)}{\rho+1+r_0{}^2(\rho-1)}\left[\frac{4x}{r^4}-\frac{8xy^2+4x(x^2-y^2)}{r^6}\right]$$

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

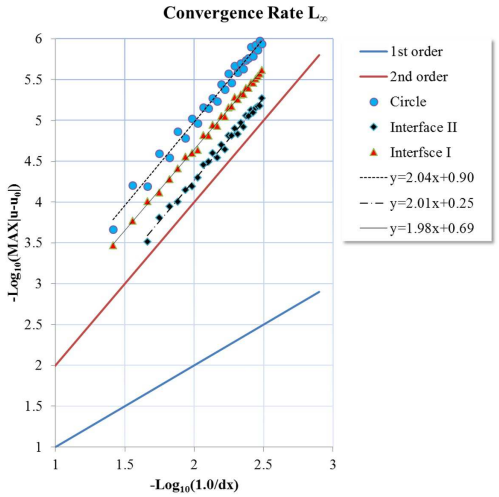


Figure 21: Numerical accuracy tests for Example 5.

is in the exterior region. The jump conditions are $[u]=0.0$ and

$$[\beta u_n]=\frac{1}{\rho+1+r_0{}^2(\rho-1)}\Big\{\Big[\Big(\rho+1-r_0{}^2(\rho-1)\frac{y^2-x^2}{r^4}\Big)\beta_I-2\beta_{II}\Big]N_x + \Big[r_0{}^2(\rho-1)\frac{2xy}{r^4}\beta_I\Big]N_y\Big\}.$$

The visualization of the solutions for $\rho=100000$ and $\rho=1/100000$ with 96×96 grids are given in Figs. 22(a) and (b), respectively. Fig. 23 provides the visualization of the local error. From the results of spatial accuracy study shown in Table 4, the convergence rates of the maximum error shows a second order accuracy. The evolution of the 1-norm condition number of the matrix for different grid resolution is given in Table 5. It can be observed that the condition number depends on the mesh size and the discontinuous diffusive coefficient β. The condition number is getting larger with the increase of the mesh size. Due to the use of the approach described in Sections 3, 4, the condition number

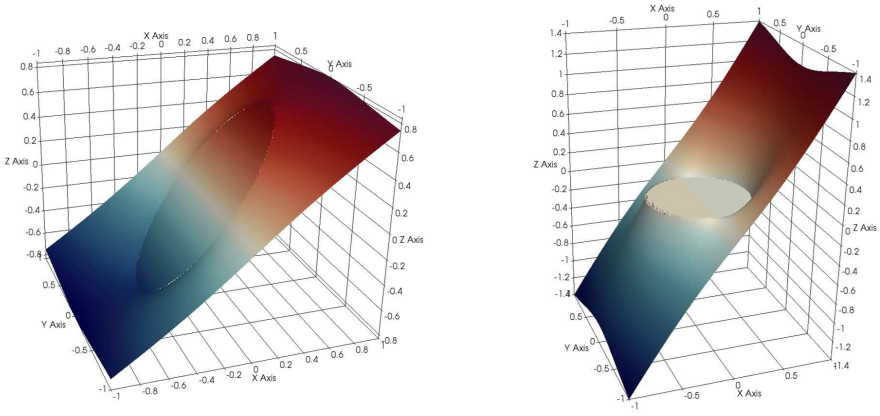
Table 4: Maximum error L^∞ for $\rho=100000$ and $\rho=1/100000$.

Δx	$\beta_I=1,\beta_{II}=100000$		$\beta_I=100000,\beta_{II}=1$	
	L^∞ error	order	L^2 error	order
H/10	6.2980E-3	-	2.8582E-2	-
H/20	1.2908E-3	2.20	1.6598E-2	1.31
H/40	2.3490E-4	2.34	2.7816E-3	2.44
H/80	5.2936E-5	2.11	6.1093E-4	2.13
H/160	1.2425E-5	2.06	1.2978E-4	2.17
H/320	3.2420E-6	1.95	2.9589E-5	2.10

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

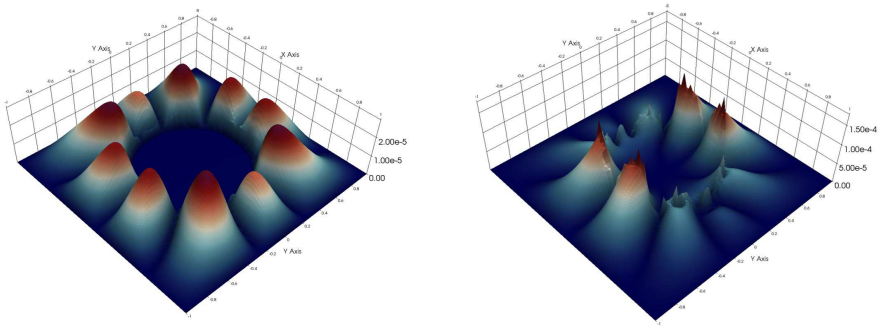
Table 5: Condition number for $\rho=1$, $\rho=100000$ and $\rho=1/100000$.

Δx	$\beta_I=1, \beta_{II}=1$	$\beta_I=1, \beta_{II}=100000$	$\beta_I=100000, \beta_{II}=1$
	Cond. Number	Cond. Number	Cond. Number
H/10	2.9790E+2	2.9705E+2	1.9364E+2
H/20	2.3738E+3	6.7813E+2	2.1895E+3
H/40	5.3602E+3	4.9073E+3	4.7168E+3
H/80	2.5130E+4	2.8873E+4	2.2644E+4
H/160	9.3284E+4	5.4919E+5	9.4962E+4
H/320	5.7815E+5	1.8102E+6	6.0966E+5



(a) (b)

Figure 22: Solution of u for Example 6.



(a) (b)

Figure 23: Localization of the error for Example 6. (a). $\beta_I=1$, $\beta_{II}=100000$. (b). $\beta_I=100000$, $\beta_{II}=1$.

is not getting large significantly even the diffusive coefficient β becomes very large. It can be concluded that the present method is robust in dealing with irregular interface problems with large jump in β.

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

5.7 Example 7

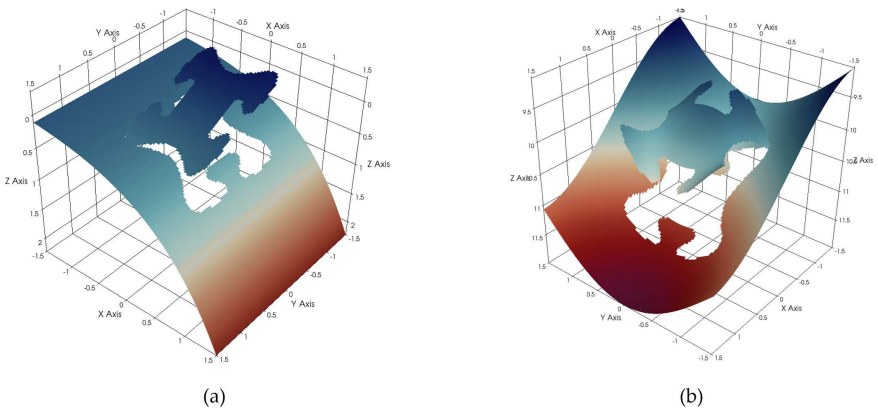
This example considers an elliptic problem $\nabla\cdot(\beta\nabla u)=f(x,y)$ in two dimensions with $x\in[-1,1]$ and $y\in[-1,1]$. The discontinuous of the diffusive coefficient β is considered in this case. The exact solution is $u=\cos(x)\sin(y)$ in the interior region and $u=e^x$ in the exterior region with appropriate Dirichlet boundary conditions. We define the diffusive coefficient β as $\beta_I(x,y)=\alpha_I(10+\sin(x)+\cos(y))$ in the exterior region and $\beta_{II}(x,y)=\alpha_{II}(10+\sin(x)\cos(y))$ in the interior region. The jump conditions are

$$[u]=\cos(x)\sin(y)-e^x \tag{5.5}$$

and

$$[\beta u_n]=[e^x\beta_I+(\sin(x)\sin(y))\beta_{II}]N_x-[(\cos(x)\cos(y))\beta_{II}]N_y. \tag{5.6}$$

The geometry of the interface together with the solution and the diffusive coefficient β are shown in Fig. 24, in which $\alpha_I=\alpha_{II}=1.0$ is considered. The numerical results shown in Table 6 indicate a second order convergence rate for the solution. The condition numbers with different mesh resolution are provided in Table 7.



(a) (b)

Figure 24: (a). Solution of u. (b). Diffusive coefficient $\beta(x,y)$.

Table 6: Maximum error L^∞ for different α_I and α_{II}.

Δx	$\alpha_I=1,\alpha_{II}=100$		$\alpha_I=100,\alpha_{II}=1$	
	L^∞ error	order	L^2 error	order
H/20	6.9017E-4	-	1.2988E-3	-
H/40	2.5571E-4	1.64	1.2344E-4	3.24
H/80	7.2884E-5	1.87	2.1315E-5	2.41
H/160	1.6634E-5	2.09	4.3900E-6	2.20
H/320	4.3042E-6	1.97	9.1334E-7	2.19

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

Table 7: Condition number for different α_I and α_{II}.

Δx	$\alpha_I=1, \alpha_{II}=1$	$\alpha_I=1, \alpha_{II}=100$	$\alpha_I=100, \alpha_{II}=1$
	Cond. Number	Cond. Number	Cond. Number
H/20	2.1676E+3	1.4219E+3	3.2151E+3
H/40	4.8882E+3	1.1090E+4	5.6048E+3
H/80	2.3023E+4	4.4388E+4	3.3709E+4
H/160	9.1351E+4	1.8516E+5	1.1907E+5
H/320	5.8041E+5	7.3347E+5	7.1717E+5

5.8 Example 8

The last example considers an elliptic problem $\nabla\cdot(\beta\nabla u)=f(x,y,z)$ in three dimensions with $x,y,z\in[-1.0,1.0]$. The shape of the interface is described by $\phi=r_0-(x^2+y^2+z^2)^{0.5}$, with $r_0=0.5$.The exact solution is $u=10(-xy^2-z^2-zx^2)$ in the interior region and $u=x^2+y^2+z^2$ is in the exterior region with appropriate Dirichlet boundary conditions. The diffusive coefficient β is $\beta_I(x,y,z)=2\sin(x)+z\cos(y)$ in the exterior region and $\beta_{II}(x,y,z)=\sin(z)+\cos(xy)$ in the interior region.

The jump conditions are

$$[u]=10(-xy^2-z^2-zx^2)-(x^2+y^2+z^2) \tag{5.7}$$

and

$$\begin{aligned}[\beta u_n]=&[\beta_I(-10y^2-20xz)-2x\beta_{II}]N_x+[\beta_I(-20xy)-2y\beta_{II}]N_y\\&+[\beta_I(-10x^2-20z)-2z\beta_{II}]N_z.\end{aligned} \tag{5.8}$$

Fig. 25 is the numerical result of u for the inner region and the outer region. The solution and the diffusive coefficient u of a slice in z-direction are shown in Fig. 26. The numerical results in Table 8 indicate a second order convergence rate for the solution.

Table 8: Maximum error L^∞ for Example 7.

Δx	L^∞ error	order	L^2 error	order
H/40	1.7706E-4	-	4.3392E-5	-
H/80	4.7929E-5	1.92	1.5330E-5	1.68
H/160	1.0923E-5	2.09	2.9763E-6	2.26

6 Discussion on iterative solver

One of the major challenges of this research is to obtain an efficient solver for the sparse linear system. Since the additional constraint equations are assembled to form the linear system, the additional equations may break the symmetry and positive definiteness

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

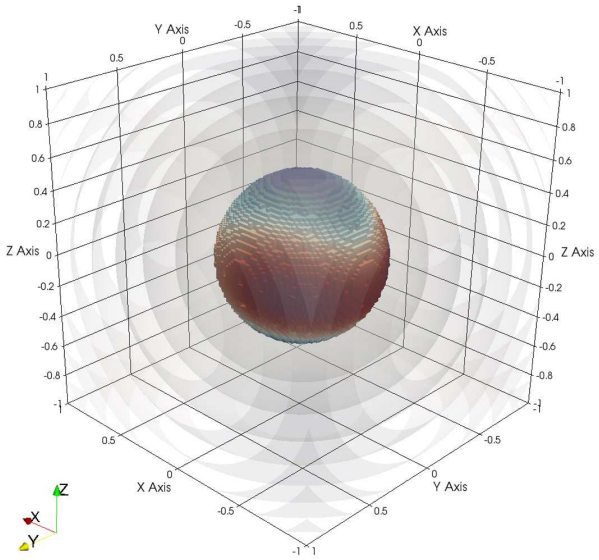
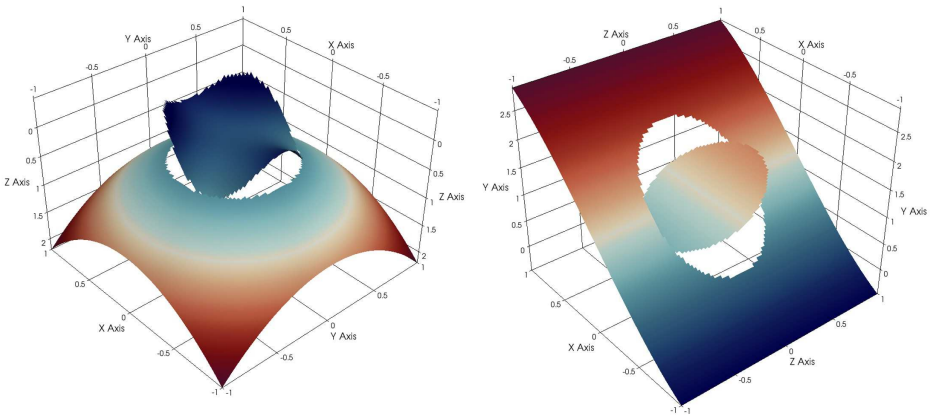


Figure 25: Solution of u for the inner region and outer region.



(a) (b)

Figure 26: z-slice of u (a) and β (b) at $n=100$.

of the original matrix and enlarge the condition number of the linear system, making it difficult to solve. The most appropriate solver for such linear system is Krylov subspace iteration methods [20] such as BICGSTAB and GMRES etc., which are designed for general sparse matrices without any assumptions about the structure of the matrix. Besides, to make the linear system more suitable for numerical solution, the preconditioning method is considered to reduce the condition number of the problem. We have tried several preconditioners including ILU (Incomplete LU Factorization), SOR (Succes-

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

Table 9: Average error L^2 and maximum error L^∞.

Solver	Preconditioner	$\beta_I=1, \beta_{II}=100000$	
		Iteration (max steps: 400)	Stopped Error
BICGSTAB	SOR	400	>1.0E-8
BICGSTAB	ILU	103	1.0E-14
BICGSTAB	GAMG	28	1.0E-14
GMRES	SOR	400	>1.0E-8
GMRES	ILU	126	1.0E-14
GMRES	GAMG	32	1.0E-14

sive Over Relaxation), and GAMG (Geometric Algebraic Multi-Grid) in this study. The numerical tests (Table 9) are based on the 2-D example in Section 5.5 with the grid number of 306×306. From Table 9, the BICGSTAB with GAMG preconditioner shows the fastest convergence rate.

7 Conclusions

In this study, an efficient Ghost Fluid Method for solving an embedded interface problem is proposed. In the method the physical jumps across interface, the derivative jumps and the discontinuous diffusive coefficient are handled by introducing additional virtual points. Only simple interpolation scheme are used to obtain the additional equations. The proposed method is robust and efficient for solving elliptic equations with geometrically complex domains even with very coarse grid resolution. Accuracy of the present method is verified by 1-D, 2-D and 3-D examples with irregular interfaces.

Highlights of present jump conditions treatment contain four phases. First, the discontinuity and the jump conditions are enforced through additional equations constructed by simple interpolation scheme which is easy for implementation. It can be directly calculated by any "black-box" iterative sparse matrix solver. Second, unlike existing virtual points method [10,11], the property of the derived linear system is analyzed in details. To avoid an ill-conditioned matrix, adaptive stencils are proposed for building the discontinuous approximation of the physical field. Third, the method is second order accuracy and works well even with very coarse grid. Compared with traditional first order GFM treatment [9], our method reaches second order convergence rates for spatial accuracy tests and the absolute error decreased remarkably. Finally, the proposed method is robust and efficient for solving elliptic problems with a large jump in the diffusion coefficients.

References

[1] S. R. Ashton and F. C. Alexei, A matlab-based finite difference solver for the Poisson problem with mixed Dirichlet-Neumann boundary conditions, Comput. Phys. Commun., 184 (2012),

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press

783–798.
[2] Z. Chen and J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math., 79 (1998), 175–202.
[3] C. S. Peskin, Flow patterns around heart valves: A numerical method, J. Comput. Phys., 10 (1972), 252–271.
[4] J. Mohd-Yusof, Combined immersed-boundary/*B*-spline methods for simulations of flow in complex geometries, Center for Turbulence Research Annual Research Briefs, 161 (1997), 317–327.
[5] 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.
[6] Z. Li and K. Ito, Maximum principle preserving schemes for interface problems with discontinuous coefficients, SIAM J. Sci. Comput., 23 (2001), 339–361.
[7] X. Zhong, A new high-order immersed interface method for solving elliptic equations with imbedded interface of discontinuity, J. Comput. Phys., 225 (2007), 1066–1099.
[8] 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 (1999), 457.
[9] X. D. Liu, R. P. Fedkiw and M. Kang, A boundary condition capturing method for Poissons equation on irregular domains, J. Comput. Phys., 160 (2000), 151–178.
[10] Y. Zhou, S. Zhao, M. Feig and G. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys., 213 (2006), 1–30.
[11] Y. Zhou and G. Wei, On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method, J. Comput. Phys., 219 (2006), 228–246.
[12] S. Yu and G. Wei, Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities, J. Comput. Phys., 227 (2007), 602–632.
[13] H. Johansen and P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, J. Comput. Phys., 147 (1998), 60–85.
[14] M. Cisternino and L. Weynans, A parallel second order Cartesian method for elliptic interface problems, Tech. Rep., 7573 INRIA; 2011.
[15] F. Gibou, R. P. Fedkiw, L. T. Cheng and M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys., 176 (2002), 205–227.
[16] J. Bedrossian, J. H. Von Brecht and S. Zhu et al., A second order virtual node method for elliptic problems with interfaces and irregular domains, J. Comput. Phys., 229 (2010), 6405–6426.
[17] J. L. Hellrung, L. Wang and E. Sifakis et al., A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions, J. Comput. Phys., 231 (2012), 2015–2048.
[18] J. Choi, R. Oberoi, J. Edwards and J. Rosati, An immersed boundary method for complex incompressible flows, J. Comput. Phys., 224 (2007), 757–784.
[19] C. Liu and C. H. Hu, An efficient immersed boundary treatment for complex moving object, J. Comput. Phys., 274 (2014), 654–680.
[20] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, 1996.

https://doi.org/10.4208/cicp.OA-2016-0155 Published online by Cambridge University Press