Journal of Computational and Applied Mathematics 272 (2014) 195–220

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

MIB Galerkin method for elliptic interface problems

Kelin Xia [a,b], Meng Zhan [b], Guo-Wei Wei [a,c,*]

[a] Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA

[b] Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Wuhan 430071, China

[c] Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA

ARTICLE INFO

Article history:
Received 12 January 2014
Received in revised form 1 May 2014

Keywords:
Finite element method
Galerkin formulation
Matched interface and boundary
Elliptic equations
Discontinuous coefficients

ABSTRACT

Material interfaces are omnipresent in the real-world structures and devices. Mathematical modeling of material interfaces often leads to elliptic partial differential equations (PDEs) with discontinuous coefficients and singular sources, which are commonly called elliptic interface problems. The development of high-order numerical schemes for elliptic interface problems has become a well defined field in applied and computational mathematics and attracted much attention in the past decades. Despite of significant advances, challenges remain in the construction of high-order schemes for nonsmooth interfaces, i.e., interfaces with geometric singularities, such as tips, cusps and sharp edges. The challenge of geometric singularities is amplified when they are associated with low solution regularities, e.g., tip-geometry effects in many fields. The present work introduces a matched interface and boundary (MIB) Galerkin method for solving two-dimensional (2D) elliptic PDEs with complex interfaces, geometric singularities and low solution regularities. The Cartesian grid based triangular elements are employed to avoid the time consuming mesh generation procedure. Consequently, the interface cuts through elements. To ensure the continuity of classic basis functions across the interface, two sets of overlapping elements, called MIB elements, are defined near the interface. As a result, differentiation can be computed near the interface as if there is no interface. Interpolation functions are constructed on MIB element spaces to smoothly extend function values across the interface. A set of lowest order interface jump conditions is enforced on the interface, which in turn, determines the interpolation functions. The performance of the proposed MIB Galerkin finite element method is validated by numerical experiments with a wide range of interface geometries, geometric singularities, low regularity solutions and grid resolutions. Extensive numerical studies confirm the designed second order convergence of the MIB Galerkin method in the L_∞ and L_2 errors. Some of the best results are obtained in the present work when the interface is C^1 or Lipschitz continuous and the solution is C^2 continuous.

1. Introduction

In the last few decades, a wide variety of mathematical techniques, including finite difference, finite element, finite volume, wavelet, radial basis function, meshless and spectral methods, have been developed to solve partial differential equations (PDEs). However, for elliptic PDEs with discontinuous coefficients and singular sources, i.e., elliptic interface problems, ordinary numerical methods do not work, and special interface schemes are required to deal with interface jump

* Corresponding author at: Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA. Tel.: +1 517 353 4689; fax: +1 517 432 1562.

E-mail address: wei@math.msu.edu (G.-W. Wei).

http://dx.doi.org/10.1016/j.cam.2014.05.014
0377-0427/

conditions. Elliptic interface problems have a variety of applications in many scientific and engineering disciplines, including fluid dynamics [1–8], electromagnetic wave propagation [9–14], materials science, [15,16] and biological systems [17–21]. The past few decades have witnessed intensive research activity in interface problems [22–27,16,28–32,12,6,7,33–43]. Owing to its mathematical complexity and essential importance in a number of application areas, the study of interface problems has evolved into a well defined field in applied and computational mathematics.

Peskin pioneered the immersed boundary method (IBM) in 1977 [44–46]. Mayo proposed integral equation methods for this class of problems. [47,48]. A remarkable contribution in the field was due to LeVeque and Li, who constructed the second order interface scheme, the immersed interface method (IIM) [49]. Recently, a few other interesting interface methods have been reported, including a ghost fluid method (GFM) [50,51], the upwinding embedded boundary method [42], a piecewise-polynomial discretization [52], and a virtual node method [53].

In the past decade, we have developed a highly accurate algorithm, the matched interface and boundary (MIB) method [54,55,13,56,57] for solving elliptic interface problems. Some essential ideas behind the MIB method were introduced in our earlier work in solving Maxwell's equations [13]. One of the main features of the MIB method is that it is of arbitrarily high-order accuracy in principle, based on high order Lagrange polynomials. The 16th order accurate MIB scheme has been demonstrated with a simple interface geometry [13,56], and sixth-order accurate MIB schemes have been developed for complex interfaces in two-dimensional (2D) and three-dimensional (3D) domains [55,56]. A major driven force behind the development of the MIB methodology is practical needs in scientific and engineering applications. For example, a few generations of MIB schemes have been constructed to address challenges in solving Poisson–Boltzmann equation (PBE) describing the electrostatic potential in biomolecular systems [18–21]. It takes much effort to achieve the second order convergence in solving the PBE for arbitrarily complex protein interfaces with geometric singularities and multiple material interfaces [58]. Recently, Zhao has developed impressive MIB schemes for the Helmholtz equation [14,59]. Note that interface conditions in this problem are more complex. High order schemes are particularly valuable for electromagnetic wave scattering and propagation due to their nature of involving high frequencies. Zhao's MIB schemes achieve the fourth order accuracy for the Helmholtz equation with arbitrarily curved interfaces [14]. More recently, Zhou et al. have developed a second order accurate MIB method to solve the Navier–Stokes equations with discontinuous viscosity and density on non-staggered Cartesian grids [60]. Their approach has been applied to problems in geodynamics. A comparison of the GFM, IIM and MIB approaches is discussed in our earlier work [56,57]. Recently, Beale and Layton have provided a proof of the second order convergence of the IIM for smooth interfaces [43]. However, Rigorous convergence analysis of most other interface schemes, including the MIB method, is yet to be developed.

Finite element methods (FEMs) are also a class of efficient approaches to the above elliptic interface methods. Babuška constructed one of the first FEMs for elliptic interface problems in 1970 [61]. Since then, elliptic interface problems have attracted extensive effort in the FEM community as well [62–65]. Recently, non-conforming FEM [63] and discontinuous Galerkin (DG) FEM [66,64] have been developed for elliptic equations with discontinuous coefficients. According to the topological relation between discrete elements and the interface, one can classify FEM based interface methods into two types: interface-fitted FEMs and immersed FEMs. In the interface-fitted FEMs, or body-fitted FEMs, element meshes are aligned with the interface [67,68,63,66,64]. Two major factors that determine the performance of interface-fitted FEMs are respectively the Galerkin formulation and the quality of element meshes near the interface. Standard FEM techniques, such as priori and/or posteriori error estimation based local mesh refinements can be employed to improve accuracy of interface-fitted FEMs. However, in this type of methods, mesh generation and refinement can be a technically demanding and computationally time consuming process. To avoid the complicated mesh generation process, immersed FEMs have been proposed to allow the interface to cut through elements so that simple structured Cartesian meshes can be employed [62,69–72]. Appropriate interface algorithms similar to those used in the finite-difference based elliptic interface methods are designed in immersed FEMs to deal with complex interface geometries. As a result, the performance of immersed FEMs typically depends on the effectiveness of interface schemes for complex interface geometries. For example, extended finite element method [73] makes use of enrichment functions to implement interface jump conditions. Therefore, one may consider immersed FEMs as the Galerkin formulations of finite difference based interface schemes. Based on the Galerkin formulation, FEM approximations of elliptic interface problems can be rigorously analyzed [65,66,74,75].

Recently, Wang and Ye have introduced a novel weak Galerkin formulation for solving PDEs [76]. An essential ingredient of the Wang–Ye Galerkin method is the use of a novel concept, *the discrete gradient*, in the finite element procedure, which endows the Wang–Ye Galerkin method with high flexibility to deal with geometric complexities, boundary conditions and interface jump conditions. The discrete gradient is numerically implemented via the use of discontinuous functions, a procedure that is akin to the DG FEMs. However, unlike DG methods, the Wang–Ye Galerkin method enforces only weak continuity of variables naturally through well defined discrete gradient operators. Therefore, the Wang–Ye Galerkin method avoids the determination of free parameters due to the excessive flexibility given to individual elements. Consequently, Wang–Ye Galerkin method is found to be absolutely stable once properly constructed for solving PDEs [76], including elliptic interface problems [77].

There have been numerous advances in elliptic interface problems in the past decades. At present, it is fairly easy to construct second order accurate elliptic interface schemes for smooth interfaces in 2D and 3D settings. Nonetheless, there are still many unsolved problems in the field. One of these challenges is the development of higher order interface schemes motivated by problems associated with high frequency waves, such as the vibration analysis of engineering structures, the propagation and scattering of electromagnetic and acoustic waves, shock–vortex interactions in compressible fluid flows,

turbulence, combustion, etc. Although it is not difficult to construct high order methods and even spectral methods for the above problems with only straight interfaces in simple domains, it is truly difficult to design high order convergence interface schemes for arbitrarily complex interface geometries. Zhao proposed a fourth order MIB scheme for wave propagation described by the Helmholtz equation with arbitrarily curved interfaces in 2D [14]. We have developed up to sixth order MIB schemes for the Poisson equation with ellipsoidal interfaces in 3D [55]. A challenging problem, namely, the construction of sixth-order 3D interface schemes for arbitrarily curved smooth interfaces, has been posed as a standing open problem in the field since 2007 [55].

Another problem in the elliptic interface research is the development of second order or higher order schemes for elliptic problems with nonsmooth interfaces or interfaces with Lipschitz continuity [52,28,71,54,55,19,58]. In many physical problems, such as electromagnetic devices, molecular surfaces in biological systems, and nano-technology, one frequently encounters nonsmooth material interfaces, which are referred as geometric singularities [58,78,54,55,19]. Geometric singularities lead to much difficulty in the collocation based design of interface algorithms. They also cause problems in the construction of Galerkin based immersed FEMs. The first second order accurate scheme for 2D nonsmooth interfaces was constructed in 2007 [54]. Recently, other interesting second order methods have also been reported for this class of problems in 2D [52,71,58]. At present, the best results in 2D reported in the literature are all limited to the second order convergence for arbitrarily nonsmooth interfaces [52,71,54,58]. Therefore, another open problem in the field is to construct 2D interface schemes of numerical orders higher than two for arbitrarily nonsmooth interfaces.

Since most realistic problems are in 3D settings, the development of high order schemes for nonsmooth interfaces in 3D domains is one of most important tasks in elliptic interface problems. Motivated by the need in dealing with molecular surface of macromolecules, second order MIB schemes have been developed for 3D elliptic PDEs with arbitrarily non-smooth interfaces or geometric singularities [55]. Such schemes have found successful applications in the electrostatic analysis of biomolecular systems [19–21,79,80] and light propagation in biomedical imaging [81]. In fact, a fourth order MIB scheme for 3D elliptic PDEs with nonsmooth interfaces has also been constructed [55]. However, the success of such a scheme is restricted to only a few special interface geometries. One encounters much difficulty in developing fourth order schemes for arbitrarily non-smooth interfaces in 3D domains, which is, in our opinion, one of the most important open problems in the field.

Technically, the above mentioned difficulties due to geometric singularities in the elliptic interface problems significantly deteriorate in many physical situations because geometric singularities can induce solution singularities. For example, it is well known that the electric field diverges near the geometric singularities, such as tips of electrodes, antenna and elliptic cones, and sharp edges of planar conductors. This phenomenon, known as tip-geometry effect, is often used in the design of electric devices, such as atomic force microscopy. Since electrostatic potential is modeled by the Poisson equation in the electrostatic analysis and the electric field is related to the gradient of the electrostatic potential, the solution to the Poisson equation has a lower regularity, i.e., the gradient may not exist at geometric singularities. In general, collocation formulations encounter difficulties for this class of solution singularities, although some special approaches may be used for isolated geometric singularities [19,58]. Whereas, Galerkin formulations are more suitable to deal with solution singularity problems. Hou et al. constructed an immersed FEM approach which is of first order convergence in the solution and 0.7th order convergence in the gradient of the solution when the solution of the Poisson equation is C^1 continuous and the interface is Lipschitz continuous [71]. For H^2 continuous solution and Lipschitz continuous interface, 1.5-order convergence in solution is obtained with 0.65th order for the derivative. Recently, a major progress in this problem was made by using the Wang–Ye Galerkin method, which yields at least 1.75th order convergence in the solution and 1st order convergence in the gradient [77]. It appears that the weak Galerkin nature of the Wang–Ye Galerkin method [76] has played an essential role in achieving such a success.

The objective of the present work is to formulate Galerkin based MIB methods, motivated by the success of the Wang–Ye Galerkin method for resolving the low solution regularity problem. The proposed MIB Galerkin method combines the flexibility of the MIB method for geometric singularities with the power of the Galerkin formulation for solution singularities. In the present MIB Galerkin approach, we utilize Cartesian grid based triangular elements to by-pass mesh generation and refinement. We make use of classical continuous finite element basis functions throughout the whole computational domain. Because complex interfaces cut through elements, we define two sets of overlapping elements, called MIB elements, in the vicinity of the interface to ensure the continuity of the basis functions across the interface. Consequently, the differential operators near the interface are evaluated as if there is no discontinuous coefficients. However, the overlapping domains lead to extra nodes and elements which are not admissible in the classical sense of FEM partitions. In the MIB method, function values on extra nodes, i.e., overlapping domains, are determined by the enforcement of interface jump conditions. The resulting matrix has wider bandwidths at interface locations, but this approach avoids resorting to the use of Lagrange multipliers because the interface jump conditions are directly incorporated into the discretization. On the other hand, the use of Lagrange multipliers will lead to a larger discretization matrix. Similar to the original MIB collocation method, the present MIB Galerkin method utilizes a set of lowest order interface jump conditions to enhance the stability of the interface scheme. Comparing to the original MIB method, which requires many special schemes to deal with geometric singularities, the present MIB Galerkin method is quite simple in its algorithms. Additionally, the present MIB Galerkin is able to directly handle solution singularities, i.e., low solution regularities.

The rest of this paper is organized as follows. Section 2 is devoted to the theory of the MIB Galerkin formulation. We start with a critical review of the original MIB method, which explains the MIB strategy in both collocation and Galerkin

formulations. Then, the MIB Galerkin setting is described, followed by the detailed formulation. We introduce two sets of overlapping MIB elements to describe the approximation of the true solution and the fictitious values in the vicinity of the interface. The computational algorithms of the present MIB Galerkin method are presented in Section 3. We use classical linear interpolation basis functions on all elements. As in the MIB collocation formulation, the major task of the present method is to determine fictitious values on irregular elements near the interface. The same as in our earlier MIB method, fictitious values are evaluated via the enforcement of the interface jump conditions. We discuss a normal algorithm in which the interface cuts through two edges of a triangular element. Additionally, we consider a situation in which the interface passes through a vertex of an element. We note that the present MIB Galerkin algorithm is significantly simpler than that of the original MIB method. In Section 4, we carry out extensive numerical experiments to validate the order of accuracy and demonstrate the performance of the proposed MIB Galerkin formulation. We first study a variety of geometric morphologies, including interfaces with geometric singularities. The designed second order convergence is confirmed in our numerical solution and its gradient. We also test the persistence of the convergence against high grid resolutions in a number of interface geometries. Additionally, we investigate situations with singular solutions, i.e., solutions with low regularity. Some of the best results are observed for these challenging tests. This paper ends with some concluding remarks.

2. MIB Galerkin method

In a 2D elliptic interface problem, we consider an open bounded domain $\Omega \subset \mathbb{R}^2$, which is divided into two subdomains, Ω^+ and Ω^-, by a given interface Γ. The boundary $\partial\Omega$ and interfaces Γ may be Lipschitz continuous. The interface can be characterized by a piecewise smooth level-set function $\phi \in \Omega$, such that $\Gamma = \{\mathbf{x}|\phi = 0, \forall \mathbf{x} \in \Omega\}$. As such, two subdomains can be given by $\Omega^+ = \{\mathbf{x}|\phi \geq 0, \forall \mathbf{x} \in \Omega\}$ and $\Omega^- = \{\mathbf{x}|\Phi < 0, \forall \mathbf{x} \in \Omega\}$. The 2D elliptic interface problem is given by

$$-\nabla \cdot \beta(\mathbf{x})\nabla u(\mathbf{x}) = g(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega \tag{1}$$

$$u = g_b, \quad \forall \mathbf{x} \text{ on } \partial\Omega \tag{2}$$

where $g(\mathbf{x})$ is a piecewise continuous function, g_b is the boundary value, and $\beta(\mathbf{x})$ is a variable coefficient that is discontinuous on the interface Γ. As a result, two jump conditions are required to make the problem well posed

$$[u] = u^+ - u^- = \Phi, \quad \forall \mathbf{x} \text{ on } \Gamma \tag{3}$$

$$[\beta u_n] = \beta^+ u_n^+ - \beta^- u_n^- = \Psi, \quad \forall \mathbf{x} \text{ on } \Gamma \tag{4}$$

where u^+, u_n^+ and β^+ denote their limiting value from Ω^+ side of the interface Γ, and u^-, u_n^- and β^- denote their limiting value from Ω^- side of the interface Γ. The derivatives u_n^+ and u_n^- are evaluated along the normal direction on the interface. The regularity of $\Phi(\mathbf{x})$ and $\Psi(\mathbf{x})$ is at least C^1 continuous. Eqs. (1)–(4) define the elliptic interface problem to be solved in the present work. Since Eqs. (3) and (4) are essential for the well-posedness of the original problem, we call them the set of essential jump conditions.

2.1. Critical review of MIB collocation methods

To solve elliptic interface problem defined above, we typically extend interface jump conditions in the MIB method. To increase the number of interface conditions, we usually differentiate Eq. (3) along the tangential direction of the interface to obtain an additional interface condition,

$$[u_\tau] = u_\tau^+ - u_\tau^-.$$

This can be done since interface is not aligned with the mesh lines in general. It does not create any additional accuracy reduction in the discretization because the resulting equation is of first order, just like jump condition (4). In the Cartesian representation, we define the normal vector and the tangential vector at an arbitrary interface point as $\vec{n} = (\cos\theta, \sin\theta)$ and $\vec{\tau} = (-\sin\theta, \cos\theta)$, respectively. Three jump conditions can be expressed as

$$[u] = u^+ - u^- = \Phi \tag{5}$$

$$[u_\tau] = (-u_x^+ \sin\theta + u_y^+ \cos\theta) - (-u_x^- \sin\theta + u_y^- \cos\theta) = \Phi_\tau \tag{6}$$

$$[\beta u_n] = \beta^+(u_x^+ \cos\theta + u_y^+ \sin\theta) - \beta^-(u_x^- \cos\theta + u_y^- \sin\theta) = \Psi. \tag{7}$$

This set of jump conditions involves at most first order derivatives. Therefore, we called it the set of lowest order jump conditions in our earlier work [18–20]. In our MIB method, the enforcement of this set of conditions determines the fictitious values to be used in the uniform discretization of differentiation operators in a given PDE.

The main features of the original MIB method, or the MIB collocation method, are critical reviewed. The similarities and differences between MIB collocation and MIB Galerkin are also discussed.

1. The original MIB method is based on the collocation formulation, which limits its ability to handle problems with low solution regularity. A main goal of the present work is to remove this limitation by a Galerkin formulation.

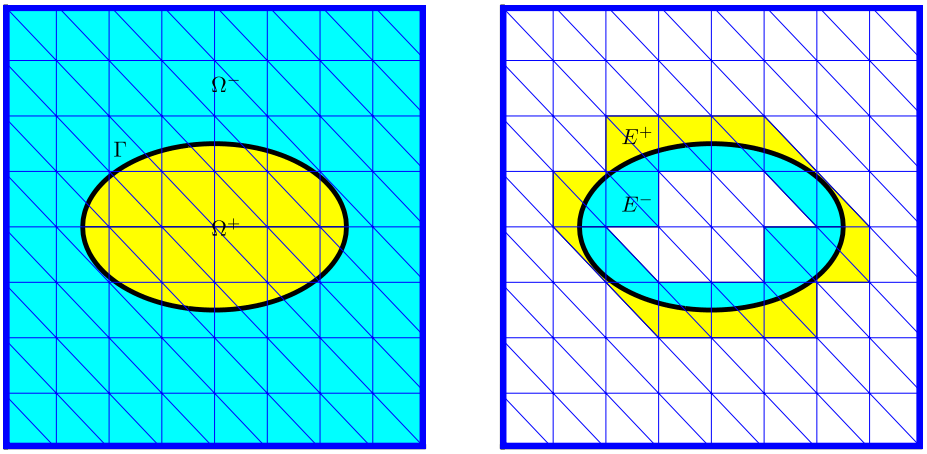

Fig. 1. The domain decomposition via the interface Γ (the ellipse) in a triangular Cartesian mesh. Left chart: Partition of the domain Ω into Ω^+ and Ω^- by interface Γ; Right chart: Illustration of irregular domains E^+ and E^-.

2. MIB collocation employs simple Cartesian meshes so as to by-passes mesh generation and local mesh refinement. Additionally, topological relation among grid nodes is simple so that high order schemes can be easily constructed. In contrast, the present MIB Galerkin approach utilizes the triangular elements, which are generated by introducing diagonal lines to the original Cartesian grid.
3. The MIB collocation method employs a dimensional splitting strategy inherited from the discrete singular convolution (DSC) algorithm [82,83] to reduce 2D and 3D interface problems into 1D ones. Therefore, the topological complexity in handling 3D interface geometries is significantly reduced. In contrast, MIB Galerkin is based on 2D or 3D finite elements.
4. A main feature of the MIB collocation method is that it enforces only lowest order interface jump conditions—a strategy introduced in our earlier work [13]. To achieve arbitrarily high order accuracy, we iteratively utilize the same set of lowest order interface jump conditions [13,19]. This strategy avoids the numerical instability and accuracy reduction in implementing higher order jump conditions. This approach is also employed in the MIB Galerkin formulations. In fact, in the Galerkin formulation, one can omit condition (6) because normally two sets interface jump conditions are utilized for each triangular element.
5. In our MIB collocation method, fictitious values are created on the irregular domains so that a uniform discretization scheme can be directly employed throughout the computational domain, including the vicinity of the interface and singular source [13,18–21]. This approach was intensively used in our earlier DSC algorithm [82,83] and is, to certain extent, similar to the fictitious domain method [84]. It is employed in the present MIB Galerkin formulation via the introduction of overlapping MIB elements. An advantage of this approach is that it avoids the confusion in determining the discretization scheme in the vicinity of the interface when the geometry is very complex.
6. In the MIB collocation method, enforcing the jump conditions is separated from the discretization of the governing PDE. Therefore, the interface modeling is virtually independent of the governing PDE and thus, the technique developed for solving one PDE can be applied to a vast range of other PDEs, which may differ by interface conditions. This feature is maintained in the present MIB Galerkin formulation.
7. In the original arbitrarily high order MIB schemes [13,19], the Lagrange polynomials of appropriate orders are utilized in discretizing PDE operators and in approximating fictitious values through enforcing interface jump conditions. In contrast, the present MIB Galerkin formulation utilizes general interpolation polynomials of appropriate orders to furnish the MIB element spaces and to represent fictitious values.

2.2. MIB Galerkin setting

To carry out a Galerkin formulation of the elliptical interface problem, one has to choose one of two types of FEM meshes, the interface-fitting meshes and the regular Cartesian meshes, as discussed in the Introduction. In fact, each type of meshes has its advantages and disadvantages. Both types of meshes have been extensive used in the literature. To make our MIB Galerkin formulation unique, we choose the Cartesian type of meshes in our work. The triangular Cartesian meshes are automatically generated by introducing diagonal lines to the rectangular Cartesian meshes. Since our triangular partition is based on the rectangular Cartesian grid, we can also directly employ the rectangular Cartesian grid when the corresponding basis functions are constructed. As depicted in Fig. 1, domain Ω is divided by the interface Γ into two domains, Ω^+ and Ω^-. By using the Cartesian type of meshes, a complex interface will always cut through certain triangular Cartesian elements, as depicted in Fig. 1. We define those elements that intersect the interface as irregular elements. To make it clear, for an element to be an irregular element, the interface must cut through it. The domain E is the irregular domain that constitutes all the irregular elements. Due to the interface, we can partition the irregular domain E into E^+ and E^-, where $E^+ = \Omega^- \cap E$ and $E^- = \Omega^+ \cap E$, see Fig. 1. We also define irregular nodes as the vertices of an irregular element.

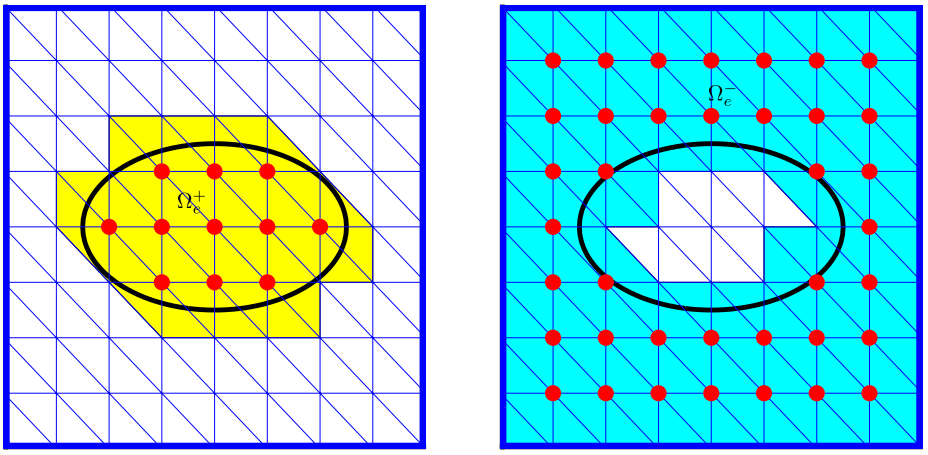

Fig. 2. Illustration of the extended domain Ω_e^+ (left chart) and the extended domain Ω_e^- (right chart) across the interface Γ.

Obviously, there are irregular elements in both Ω^+ and Ω^-, see the Right chart of Fig. 1. As the solution and the gradient of the solution may be discontinuous across the interface, commonly used basis functions cannot be directly employed for an irregular element. The immersed interface FEM solves this problem by employing two sets of linear polynomials on irregular elements [85].

In our MIB Galerkin method, we extend domains Ω^+ and Ω^- across the interface Γ as shown in Fig. 2. The extended domain Ω_e^+ constitutes the original domain Ω^+ and the irregular domain E^+, i.e., $\Omega_e^+ = \Omega^+ \cup E^+$. Similarly, extended domain Ω_e^- constitutes the original domain (Ω^-) and the irregular domain (E^-), i.e., $\Omega_e^- = \Omega^- \cup E^-$. As a result, we have two sets of overlapping elements, i.e., MIB elements, on the locations of irregular elements near the interface. We still employ the classical FEM basis on each MIB element, as if there were no interface. On MIB elements, we obtain two types of solution values, i.e., the fictitious values and the approximations of the true solution values. The fictitious values are used in discretization of the original PDE on the MIB elements, which gives rise to the approximations of the true solution values. Numerically, fictitious values are obtained through the enforcement of the interface jump conditions.

In the present formulation, we define extended discontinuous coefficient $\beta_e(\mathbf{x})$ over extended domains

$$\beta_e^+(\mathbf{x}) = \begin{cases} \beta^+(\mathbf{x}), & \forall \mathbf{x} \in \Omega^+ \\ \beta_{E^+}^+(\mathbf{x}), & \forall \mathbf{x} \in E^+, \end{cases} \tag{8}$$

and

$$\beta_e^-(\mathbf{x}) = \begin{cases} \beta^-(\mathbf{x}), & \forall \mathbf{x} \in \Omega^- \\ \beta_{E^-}^-(\mathbf{x}), & \forall \mathbf{x} \in E^-, \end{cases} \tag{9}$$

where $\beta_{E^+}^+(\mathbf{x})$ and $\beta_{E^-}^-(\mathbf{x})$ are the smooth extensions of the coefficient $\beta^+(\mathbf{x})$ and $\beta^-(\mathbf{x})$ over domains E^+ and E^-, respectively. Similarly, we also define extended singular source function $g_e(\mathbf{x})$ over extended domains

$$g_e^+(\mathbf{x}) = \begin{cases} g^+(\mathbf{x}), & \forall \mathbf{x} \in \Omega^+ \\ g_{E^+}^+(\mathbf{x}), & \forall \mathbf{x} \in E^+, \end{cases} \tag{10}$$

and

$$g_e^-(\mathbf{x}) = \begin{cases} g^-(\mathbf{x}), & \forall \mathbf{x} \in \Omega^- \\ g_{E^-}^-(\mathbf{x}), & \forall \mathbf{x} \in E^-, \end{cases} \tag{11}$$

where $g_{E^+}^+(\mathbf{x})$ and $g_{E^-}^-(\mathbf{x})$ are the smooth extensions of the functions $g^+(\mathbf{x})$ and $g^-(\mathbf{x})$ over domains $E^+(\mathbf{x})$ and $E^-(\mathbf{x})$, respectively. Based on the above setting, we further describe our MIB Galerkin formulation in the next subsection.

We emphasize that the situation illustrated in Fig. 1 is a special case. In general, the interface Γ may pass through the domain boundary $\partial\Omega$. Therefore the boundaries of both extended domains may also be part of the boundary of the computational domain, i.e., $\partial\Omega_e^- \cap \partial\Omega \neq \emptyset$ and $\partial\Omega_e^+ \cap \partial\Omega \neq \emptyset$.

2.3. MIB Galerkin formulation

Let $\mathcal{T}_h$ be a triangular partition of the domain Ω with mesh size h. The partition can be grouped into two sets of elements denoted by $\mathcal{T}_h^+ = \mathcal{T}_h \cap \Omega_e^+$ and $\mathcal{T}_h^- = \mathcal{T}_h \cap \Omega_e^-$, respectively. Obviously, $\mathcal{T}_h^+$ and $\mathcal{T}_h^-$ are finite element partitions for subdomains Ω_e^+ and Ω_e^-, respectively. For each triangle $K^+ \in \mathcal{T}_h^+$, denote by $P_j(K^+)$ a set of polynomials in K^+ with degree

no more than j. Similarly, for each triangle $K^- \in \mathcal{T}_h^-$, denote by $P_k(K^-)$ a set of polynomials in K^- with degree no more than k. We introduce two trial MIB element spaces as follows

$$U_h^+ := \left\{u_h^+ \in H^1(\Omega_e^+) : \ u_h^+|_{K^+} \in P_j(K^+), \ \forall K^+ \in \mathcal{T}_h^+\right\} \tag{12}$$

$$U_h^- := \left\{u_h^- \in H^1(\Omega_e^-) : \ u_h^-|_{K^-} \in P_k(K^-), \ \forall K^- \in \mathcal{T}_h^-\right\}. \tag{13}$$

For a given MIB element $K^+ \in \mathcal{T}_h^+$, let ϕ_i^+, $i = 1, \ldots, N^+$ be a set of basis functions for $P_j(K^+)$. Every u_h^+ has the following representation

$$u_h^+(\mathbf{x}) = \sum_{i=1}^{N^+} u_{h,i}^+ \phi_i^+(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_e^+, \tag{14}$$

where $\{u_{h,i}^+\}$ is a set of coefficients to be determined in solving the original PDE. Similarly, for a given MIB element $K^- \in \mathcal{T}_h^-$, let ϕ_i^-, $i = 1, \ldots, N^-$ be a set of basis functions for $P_k(K^-)$. Every u_h^- has the following representation

$$u_h^-(\mathbf{x}) = \sum_{i=1}^{N^-} u_{h,i}^- \phi_i^-(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega_e^-, \tag{15}$$

where $\{u_{h,i}^-\}$ is another set of coefficients to be determined in the MIB Galerkin algorithms.

We define two test spaces by

$$U_h^{0,+} := \left\{v^+ \in U_h^+ : \ v^+ = 0 \text{ on } \partial\Omega_e^+\right\} \tag{16}$$

$$U_h^{0,-} := \left\{v^- \in U_h^- : \ v^- = 0 \text{ on } \partial\Omega_e^-\right\}. \tag{17}$$

To solve Eqs. (1)–(4) in domain Ω_e^+, we have

$$-\int_{\Omega_e^+} \nabla \cdot (\beta_e^+(\mathbf{x}) \nabla u_h^+(\mathbf{x})) v^+ d\mathbf{x} = \int_{\Omega_e^+} g_e^+(\mathbf{x}) v^+ d\mathbf{x}, \quad \forall \mathbf{x} \in \Omega_e^+. \tag{18}$$

Integration by parts, we have

$$-\int_{\partial\Omega_e^+} \beta_e^+(\mathbf{x}) \frac{\partial u_h^+(\mathbf{x})}{\partial n^+} v^+ ds + \int_{\Omega_e^+} \beta_e^+(\mathbf{x}) \nabla u_h^+(\mathbf{x}) \cdot \nabla v^+ d\mathbf{x} = \int_{\Omega_e^+} g_e^+ v^+ d\mathbf{x}, \quad \forall \mathbf{x} \in \Omega_e^+. \tag{19}$$

Due to the property of the test space (16), we end up with the following approximation in domain Ω_e^+

$$\int_{\Omega_e^+} \beta_e^+(\mathbf{x}) \nabla u_h^+(\mathbf{x}) \cdot \nabla v^+ d\mathbf{x} = \int_{\Omega_e^+} g_e^+(\mathbf{x}) v^+ d\mathbf{x}, \quad \forall \mathbf{x} \in \Omega_e^+, \ \forall v^+ \in U_h^{0,+}. \tag{20}$$

Similarly, we have the following the finite element approximation in domain Ω_e^-

$$\int_{\Omega_e^-} \beta_e^-(\mathbf{x}) \nabla u_h^-(\mathbf{x}) \cdot \nabla v^- d\mathbf{x} = \int_{\Omega_e^-} g_e^-(\mathbf{x}) v^- d\mathbf{x}, \quad \forall (\mathbf{x}) \in \Omega_e^-, \ \forall v^- \in U_h^{0,-}. \tag{21}$$

Obviously, Eqs. (20) and (21) are not well posed in the normal sense. Similarly, our partitions $\mathcal{T}_h^+$ and $\mathcal{T}_h^-$ are not admissible in the normal sense either due to the overlapping feature of the MIB elements. To resolve these problems, we define a set of normal solutions U_h as a restriction of $\{u_h^+\}$ and $\{u_h^-\}$ to the domain Ω^+ and Ω^-, respectively

$$U_h := \left\{u_h = \begin{pmatrix} u_h^+(\mathbf{x}) : \forall \mathbf{x} \in \Omega^+ \\ u_h^-(\mathbf{x}) : \forall \mathbf{x} \in \Omega^-, \end{pmatrix}\right\}, \tag{22}$$

where $\{u_h\}$ are to be determined by Eqs. (20) and (21). The total number of unknowns in the set $\{u_h\}$ equals the total number of node points in the inner domain $(\Omega \backslash \partial\Omega)$. However, since Eqs. (20) and (21) are defined on extended domains, the number of unknowns in these two equations is larger than the total number of nodes in the inner domain $(\Omega \setminus \partial\Omega)$. To resolve this problem, we define two sets of fictitious values

$$F_h^+ := \left\{f_h^+(\mathbf{x}) = u_h^+(\mathbf{x}) : \forall \mathbf{x} \in E^+\right\} \tag{23}$$

$$F_h^- := \left\{f_h^-(\mathbf{x}) = u_h^-(\mathbf{x}) : \forall \mathbf{x} \in E^-\right\}. \tag{24}$$

In the MIB Galerkin method, the set of all fictitious values, denoted as $F_h = F_h^+ \cup F_h^-$, is determined by the interface jump conditions. Note that if an element node intersects with the interface Γ, both a fictitious value and an approximation solution are obtained on the node. Note that domain Ω^+ includes the interface while Ω^- does not. This affects the solution of the set of fictitious values on the interface. However, the role of Ω^+ and Ω^- can be interchanged based on the interface geometry and the convenience in solving the PDE. The procedure for the solution of F_h is described in Section 3.

3. Solution algorithms

3.1. Basis functions

In the present work, we consider a linear polynomial basis functions for both Ω_e^+ and Ω_e^-. Assume that there is a total of N^+ nodes in the inner computational domain ($\Omega_e^+ \setminus \partial\Omega$). At an arbitrary node $\mathbf{x}_j$, we employ a set of linear interpolation basis functions, $\{\phi_j^+(x, y)\}$, defined as

$$\phi_j^+(\mathbf{x}) = \begin{cases} a_{j,1} + b_{j,1}x + c_{j,1}y, & \forall (x, y) \in K_{j,1}^+ \\ a_{j,2} + b_{j,2}x + c_{j,2}y, & \forall (x, y) \in K_{j,2}^+ \\ \cdots & \\ a_{j,6} + b_{j,6}x + c_{j,6}y, & \forall (x, y) \in K_{j,6}^+, \end{cases} \quad j = 1, 2, \ldots, N^+ \tag{25}$$

where $\{K_{j,l}^+\}_{l=1,\ldots,6}$ are six triangular elements that share the jth node. As an interpolation basis function, $\phi_j^+(\mathbf{x})$ has the property

$$\phi_j^+(\mathbf{x}_k) = \begin{cases} 1, & k = j \\ 0 & k \neq j, \end{cases} \quad j, k = 1, 2, \ldots, N^+. \tag{26}$$

By applying interpolation property in Eq. (26) to six adjacent nodes of node $\mathbf{x}_j$, we can uniquely determine all coefficients in Eq. (25), namely $a_{j,1}, b_{j,1}, \ldots, c_{j,6}$. The set of linear interpolation basis functions, $\{\phi_j^-(\mathbf{x})\}$ is defined in domain Ω_e^- in a similar manner.

To simplify the notation, we define the following bilinear form

$$a(u_h, v_h) = (\beta \nabla u_h, \nabla v_h), \tag{27}$$

where the inner product (x, y) represents the integration of the product of xy over the region where the functions are defined. Therefore, we can express Eqs. (20) and (21) as,

$$a(u_h^+, v_h^+) = (g_e^+, v_h^+), \tag{28}$$

$$a(u_h^-, v_h^-) = (g_e^-, v_h^-). \tag{29}$$

For a regular node $\mathbf{x}_i \in \Omega_e^+ \setminus E$, we only need to consider six adjacent nodes. We denote these seven nodes as $i_1, i_2, \ldots, i_7$. We use the approximation in Eq. (14) to discretize Eq. (28) and the resulting nonzero components at global index i are

$$(a(\phi_{i_1}^+, \phi_i^+), a(\phi_{i_2}^+, \phi_i^+) \cdots a(\phi_{i_7}^+, \phi_i^+)) \cdot \begin{pmatrix} u_{h,i_1}^+ \\ u_{h,i_2}^+ \\ \cdots \\ u_{h,i_7}^+ \end{pmatrix} = (g_e^+, \phi_i^+), \tag{30}$$

where $\{u_{h,n}^+, n = i_1, i_2, \ldots, i_7\}$ are coefficients on seven related nodes. Similarly, for a regular node $\mathbf{x}_j \in \Omega_e^- \setminus E$,

$$(a(\phi_{j_1}^-, \phi_j^-), a(\phi_{j_2}^-, \phi_j^-) \cdots a(\phi_{j_7}^-, \phi_j^-)) \cdot \begin{pmatrix} u_{h,j_1}^- \\ u_{h,j_2}^- \\ \cdots \\ u_{h,j_7}^- \end{pmatrix} = (g_e^-, \phi_j^-), \tag{31}$$

where $\{u_{h,n}^-, n = j_1, j_2, \ldots, j_7\}$ are coefficients on seven related nodes. However, Eqs. (30) and (31) cannot be solved directly because of the lack of boundary condition on ∂E.

For an irregular node $\mathbf{x}_i \in E^-$, a similar Galerkin approximation can be obtained. However, fictitious values are now involved in the scheme

$$(a(\phi_{i_1}^+, \phi_i^+), a(\phi_{i_2}^+, \phi_i^+) \cdots a(\phi_{i_7}^+, \phi_i^+)) \cdot \begin{pmatrix} u_{h,i_1}^+ \\ u_{h,i_2}^+ \\ \cdots \\ f_{h,i_7}^+ \end{pmatrix} = (g_e^+, \phi_i^+), \tag{32}$$

where f_{h,i_7}^+ is a fictitious value. The number of fictitious values depends on the local geometry, i.e., the relation between the interface and the mesh. Similarly, for an irregular node $\mathbf{x}_i \in E^+$, we have matrix elements

$$(a(\phi_{j_1}^-, \phi_j^-), a(\phi_{j_2}^-, \phi_j^-) \cdots a(\phi_{j_7}^-, \phi_j^-)) \cdot \begin{pmatrix} u_{h,j_1}^- \\ u_{h,j_2}^- \\ \cdots \\ f_{h,j_7}^- \end{pmatrix} = (g_e^-, \phi_j^-). \tag{33}$$

Eqs. (30)–(33) describe all of the matrix elements in the present MIB Galerkin method.

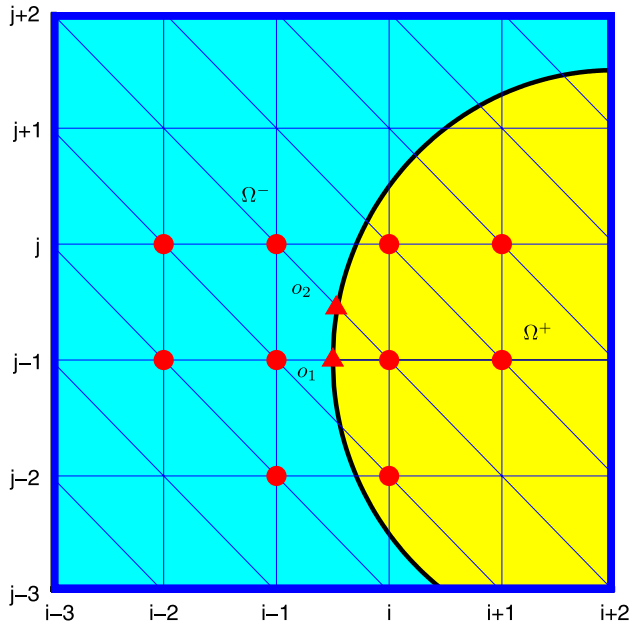

Fig. 3. The local geometry of a smooth interface intersecting the Cartesian based triangular mesh.

3.2. Algorithms for fictitious values

In our MIB collocation schemes, fictitious values are obtained by the smooth extensions of the original function values across the interface. We split a high dimensional problem into 1D representation. On each side of the interface, a second-order interpolation polynomial is defined with its coefficients to be determined by the function values at grid points and interface jump conditions [57]. We construct simple linear matrix equations for determining fictitious values. Through solving the matrix equations, we can represent the fictitious values via the approximation function values at grid nodes and jump conditions. A similar approach for dealing with fictitious values is employed in the present MIB Galerkin formulation. However, because of the use of the triangular mesh, we have to determine the fictitious values directly in a 2D setting. Therefore, we build 2D second-order polynomials as interpolation functions to determine fictitious values. Due to the use of triangular mesh, we have additional intersection points of the interface and the mesh located on the diagonal lines of the Cartesian grid. These additional intersection points are also utilized, together with the original set of intersection points to enforce interface jump conditions. A linear matrix equation is formed by employing function values at element nodes and interface jump conditions. We can then attain a representation for each fictitious value. The coefficients for fictitious values are solved with the least square method. A detailed description is given below.

3.2.1. Scheme for fictitious values

To simplify the notation, we denote a doubly indexed node (x_i, y_j) by (i, j). Considering the element labeled by vertices $(i, j-1)$, $(i-1, j-1)$ and $(i-1, j)$, there are two intersecting points between the interface and element edges, namely, points o_1 and o_2, see Fig. 3. The linear basis function at node $(i, j-1)$ involves two nodes, $(i-1, j-1)$ and $(i-1, j)$, on the other side of the interface. As a result, the discretized matrix equation (32) at node $(i, j-1)$ comprises two fictitious values $f_h^+(x_{i-1}, y_{j-1})$ and $f_h^+(x_{i-1}, y_j)$. Similarly, linear basis function at node $(i-1, j)$ involves two nodes, $(i, j-1)$ and (i, j), on the other side of the interface and the discretized matrix equation (33) comprises two fictitious values $f_h^-(x_i, y_{j-1})$ and $f_h^-(x_i, y_j)$. We need to determine the fictitious value at each vertex. Based on the previous MIB collocation scheme [57], in order to guarantee the second order accuracy, the approximation error caused by using fictitious values to represent the smooth extensions of the original function should be of $o(h^2)$. Therefore, we construct two 2D second order polynomial functions, one on domain Ω_e^+ and the other on domain Ω_e^-

$$u_h^{e+}(x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy, \quad (x, y) \in \Omega_e^+ \tag{34}$$

$$u_h^{e-}(x, y) = b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 y^2 + b_5 xy, \quad (x, y) \in \Omega_e^- \tag{35}$$

where $\{a_l\}_{l=0,1,\ldots,5}$ and $\{b_n\}_{n=0,1,\ldots,5}$ are a set of 12 coefficients. It is noted that both $u_h^{e+}(x, y)$ and $u_h^{e-}(x, y)$ are defined near the interface. In fact, $u_h^{e+}(x, y) = u_h^+(x, y)$ is a normal solution (i.e., $u_h^{e+}(x, y) \in U_h$) when $(x, y) \in \Omega^+$. However, $u_h^{e+}(x, y) = f_h^+(x, y)$ is a fictitious value when $(x, y) \in E^+$. Similarly, we have $u_h^{e-}(x, y) = u_h^-(x, y)$ when $(x, y) \in \Omega^-$, and $u_h^{e-}(x, y) = f_h^-(x, y)$ when $(x, y) \in E^-$. As discussed before, normal solutions are eventually obtained from the discretization matrix equations. Whereas, fictitious values are given by either Eq. (34) or Eq. (35).

In order to obtain fictitious values, we need to determine 12 coefficients in Eqs. (34) and (35), which in turn, requires at least 12 linear equations. As triangular elements used in this work are generated from a Cartesian grid, a major advantage of the present mesh is that the logical relations among all nodes are clear and straightforward. Therefore, we can make use of the information not only within an irregular element and its adjacent elements, but also from a desirable local region near the interface. This topological advantage gives us a great flexibility when dealing with the geometric singularities.

We first denote coefficients in Eqs. (34) and (35) as a vector C

$$C = (a_0, a_1, a_2, a_3, a_4, a_5, b_0, b_1, b_2, b_3, b_4, b_5)^T. \tag{36}$$

For node (i, j), we can have the linear equation

$$u_h^+(x_i, y_j) = (1, x_i, y_j, x_i^2, y_j^2, x_i y_j, 0, 0, 0, 0, 0, 0) \cdot C, \quad (x_i, y_j) \in \Omega_e^+ \tag{37}$$

$$u_h^-(x_i, y_j) = (0, 0, 0, 0, 0, 0, 1, x_i, y_j, x_i^2, y_j^2, x_i y_j) \cdot C, \quad (x_i, y_j) \in \Omega_e^-. \tag{38}$$

It is possible to uniquely determine coefficients C by 12 nodes selected from appropriate domains, which, however, is not an admissible solution to the original interface problem because one needs to enforce the interface jump conditions.

At intersecting point o_1 between the interface and the mesh, we denote the coordinate as (x_{o_1}, y_{o_1}) and the normal direction as (n_x, n_y). The derivatives can be expressed as

$$u_x^+ = (a_1 + 2a_3 x + a_5 y) \tag{39}$$

$$u_x^- = (b_1 + 2b_3 x + b_5 y) \tag{40}$$

$$u_y^+ = (a_2 + 2a_4 x + a_5 x) \tag{41}$$

$$u_y^- = (b_2 + 2b_4 x + b_5 x). \tag{42}$$

Note that due to Eqs. (39)–(42), one does not need to evaluate derivatives from nodes values. Such evaluations pose major complexity and difficulty in collocation algorithm development for geometric and interface singularities. Three interface jump conditions associated with node o_1 are given by

$$\begin{aligned}
[u]_{o_1} &= (1, x_{o_1}, y_{o_1}, x_{o_1}^2, y_{o_1}^2, x_{o_1} y_{o_1}, -1, -x_{o_1}, -y_{o_1}, -x_{o_1}^2, -y_{o_1}^2, -x_{o_1} y_{o_1}) \cdot C \\
[\beta u_n]_{o_1} &= (0, \beta^+ n_x, \beta^+ n_y, 2\beta^+ n_x x_{o_1}, 2\beta^+ n_y y_{o_1}, \beta^+ (n_x y_{o_1} + n_y x_{o_1}), \\
&\quad 0, -\beta^- n_x, -\beta^- n_y, -2\beta^- n_x x_{o_1}, -2\beta^- n_y y_{o_1}, -\beta^- (n_x y_{o_1} + n_y x_{o_1})) \cdot C \\
[u_\tau]_{o_1} &= (0, -n_y, n_x, -2n_y x_{o_1}, 2n_x y_{o_1}, -n_y y_{o_1} + n_x x_{o_1}, 0, n_y, -n_x, 2n_y x_{o_1}, -2n_x y_{o_1}, n_y y_{o_1} - n_x x_{o_1}) \cdot C.
\end{aligned} \tag{43}$$

As shown in Fig. 3, we can employ two intersecting points o_1 and o_2 to attain a total of six linear equations from interface jump conditions. Nodes play important roles in the construction of the fictitious scheme. We assign weights to the nodes surrounding the triangle of interest. In this case, the triangle is constructed by vertex $(i-1, j-1)$, $(i-1, j)$ and $(i, j-1)$ as demonstrated in Fig. 3. The weight for a node is defined as the summation of the distances between the node and three vertices from the triangle of interest. In each subdomain, nodes are chosen according to their weights. A smaller weight has a higher priority. Combining with the jump conditions from the intersection points, we can build up a linear matrix equation

$$A \cdot C = B, \tag{44}$$

where A is a 16×12 matrix, and B is a vector containing function values at nearby nodes and interface jump conditions. Vector C comprises coefficients to be determined. As the linear matrix equation is over-determined, we make use of the least square method to compute an inverse matrix of A so that the vector C is represented as

$$C = (A'A)^{-1} A' B. \tag{45}$$

We decompose B into two vectors B_1 and B_2. Vector B_1 has ten components, i.e., function values at nodes. For the situation illustrated in Fig. 3, vector B_1 can be expressed as

$$B_1 = (u_h^-(x_{i-2}, y_{j-1}), u_h^-(x_{i-2}, y_j), u_h^-(x_{i-1}, y_{j-2}), u_h^-(x_{i-1}, y_{j-1}), u_h^-(x_{i-1}, y_j), \tag{46}$$

$$u_h^+(x_i, y_{j-1}), u_h^+(x_i, y_j), u_h^+(x_i, y_{j-2}), u_h^+(x_{i+1}, y_{j-1}), u_h^+(x_{i+1}, y_j))^T. \tag{47}$$

Vector B_2 has six components, i.e., the jump conditions at two interface intersecting points. For the situation illustrated in Fig. 3, vector B_2 is given by

$$B_2 = ([u]_{o_1}, [\beta u_n]_{o_1}, [u_\tau]_{o_1}, [u]_{o_2}, [\beta u_n]_{o_2}, [u_\tau]_{o_2})^T. \tag{48}$$

If we denote a 12×16 matrix G as

$$G = (A'A)^{-1} A', \tag{49}$$

we can divide G into two matrices $G1$ and $G2$. Then we can rewrite vector coefficient C as

$$\{C\}_{12\times 1} = (\{G_1\}_{12\times 10}, \{G_2\}_{12\times 6}) \cdot \begin{pmatrix} \{B_1\}_{10\times 1} \\ \{B_2\}_{6\times 1} \end{pmatrix} \tag{50}$$

$$\{C\}_{12\times 1} = \{G_1\}_{12\times 10}\{B_1\}_{10\times 1} + \{G_2\}_{12\times 6}\{B_2\}_{6\times 1}. \tag{51}$$

After the determination of vector C, fictitious values are calculated. In Fig. 3, for the irregular triangle element with vertices $(i-1, j-1)$, $(i-1, j)$ and $(i, j-1)$, the fictitious value can be represented as

$$f_h^+(x_{i-1}, y_{j-1}) = (1, x_{i-1}, y_{j-1}, x_{i-1}^2, y_{j-1}^2, x_{i-1}y_{j-1}, 0, 0, 0, 0, 0, 0) \cdot C \tag{52}$$

$$f_h^+(x_{i-1}, y_j) = (1, x_{i-1}, y_j, x_{i-1}^2, y_j^2, x_{i-1}y_j, 0, 0, 0, 0, 0, 0) \cdot C \tag{53}$$

$$f_h^-(x_i, y_{j-1}) = (0, 0, 0, 0, 0, 0, 1, x_i, y_{j-1}, x_i^2, y_{j-1}^2, x_i y_{j-1}) \cdot C. \tag{54}$$

If we denote T as a location coefficient vector, then a fictitious value can be written as $f_h = T \cdot C$. By using Eqs. (50) and (51), we have a general expression for a fictitious value

$$\begin{aligned} f_h &= \{T\}_{1\times 12} \cdot (\{G_1\}_{12\times 10}\{B_1\}_{10\times 1} + \{G_2\}_{12\times 6}\{B_2\}_{6\times 1}) \\ &= \{T\}_{1\times 12}\{G_1\}_{12\times 10}\{B_1\}_{10\times 1} + \{T\}_{1\times 12}\{G_2\}_{12\times 6}\{B_2\}_{6\times 1}. \end{aligned} \tag{55}$$

Once the mesh is set up for a given interface, the location information for each node and each interface intersecting point is known. As a result, vectors B_2 and T are known. We can calculate the matrices G_1 and G_2 from Eq. (49). Vector B_1 contains approximation function values at mesh nodes. As we have the expression of the fictitious values, we just need to insert this expression into the discretization matrix of the governing equation, Eq. (32) or Eq. (33), and solve the linear matrix equation to obtain the solution at all the node points.

Normally, for a simple geometry, the matrix $A'A$ in Eq. (49) is invertible. However, in some special case, this condition is not satisfied, especially when there is a nonsmooth interface [55]. To guarantee the invertibility, an improved scheme is employed, in which we systematically add in pairs of nodes from both Ω^+ and Ω^-. In order to make our scheme robust, these nodes are always chosen from the ones with lowest weight values. To avoid confusion, weights are defined as the summation of the distances to three triangle vertices as stated above.

The basic process can be summarized as follows. We pick up the minimal coordinate index in the irregular triangle of interest. For instance, in Fig. 3, the index is $(i-1, j-1)$. We calculate weights for all the nodes in a predetermined domain, say $\{(k, l)|(i-1)-5 < k < (i-1)+5;\ (j-1)-5 < l < (j-1)+5\}$. In each subdomain ($\Omega^+$ or Ω^-), the priority of a given node is determined by its weight. Usually, the first five nodes with lowest weights from each subdomain are selected to construct our fictitious scheme. However, the matrix $A'A$ may not be invertible for some special geometries. To resolve this problem, we add one additional node according to its priority in each of Ω^+ and Ω^-. This procedure is repeated until the matrix $A'A$ is invertible.

3.2.2. Scheme for the derivative of the solution

As the present triangular mesh is based on the Cartesian grid with additional diagonal lines, the derivative of the solution on mesh nodes can be approximated by central schemes. For a regular node (i, j), the derivative of the solution can be written as

$$(u_h^+)_x(x_i, y_j) = \frac{u_h^+(x_{i+1}, y_j) - u_h^+(x_{i-1}, y_j)}{2dx}, \tag{56}$$

$$(u_h^+)_y(x_i, y_j) = \frac{u_h^+(x_i, y_{j+1}) - u_h^+(x_i, y_{j-1})}{2dy}, \tag{57}$$

where dx and dy are the grid sizes of the mesh at x and y directions, respectively. If node (i, j) is an irregular one as the situation in Fig. 3, then adjacent nodes $(i-1, j)$ and $(i, j+1)$ locate on the other side of the interface. We need to make use of fictitious values and rewrite the central scheme as

$$(u_h^+)_x(x_i, y_j) = \frac{u_h^+(x_{i+1}, y_j) - f_h^+(x_{i-1}, y_j)}{2dx}, \tag{58}$$

$$(u_h^+)_y(x_i, y_j) = \frac{f_h^+(x_i, y_{j+1}) - u_h^+(x_i, y_{j-1})}{2dy}. \tag{59}$$

It is seen that if we calculate solutions u_h^+ and u_h^-, fictitious values can be attained through Eq. (55).

For convenience, let denote $h = dx = dy$. As we employ the second order polynomial to approximate the fictitious values in Eq. (34), we have the relation

$$u_h^{e+}(x_i, y_j) = u(x_i, y_j) + O(h^3). \tag{60}$$

This relation can be used to evaluate the approximation order of the derivative scheme for the irregular node point.

$$\begin{aligned}(u_h^+)_x(x_i, y_j) &= \frac{u_h^+(x_{i+1}, y_j) - f_h^+(x_{i-1}, y_j)}{2h} \\ &\leq \frac{|u_h^+(x_{i+1}, y_j) - u(x_{i+1}, y_j)| + |u(x_{i-1}, y_j) - f_h^+(x_{i-1}, y_j)| + |u(x_{i+1}, y_j) - u(x_{i-1}, y_j)|}{2h} \\ &= u_x(x_i, y_j) + O(h^2), \end{aligned} \tag{61}$$

where $u(x_{i+1}, y_j)$ and $u(x_{i-1}, y_j)$ are exact values and $u_x(x_i, y_j)$ is the exact derivative. We have taken the advantage of an error cancellation due to the symmetry of the central scheme. Similar estimations can be obtained for $(u_h^+)_y(x_i, y_j)$, $(u_h^-)_x(x_i, y_j)$ and $(u_h^-)_y(x_i, y_j)$. Therefore, we can attain the second order accuracy for the derivatives even at irregular nodes.

4. Numerical studies

In this section, we examine the performance of the proposed MIB Galerkin finite element method for solving the Poisson equation with material interfaces, geometric singularities and solution singularities. To avoid misunderstanding, the L_∞ error of the derivative of the solution is defined as:

$$L_\infty = \max_{\forall (x_i, y_j) \in \Omega \setminus \partial\Omega} \{|u_x(x_i, y_j) - (u_h)_x(x_i, y_j)|, |u_y(x_i, y_j) - (u_h)_y(x_i, y_j)|\}, \tag{62}$$

where $u_x(x_i, y_j)$ and $u_y(x_i, y_j)$ are exact derivative values. We employ the Krylov space based pre-conditioned conjugate gradient solver to solve the algebraic equations generated by the MIB Galerkin method. To avoid confusion, as the Cartesian grid is used, we can simply define the L_2 error as the root mean square of the error values on the grid points. As in our previous studies, we employ a number of standard tests to validate our new MIB Galerkin method. Case 1 is designed to demonstrate the stability of our algorithm for highly oscillatory solutions. The interfaces with large curvatures are investigated in Case 2 and Case 3. Case 4 is devoted to validate the proposed MIB Galerkin method for the geometry singularities, i.e., the tips of a pentagon. Case 5 is utilized to demonstrate the robustness of the present method for very complex interface geometries. For all the above cases, the second order accuracy can be attained in the L_∞ error for both the solution and the gradient of the solution.

In the last three cases, we demonstrate the ability of the proposed MIB Galerkin finite element method for a class of challenging problems, for which existing methods encounter difficulties [71]. Case 6 is designed to explore the behavior of the present method for solution singularity associated with geometric singularity. Case 7 is employed to examine the performance of the proposed algorithm for a C^1 continuous interface geometry. We keep the solution in domain Ω^+ as a constant, so it has good quality and belongs to C^∞ space. The solutions in domain Ω^- are chosen from three different situations, i.e., C^2, C^1 and H^2. In Case 8, a Lipschitz continuous interface geometry is employed to further test our method, while the same set of solutions are employed. Numerical experiments are carried out to show the advantages and limitations of the present method. Essentially, our algorithm can incorporate the interface information efficiently. The same order of accuracy is attained for both C^1 continuous and Lipschitz continuous interface geometries. When the solution in Ω^- is C^2 continuous, essentially second order accuracy can be achieved in the L_∞ error for both the solution and the derivative of the solution. When the solution in domain Ω^- is only C^1 continuous or H^2 continuous, solutions remain essentially second order accurate, while the accuracy of the derivative of the solution deteriorates to the first order.

4.1. Complex interface geometries

- **Case** 1(a): We first investigate a classical circular interface problem. The 2D Poisson equations (1)–(4) is solved on domain $[-1, 1] \times [-1, 1]$. To specify the interface, we design level set function $\phi(x, y)$

$$\phi(x, y) = r^2 - (x^2 + y^2), \tag{63}$$

  where the radius $r = 2/3$. The interface is given as $\Gamma = \{(x, y)|\phi = 0, \forall (x, y) \in \Omega\}$. Two subdomains are $\Omega^+ = \{(x, y)|\phi \geq 0, \forall (x, y) \in \Omega\}$, and $\Omega^- = \{(x, y)|\phi < 0; \forall (x, y) \in \Omega\}$. The discontinuous coefficients are given by

$$\beta^+(x, y) = 2, \quad \text{and} \quad \beta^-(x, y) = 3. \tag{64}$$

  The solution in two different subdomains is chosen as

$$u^+(x, y) = 7 + \sin(4\pi x)\sin(4\pi y), \quad \text{and} \quad u^-(x, y) = 5e^{-(x^2+y^2)}. \tag{65}$$

  The source term $g(x, y)$ and boundary value $g_b(x, y)$ can be given accordingly.
- **Case** 1(b): We consider Case 1(a) with following modifications. We set the domain to $[-5, 5] \times [-5, 5]$, and the radius to $r = 10/3$. The discontinuous coefficients are given by

$$\beta^+(x, y) = 2, \quad \text{and} \quad \beta^-(x, y) = 3. \tag{66}$$

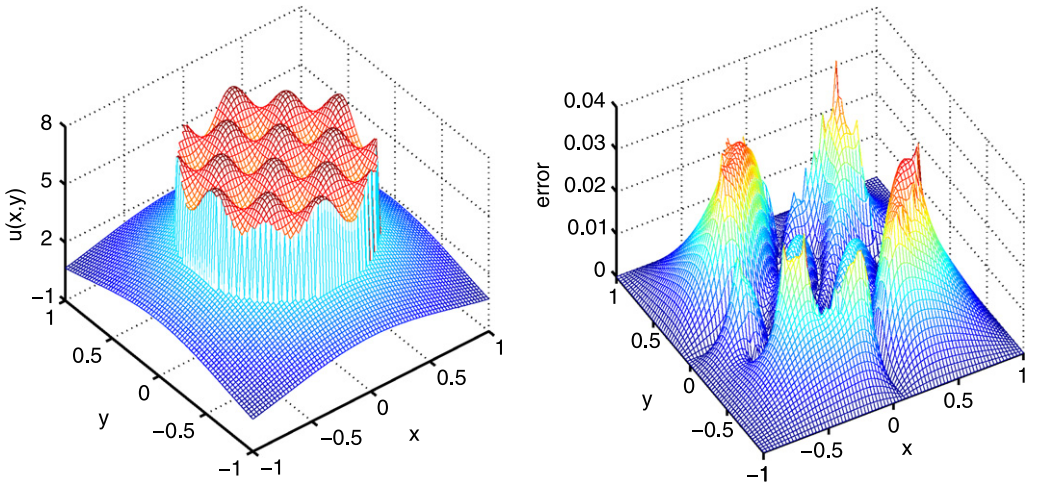

Fig. 4. The computed solution on an 80×80 mesh (left chart) and the L_∞ error (right chart) for Case 1(a).

Table 1
Numerical errors and convergence orders for highly oscillatory solutions (Case 1(a)).

$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
20×20	7.250e−1		5.507	
40×40	1.684e−1	2.11	1.527	1.85
80×80	4.434e−2	1.93	4.281e−1	1.83
160×160	6.078e−3	2.87	1.369e−1	1.64

The solution at two different subdomains is oscillatory:

$$u^+(x, y) = 7 + \sin(4\pi x)\sin(4\pi y), \quad \text{and} \quad u^-(x, y) = 5 + \sin(3\pi x)\sin(3\pi y). \tag{67}$$

In Case 1(a), we test the present method for highly oscillatory solutions, see Fig. 4. Our results are demonstrated in Table 1. From the table, it is seen that even for the L_∞ error, essentially second accuracy is achieved for both the solution and the gradient of the solution in the whole computational domain. We omit the data of the L_2 error for the reason that it is more difficult to attain the second accuracy in the L_∞ error. Indeed, it is found that the L_2 error is even smaller. In calculating the derivative of the solution, we employ the fictitious values for irregular nodes as described in Section 3.2.2. Fig. 4 also depicts the numerical solution and error on an 80×80 mesh.

The present method is tested on much denser grids in Case 1(b). In our computation, we use as many as $N = 1050$ nodes in the each direction of the 2D domain. Fig. 5 depicts the L_∞ and L_2 errors of the present method. We use the least square fitting to estimate convergence orders. It is found that the L_∞ order is 2.25 and the L_2 order is 2.21.

- **Case** 2(a): In this case, the 2D Poisson equation (1)–(4) is solved on domain $[-1, 1] \times [-1, 1]$. The interface is parametrized with polar angle θ

$$r = \frac{1}{2} + \frac{1}{7}\sin(5\theta), \tag{68}$$

and the discontinuous coefficients are given by

$$\beta^+(x, y) = (xy + 2)/5, \quad \text{and} \quad \beta^-(x, y) = (x^2 - y^2 + 3)/7. \tag{69}$$

The solution at two different subdomains is designed as

$$u^+(x, y) = x^2 + y^2, \quad \text{and} \quad u^-(x, y) = \sin(\pi x)\sin(\pi y). \tag{70}$$

The source term $g(x, y)$ and boundary value $g_b(x, y)$ can be given accordingly.

- **Case** 2(b): We consider Case 2(a) with following modifications. We set the domain to $[-5, 5] \times [-5, 5]$. The physical interface Γ is defined by the polar equation:

$$r = \frac{5}{2} + \frac{5}{7}\sin(5\theta). \tag{71}$$

The discontinuous coefficients is given by

$$\beta^+(x, y) = 5, \quad \text{and} \quad \beta^-(x, y) = 2. \tag{72}$$

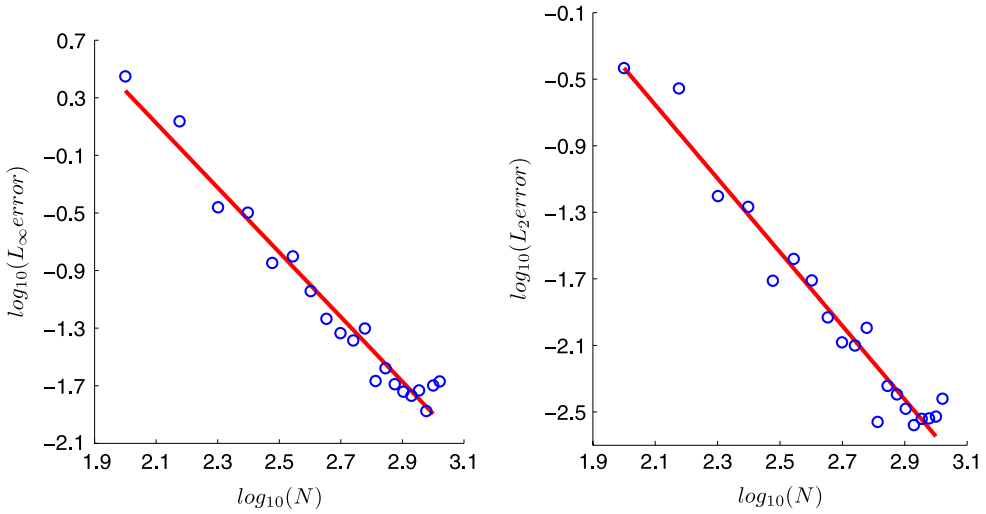

Fig. 5. The L_∞ error (left chart) and L_2 error (right chart) of computed solutions for Case 1(b).

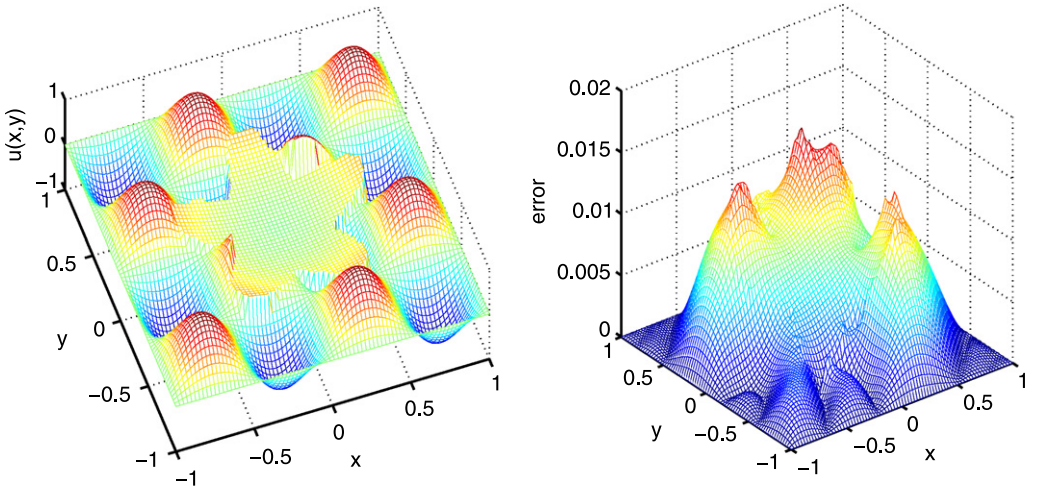

Fig. 6. The computed solution on an 80 × 80 mesh (left chart) and the L_∞ error (right chart) for Case 2(a).

Table 2
Numerical errors and convergence orders for a five-petal flower pattern interface geometry (Case 2(a)).

$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
20 × 20	2.072e−1		1.144	
40 × 40	6.355e−2	1.71	2.883e−1	1.99
80 × 80	1.571e−2	2.02	8.666e−2	1.73
160 × 160	3.575e−3	2.14	2.194e−2	1.98

The solution at two different subdomains is given by

$$u^+(x, y) = x^2 + y^2, \quad \text{and} \quad u^-(x, y) = \sin(3\pi x)\sin(3\pi y). \tag{73}$$

For Case 2(a), the interface is a five-petal flower pattern. The coefficients are chosen as two different functions in two subdomains. The results are listed in Table 2 and the numerical solution on an 80 × 80 mesh is depicted in Fig. 6. We have achieved second order accuracy in the L_∞ error for the solution and the derivative of the solution.

The present method is tested on much denser grids in Case 2(b). In our computation, we use as many as $N = 1050$ nodes in the each direction of the 2D domain. Fig. 7 depicts the L_∞ and L_2 errors of the present method. The least square fitting is used to estimate convergence orders. It is found that the L_∞ order is 2.14 and the L_2 order is 1.95.

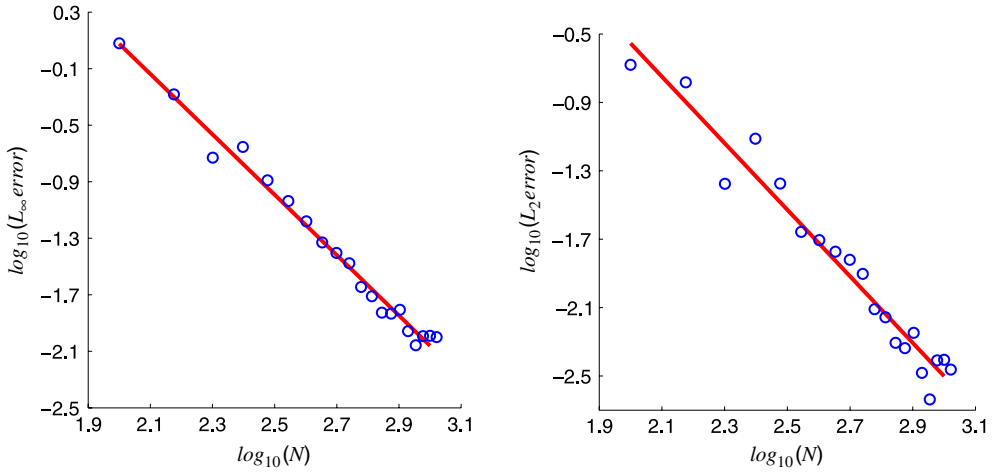

Fig. 7. The L_∞ error (left chart) and L_2 error (right chart) of computed solutions for Case 2(b).

- **Case** 3(a): In this case, the 2D Poisson equation (1)–(4) is solved on domain $[-1, 1] \times [-1, 1]$. The physical interface Γ is defined as

$$r = \frac{1}{2}\left(1 + \frac{1}{2}\sin(6\theta)\right). \tag{74}$$

The discontinuous coefficients are given by

$$\beta^+(x, y) = 1, \quad \text{and} \quad \beta^-(x, y) = 2 + \sin(x + y). \tag{75}$$

The solution at two different subdomains is designed as

$$u^+(x, y) = 5 + \sin(x^2 + y^2), \quad \text{and} \quad u^-(x, y) = \sin(\pi x)\sin(\pi y). \tag{76}$$

The source term $g(x, y)$ and boundary value $g_b(x, y)$ can be given accordingly.

- **Case** 3(b): We consider Case 3(a) with following modifications. The domain is set to $[-5, 5] \times [-5, 5]$. The physical interface Γ is defined as

$$r = \frac{5}{2}\left(1 + \frac{1}{2}\sin(6\theta)\right). \tag{77}$$

The discontinuous coefficients is the same as Eq. (75). The solution at two different subdomains is given by

$$u^+(x, y) = 10 + \sin(2\pi x)\sin(2\pi y), \quad \text{and} \quad u^-(x, y) = 7 + \sin(3\pi x)\sin(3\pi y). \tag{78}$$

In Case 3, we have a six-petal flower shaped interface geometry. This case is designed to test the proposed method for dealing with large curvatures. The solution has jumps at interfaces. Table 3 lists main results for Case 3(a). We have achieved second order accuracy in the L_∞ error for both the solution and the derivative of the solution. The numerical solution and error on an 80×80 mesh are depicted in Fig. 8.

We test the behavior of the present method on a set of dense grids in Case 3(b). In our computation, we use as many as $N = 1050$ nodes in the each direction of the 2D domain. Fig. 9 shows the L_∞ and L_2 errors of the present method. The least square fitting is used to compute convergence orders. It is found that L_∞ and L_2 orders are 1.98 and 2.02, respectively. This example, together with Case 1(b) and Case 2(b), indicates that the proposed method is able to maintain its convergence order when the grid is very dense for various interface geometries. In fact, for present method, the geometric complexity is not a problem. However, one has to make the solution sufficiently complex and highly oscillatory. Otherwise, for simple solutions, the method quickly reaches its optional accuracy at a low resolution grid and stops improving further when the grid resolution is increased.

Case 4. In this case, the 2D Poisson equations (1)–(4) is solved on domain $[-1, 1] \times [-1, 1]$. To specify the interface, we design the level set function $\phi(r, \theta)$

$$\phi(r, \theta) = \begin{cases} \dfrac{R\sin(\theta_t/2)}{\sin(\theta_t/2 + \theta - \theta_r - 2\pi(i-1)/5)} - r, & \theta_r + \pi(2i-2)/5 \le \theta < \theta_r + \pi(2i-1)/5 \\ \dfrac{R\sin(\theta_t/2)}{\sin(\theta_t/2 - \theta + \theta_r + 2\pi(i-1)/5)} - r, & \theta_r + \pi(2i-3)/5 \le \theta < \theta_r + \pi(2i-2)/5, \end{cases} \tag{79}$$

Table 3
Numerical errors and convergence orders for a six-petal flower shaped interface geometry (Case 3(a)).

$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
20×20	2.614e−2		2.192e−1	
40×40	5.435e−3	2.27	5.568e−2	1.98
80×80	1.546e−3	1.81	1.618e−2	1.78
160×160	3.335e−4	2.21	3.655e−3	2.15

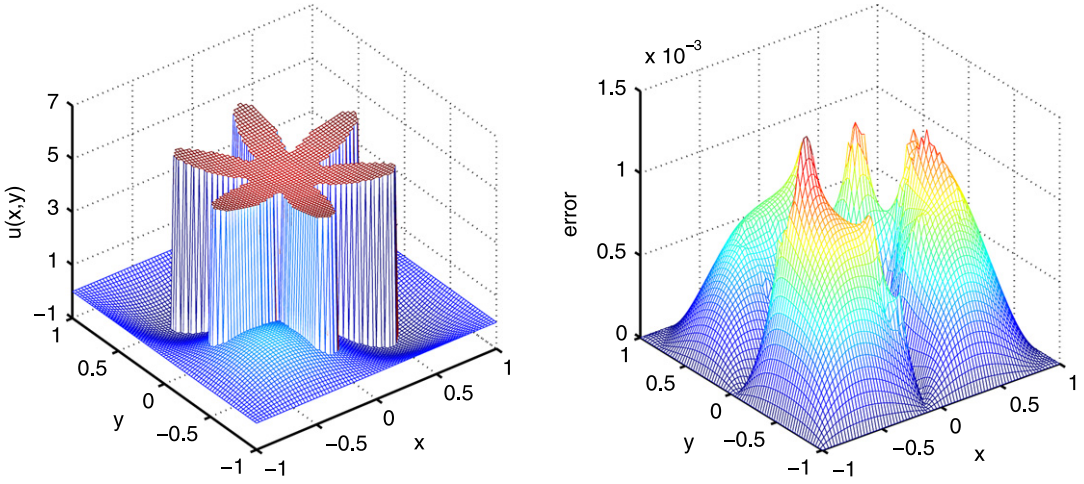

Fig. 8. The computed solution on an 80×80 mesh (left chart) and the L_∞ error (right chart) for Case 3(a).

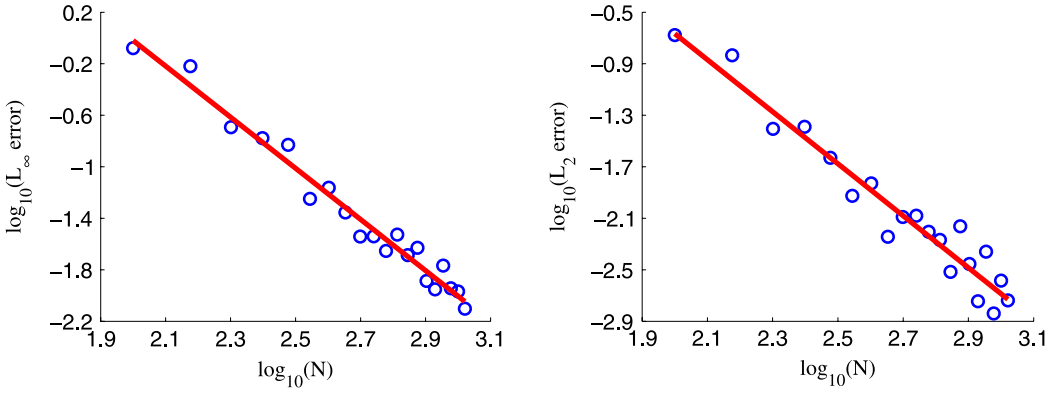

Fig. 9. The L_∞ error (left chart) and L_2 error (right chart) of computed solutions for Case 3(b).

where $\theta_t = \pi/5, \theta_r = \pi/7, R = 6/7$ and $i = 1, 2, \ldots, 5$. The discontinuous coefficients are given by

$$\beta^+(x, y) = 5, \quad \text{and} \quad \beta^-(x, y) = 3. \tag{80}$$

The solution at two different subdomains is designed as

$$u^+(x, y) = 7 + x^2 + y^2, \quad \text{and} \quad u^-(x, y) = \sin(3\pi x)\sin(3\pi y). \tag{81}$$

The source term $g(x, y)$ and boundary value $g_b(x, y)$ can be given accordingly.

In this case, the interface has a pentagon shape. It is designed to examine the performance of the proposed MIB Galerkin method for handling geometry singularities. As discussed, the present MIB Galerkin formulation deals with nonsmooth interfaces in a manner that differs much from the original MIB collocation formulation. It is known that the original MIB collocation formulation is able to deal with arbitrarily complex interfaces and nonsmooth interfaces. It is important to know whether the present MIB Galerkin formulation can do the same. The main results are listed in Table 4 and the numerical solution on an 80×80 mesh is depicted in Fig. 10. It is seen that the MIB Galerkin formulation has achieved the second order accuracy in the L_∞ error for the solution and the derivative of the solution.

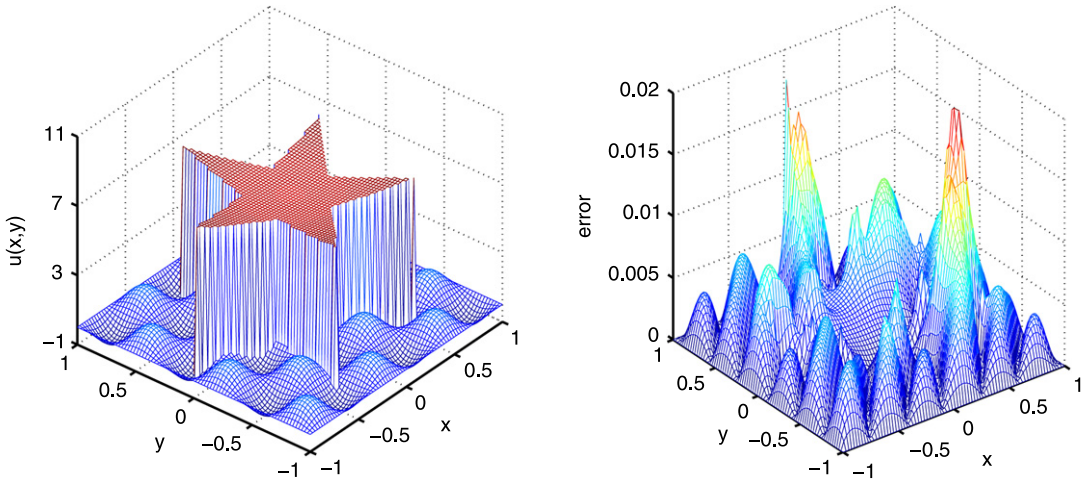

Fig. 10. The computed solution on an 80 × 80 mesh (left chart) and the L_∞ error (right chart) for Case 4.

Table 4
Numerical errors and convergence orders for a pentagon shape interface geometry (Case 4).

$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
20 × 20	6.374e−1		9.492	
40 × 40	9.271e−2	2.78	1.158	3.04
80 × 80	2.001e−2	2.21	3.180e−1	1.86
160 × 160	4.641e−3	2.11	6.372e−2	2.32

Table 5
Numerical errors and convergence orders for a jigsaw puzzle-like interface geometry (Case 5).

$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
20 × 20	1.785e−1		5.873e−1	
40 × 40	1.779e−2	3.33	1.411e−1	2.06
80 × 80	3.114e−3	2.51	3.451e−2	2.03
160 × 160	9.395e−4	1.73	1.349e−2	1.36

Case 5. We further consider a classical test example. In this case, the 2D Poisson equations (1)–(4) is solved on domain $[-1, 1] \times [0, 3]$. The jigsaw puzzle-like interface Γ is given as

$$\Gamma : \begin{cases} x(\theta) = \dfrac{6}{10}\cos(\theta) - \dfrac{3}{10}\cos(3\theta) \\ y(\theta) = \dfrac{2}{3} + \dfrac{7}{10}\sin(\theta) - \dfrac{7}{100}\sin(3\theta) + \dfrac{2}{10}\sin(7\theta). \end{cases} \tag{82}$$

The discontinuous coefficients are given by

$$\beta^+(x, y) = 1, \quad \text{and} \quad \beta^-(x, y) = 2 + \sin(x + y). \tag{83}$$

The solution at two different subdomains is designed as

$$u^+(x, y) = 5 + \sin(x^2 + y^2), \quad \text{and} \quad u^-(x, y) = \frac{1}{10}(x^2 + y^2). \tag{84}$$

The source term $g(x, y)$ and boundary value $g_b(x, y)$ can be given accordingly.

We use this case to demonstrate the robustness of the proposed MIB Galerkin method for handling arbitrarily complex interface geometries. The interface is of jigsaw puzzle-like shape. The results are listed in Table 5. For this kind of complex geometry, it is seen that the present method can still attain the second order accuracy in the L_∞ error for the solution and the derivative of the solution. In Fig. 11, we show the numerical solution and the L_∞ error on an 80 × 80 mesh.

4.2. Low solution regularities

In this subsection, we consider a class of challenging elliptic interface problems for which the existing methods encounter difficulties [71]. In these interface problems, the solution is of low regularity. Three test examples, Case 6, Case 7 and Case 8,

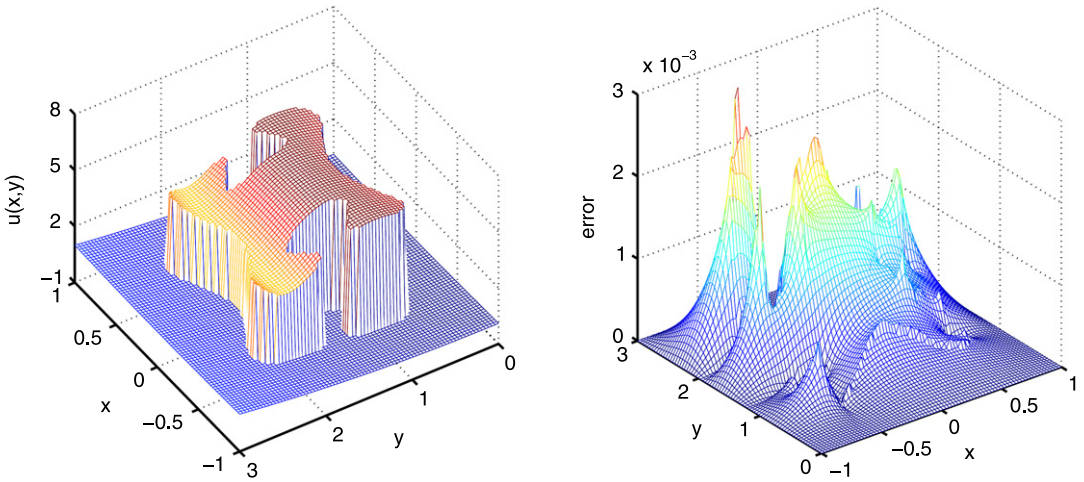

Fig. 11. The computed solution on an 80×80 mesh (left chart) and the L_∞ error (right chart) for Case 5.

Table 6
Numerical errors and convergence orders for a corner interface geometry (Case 6).

$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
20×20	4.156e−1		3.491	
40×40	5.392e−2	2.95	6.160e−1	2.50
80×80	1.023e−2	2.40	1.757e−1	1.81
160×160	2.721e−3	1.91	8.649e−2	1.02

are utilized to examine the stability and robustness of our algorithm for different regularities of the interface and the solution. Two types of interfaces, namely, C^1 continuous and Lipschitz continuous, are employed in the present study. In Case 6, the solution singularity is due to corner instability. In cases 7 and 8, the solution in domain Ω^+ is designed as a constant, while the solution in domain Ω^- is chosen from three different spaces, namely, the C^2, C^1 and H^2 spaces. For these problems, the state of the art is established by recently developed Wang–Ye Galerkin method, which achieves essentially second order convergence for the solution and the first order convergence in the gradient of the solution in the L_∞ error [77]. Numerical tests are carried out for the MIB Galerkin method to examine its accuracy and convergence order. The details are given below.

Case 6. In this case, the 2D Poisson equations (1)–(4) is solved on domain $[-1, 1] \times [-1, 1]$. To specify the interface, we design the level set function $\phi(x, y)$ for a corner interface

$$\phi(x,y): \begin{cases} -3, & x < 0 \\ -\left(y - \dfrac{x}{\sqrt{3}}\right)\left(y + \dfrac{x}{\sqrt{3}}\right), & x \geq 0. \end{cases} \tag{85}$$

The discontinuous coefficients are given by

$$\beta^+(x,y) = 3, \quad \text{and} \quad \beta^-(x,y) = 2. \tag{86}$$

The solution at two different subdomains is designed as

$$u^+(r,\theta) = r^{\frac{2}{3}} \sin\left(\frac{2}{3}\theta\right), \quad \text{and} \quad u^-(x,y) = -1 + \sin(3\pi x)\sin(3\pi y). \tag{87}$$

Here the interface is of Lipschitz continuity. The solution is H^2, but its derivative is singular at the corner. The main results are listed in Table 6 and the numerical solution on an 80×80 mesh is depicted in Fig. 12.

Case 7. In this case, the 2D Poisson equations (1)–(4) is solved on domain $[-1, 1] \times [-1, 1]$. The interface Γ is of C^1 continuous

$$\phi(x,y): \begin{cases} y - 2x, & \forall x + y > 0 \\ y - (2x + x^2), & \forall x + y \leq 0. \end{cases} \tag{88}$$

The discontinuous coefficients are given by piecewise functions

$$\beta^+(x,y) = (xy + 2)/5, \quad \text{and} \quad \beta^-(x,y) = (x^2 - y^2 + 3)/7. \tag{89}$$

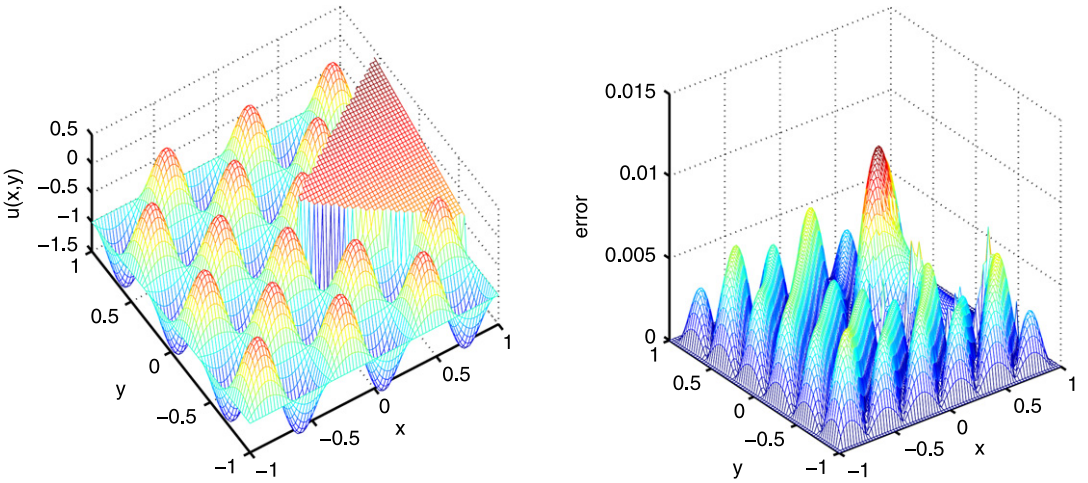

Fig. 12. The computed solution on an 80 × 80 mesh (left chart) and the L_∞ error (right chart) for Case 6.

The solution at two different subdomains is designed as,

- **Case** 7(a):

$$u^+(x,y) = 8, \tag{90}$$

$$u^-(x,y) = \begin{cases} x+y, & x+y>0 \\ \sin(x+y), & x+y\le 0. \end{cases} \tag{91}$$

This solution is of C^2 continuous.

- **Case** 7(b):

$$u^+(x,y) = 8, \tag{92}$$

$$u^-(x,y) = \begin{cases} x+y+1, & x+y>0 \\ \sin(x+y)+\cos(x+y), & x+y\le 0. \end{cases} \tag{93}$$

This solution is of C^1 continuity.

- **Case** 7(c):

$$u^+(x,y) = 8, \tag{94}$$

$$u^-(x,y) = (x^2+y^2)^{5/6} + \sin(x+y). \tag{95}$$

This solution is of H^2 continuity.

The source term $g(x,y)$ and boundary value $g_b(x,y)$ can be determined accordingly for all cases.

In Case 7, the interface is C^1 continuous. The solution in domain Ω^+ is designed as a constant, so it belongs to C^∞ space and has a good quality of smoothness. The solution in domain Ω^- is chosen from the C^2 space in Case 7(a). From the results listed in Table 7, it can be seen that the second order accuracy in the L_∞ error is attained for the solution. For the gradient of the solution, the test result indicates a convergence order of 1.85 on average in the L_∞ error, which is the best ever reported for this problem, to our knowledge. The numerical solution and L_∞ error on an 80 × 80 mesh are depicted in Fig. 13. The distribution of the error value is also depicted. It should be noticed that the maximal error occurs at the origin.

In Case 7(b), along the line $x+y=1$ in domain Ω^-, the second order derivatives does not exist, and the solution is only C^1 continuous. In this situation, we use a 13 point Gaussian quadrature form to improve the integration accuracy. As shown in Table 7, the present MIB Galerkin method attains about 1.72th order convergence in the L_∞ error for the solution and the first order convergence in the L_∞ error for the gradient of the solution. Such a result is essentially the same as the best reported by Mu et al. [77]. The numerical solution and L_∞ error on an 80 × 80 mesh are depicted in Fig. 14.

For Case 7(c), at the point (0, 0), the second order derivatives becomes unbounded. The solution belongs to the H^2 space. The problem is illustrated in Fig. 15. The best result in the literature is about 1.75th order convergence in the solution and the first order convergence in the gradient of the solution [77]. The present MIB Galerkin method offers essentially second order convergence in the solution and about 1.12th order convergence in the gradient of the solution, which is again, similar to the best performance in the literature [77].

Case 8. In this case, the 2D Poisson equations (1)–(4) is solved on domain $[-1,1]\times[-11]$. The interface Γ is given below.

$$\phi(x,y): \begin{cases} y-2x, & x+y>0 \\ y+x/2, & x+y\le 0. \end{cases} \tag{96}$$

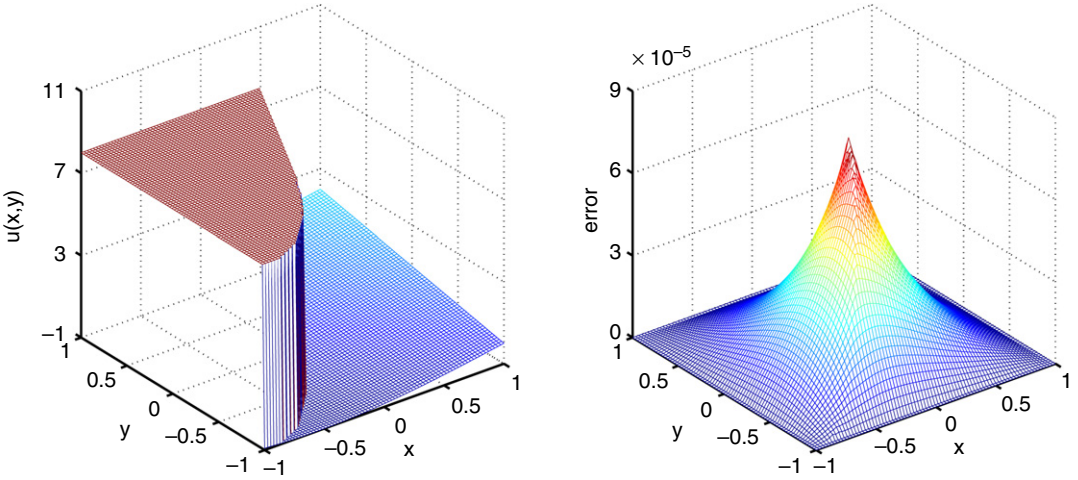

Fig. 13. The computed solution on an 80 × 80 mesh (left chart) and the L_∞ error (right chart) for Case 7(a).

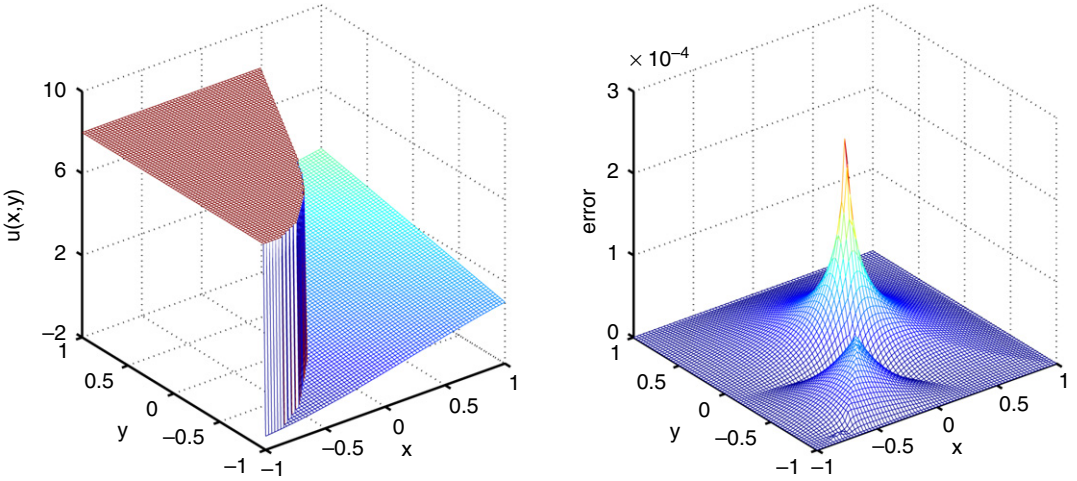

Fig. 14. The computed solution on an 80 × 80 mesh (left chart) and the L_∞ error (right chart) for Case 7(b).

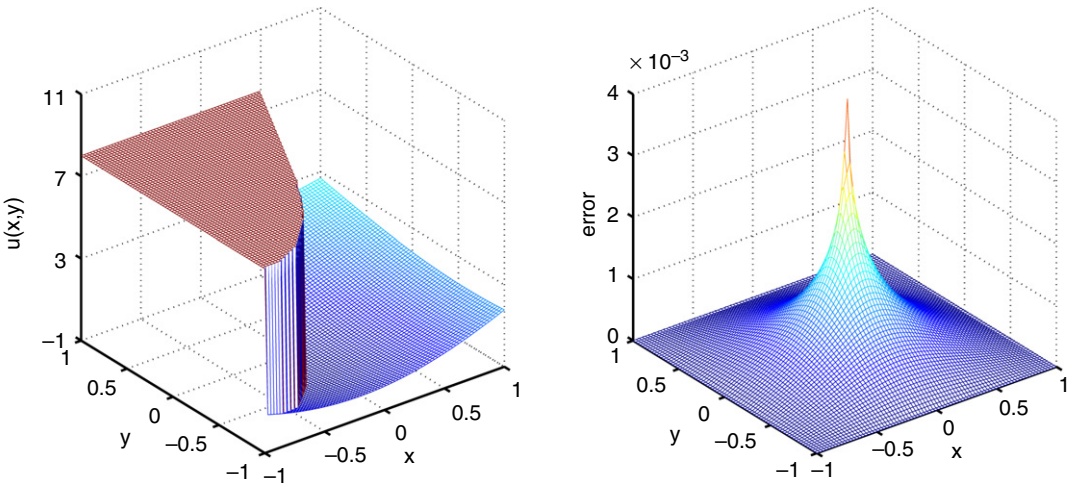

Fig. 15. The computed solution on an 80 × 80 mesh (left) and the L_∞ error (right) for Case 7(c).

Table 7
Numerical errors and convergence orders for a C^1 continuous interface (Case 7).

Case 7(a)	$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
	20×20	1.329e−3		5.836e−3	
	40×40	3.419e−4	1.96	1.662e−3	1.81
	80×80	8.270e−5	2.05	4.545e−4	1.87
	160×160	2.161e−5	1.94	1.255e−4	1.86
Case 7(b)	$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
	20×20	2.722e−3		4.788e−2	
	40×40	8.588e−4	1.66	2.413e−2	0.99
	80×80	2.601e−4	1.72	1.213e−2	0.99
	160×160	7.635e−5	1.77	6.091e−3	0.99
Case 7(c)	$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
	20×20	5.728e−2		1.864e−1	
	40×40	1.579e−2	1.86	8.709e−2	1.10
	80×80	4.143e−3	1.93	3.998e−2	1.12
	160×160	1.029e−3	2.01	1.781e−2	1.17

The discontinuous coefficients are given by

$$\beta^+(x, y) = (xy + 2)/5, \quad \text{and} \quad \beta^-(x, y) = (x^2 - y^2 + 3)/7. \tag{97}$$

The solution at two different subdomains is designed below.

- **Case** 8(a):

$$u^+(x, y) = 8, \tag{98}$$

$$u^-(x, y) = \begin{cases} x + y, & x + y > 0 \\ \sin(x + y), & x + y \le 0. \end{cases} \tag{99}$$

  This solution is of C^2 continuity.
- **Case** 8(b):

$$u^+(x, y) = 8, \tag{100}$$

$$u^-(x, y) = \begin{cases} x + y + 1, & x + y > 0 \\ \sin(x + y) + \cos(x + y), & x + y \le 0. \end{cases} \tag{101}$$

  This solution is C^1 continuous.
- **Case** 8(c):

$$u^+(x, y) = 8, \tag{102}$$

$$u^-(x, y) = (x^2 + y^2)^{5/6} + \sin(x + y). \tag{103}$$

  This solution is H^2 continuous.

The source term $g(x, y)$ and boundary value $g_b(x, y)$ can be determined accordingly for all cases.

Case 8 is designed to test our MIB Galerkin algorithm for a Lipschitz continuous interface geometry associated with low solution regularities. It can be seen from the expression of interface Γ that, at point $(0, 0)$, the interface is Lipschitz continuous but not C^1 continuous. The tip of the interface tends to reduce the accuracy of numerical methods. The present Galerkin algorithm can handle this kind of geometry singularities as shown in the earlier test. However, the present situation is quite different because the regularity of the solution is limited. The main results are summarized in Table 8. The distribution of the error value is also depicted. It should be noticed that the maximal error occurs at the origin.

In Case 8(a), the solution is of C^2 continuity. The MIB Galerkin appears to encounter no difficulty at all for this case. As shown in Fig. 16, the largest error does not occur at the origin. Therefore, the geometric singularity has been successfully taken care in the present approach. As shown in Table 8, the numerical scheme is more than second order convergence. Additionally, the convergence order of the gradient of the solution in the L_∞ error is near second order too, which is the best for this problem, to our knowledge.

In Case 8(b), the regularity of the solution is reduced to C^1. Such a reduction in regularity indeed affects the performance of the MIB Galerkin method as shown in Table 8. Although the convergence order of the solution in the L_∞ error is about 1.78 on average, the convergence of the gradient in the L_∞ error is only the first order, which is about the similar as the best performance reported in the literature [77]. Both the solution and the error are depicted in Fig. 17. The largest error occurs around the origin.

Table 8
Numerical errors and convergence orders for a Lipschitz continuous interface (Case 8).

Case 8(a)	$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
	20×20	4.874e−4		2.021e−3	
	40×40	8.462e−5	2.53	5.009e−4	2.01
	80×80	1.584e−5	2.42	1.279e−4	1.97
	160×160	3.683e−6	2.01	3.921e−5	1.71
Case 8(b)	$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
	20×20	3.149e−3		2.904e−2	
	40×40	9.518e−4	1.73	1.438e−2	1.01
	80×80	2.760e−4	1.79	7.165e−3	1.01
	160×160	7.823e−5	1.82	3.577e−3	1.00
Case 8(c)	$n_x \times n_y$	$L_\infty(u)$	Order	$L_\infty(\nabla u)$	Order
	20×20	4.989e−2		1.310e−1	
	40×40	1.270e−2	1.97	5.659e−2	1.21
	80×80	3.029e−3	2.07	2.394e−2	1.24
	160×160	6.468e−4	2.23	9.957e−3	1.27

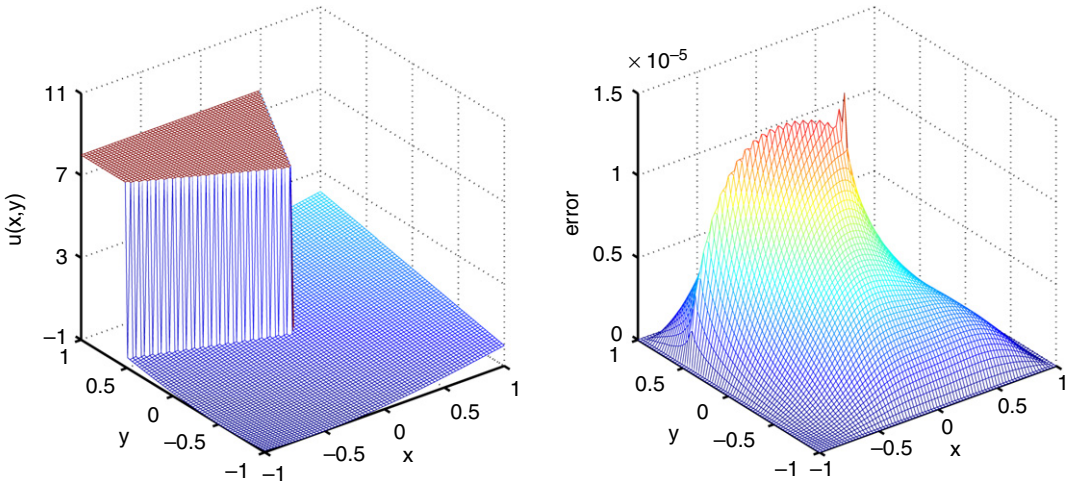

Fig. 16. The computed solution on an 80×80 mesh (left chart) and the error (right chart) for Case 8(a).

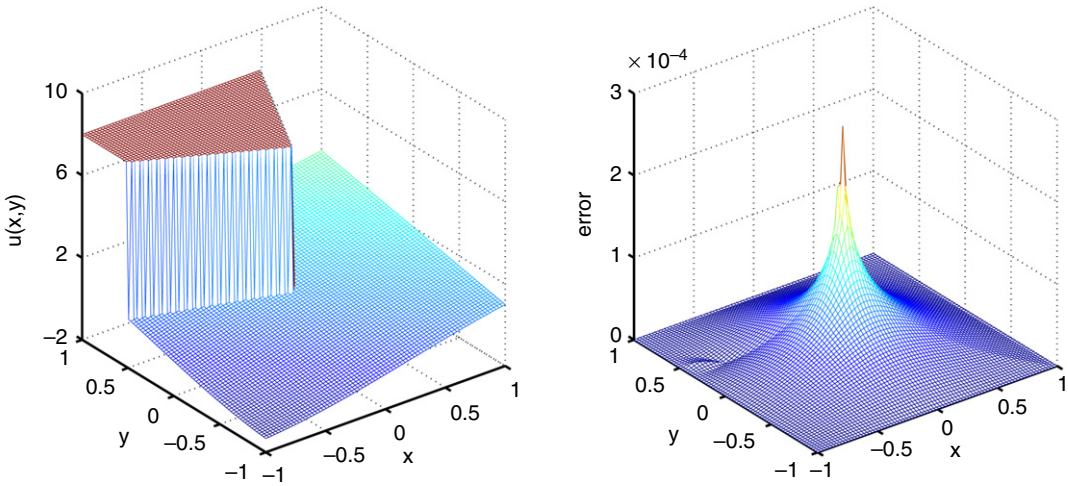

Fig. 17. The computed solution on an 80×80 mesh (left chart) and the error (right chart) for Case 8(b).

Finally, in Case 8(c), we consider the example of an H^2 solution. The solution blows up at the origin, which is also the point of the geometric singularity. Therefore, this situation is similar to the tip-geometry effects, namely, geometric singularity

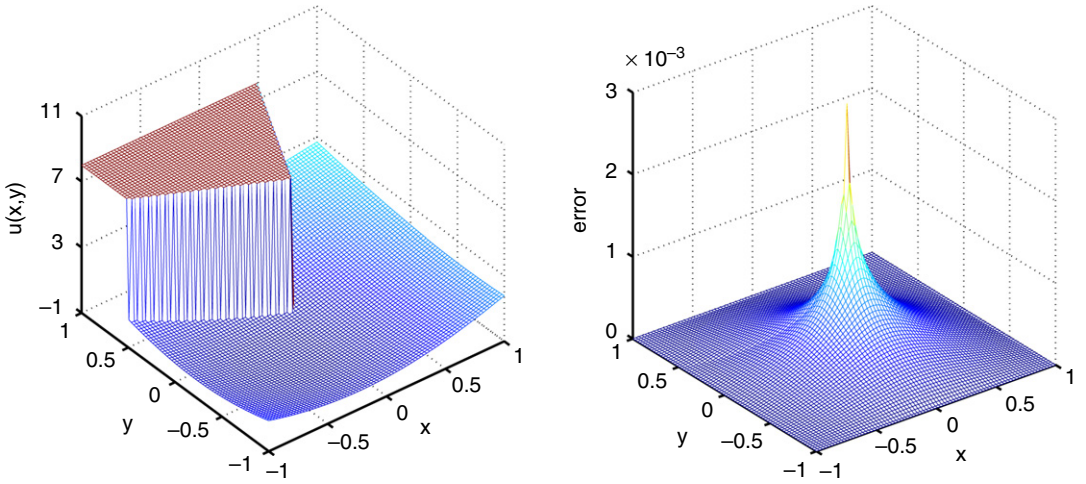

Fig. 18. The computed solution on an 80 × 80 mesh (left chart) and the error (right chart) for Case 8(c).

Table 9
Summary of the overall L_∞ orders of the MIB Galerkin method for solutions with low regularities (Cases 7 & 8).

Solution regularity	Γ is C^1 continuous		Γ is Lipschitz continuous	
	Solution	Gradient	Solution	Gradient
C^2	1.98	1.85	2.32	1.89
C^1	1.72	0.99	1.78	1.00
H^2	1.93	1.13	2.09	1.24

induced solution singularity. The MIB Galerkin method is demonstrated to perform well in the convergence order of the solution, i.e., the second order accuracy in the L_∞ error. However, its convergence order for the gradient of the solution is reduced to 1.2. Nonetheless, such a performance is still the best for this problem. Both the solution and the error are depicted in Fig. 18. Indeed, the largest error occurs at the origin.

Finally, the overall convergence orders in the L_∞ error for the MIB Galerkin method are summarized in Table 9 for Cases 7 and 8. Averaged convergence orders are listed in the table. Overall, the results of the present MIB Galerkin method are as good as the best in the literature [77]. In fact, for the cases of C^2 continuous solutions and C^1 or Lipschitz continuous interfaces, the present MIB Galerkin method delivers the best result, to our knowledge.

5. Concluding remarks

Material interfaces widely exist in nature as well as in man-made materials and devices. Mathematical modeling of material interfaces often leads to elliptic partial differential equations (PDEs) with discontinuous coefficients and singular source terms, commonly known as elliptic interface problems. The present paper introduces a new method, the Galerkin formulation of the matched interface and boundary (MIB) method, for this class of problems. The original MIB method is a systematic higher-order approach for elliptic interface problems. The 16th order scheme has been demonstrated with a simple geometry [56], and 6th order MIB scheme has been constructed for irregular interfaces in three-dimensional (3D) domains [55]. It provides the first known second order solution to elliptic interface problems with geometric singularities, i.e., nonsmooth interfaces with tips, cups, and sharp edges, in 2D [54] and 3D domains [55]. Geometric singularities are ubiquitous both in nature and in engineering devices and structures. When geometric singularities are associated with elliptic PDEs, they give rise to difficulties in mathematical analysis and in the design of numerical algorithms. What aggravates the difficulty in the computational methodology is that the solution becomes singular at the nonsmooth interface, which is a well-known natural phenomenon called tip-geometry effects in many fields. For this class of elliptic interface problems, our previous collocation based MIB method cannot be directly applied. The second accuracy was obtained in solving a modified PDE, which delivers a solution to the original elliptic interface problem [54,58]. It is desirable to develop numerical methods that directly solve the original elliptic PDE with low solution regularity induced by the geometric singularity.

It is well known that finite element methods (FEMs) using Galerkin formulations are able to perform better for problems with low regularities when the test functions are appropriately chosen. When the interface is C^1 or Lipschitz continuous and the solution is of C^1 continuity, the first order convergence in solution and 0.7 order of convergence in the gradient have been achieved by Hou et al. recently [71]. When the solution is H^2 continuous, their convergences in solution and gradient

are of 1.6th and 0.65th order, respectively. Very impressively, for the same problem, it is demonstrated that at least an order of 1.75 convergence in the solution and an order of 1 convergence in the gradient in the L_∞ error can be obtained by the Wang–Ye Galerkin FEM [76,77]. The Wang–Ye Galerkin FEM [76] is a new approach that employs discontinuous functions in the finite element procedure to gain flexibility in enforcing interface jump conditions. Such a strategy is similar in spirit to that used by the discontinuous Galerkin methods. However, the Wang–Ye Galerkin FEM employs distributions to discretize gradients in a much weak sense. It turns out that the Wang–Ye Galerkin FEM is particularly powerful in handling nonsmooth interfaces and its associated low solution regularity [77].

Motivated by the success of Galerkin formulations in elliptic interface problems, we have constructed a new Galerkin FEM based on our MIB method in this work. In the present Galerkin MIB method, we take the full advantage of the weak formulation through the Galerkin approach, and meanwhile, we make use of the flexibility and robustness of the MIB method for handling geometric singularities and complex interfaces. To avoid the time consuming mesh generation and mesh refinement, we utilize the Cartesian grid to construct triangular FEM meshes. As a result, the interface is allowed to cut through the FEM mesh. To maintain the continuity of the FEM basis functions across the interface, we define two sets of overlapping elements, called MIB elements in the solution domain. Consequently, the differentiation operators can be approximated via the classical FEM basis functions on the MIB elements as if there is no interface. The additional degrees of freedom, namely, fictitious values, on the overlapping domains are utilized to enforce the interface jump conditions in the same manner as that in the original MIB method. To this end, two sets of interpolation functions, one on each MIB element space, are defined to smoothly extend the function values across the interface. In the present Galerkin MIB as well as in the original MIB method, only a set of the lowest order interface jump conditions is allowed, which is important to the stability and accuracy of the MIB methods. After embedding the interface information in the Galerkin matrix elements, the total number of degrees of freedom in the MIB Galerkin method is the same as the number of nodes in the inner computational domain.

Extensive numerical experiments are designed to test the accuracy, examine the efficiency, and study the stability of the proposed MIB Galerkin finite element method. We consider interfaces with large curvatures, arbitrarily complex interfaces, and nonsmooth interfaces in the first set of test cases. The present method is shown to achieve the second convergence in the L_∞ error for both the solution and the gradient of the solution. In our second sets of test experiments, we further explore the capability of the present method for the problem of geometric singularity induced low solution regularities. The results from the present MIB Galerkin method are as good as the best ones in the literature achieved by the Wang–Ye Galerkin FEM [77] when the interface is C^1 or Lipschitz continuous and the solution of is C^1 or H^2 continuous. Moreover, when the interface is C^1 or Lipschitz continuous and the solution is of C^2 continuity, some of the best results are attained in the present work.

In all the numerical experiments, the final linear system is well-behaved and no singular matrix is encountered. We have also tested the persistence of convergence against the increase of grid resolutions over a number of interface geometries. It is found that the present method maintains the second order accuracy when the grid is refined more than 50 times, namely, from 20 node points to 1050 node points, in each direction.

The present 2D MIB Galerkin method can be extended in many aspects. For example, for real-world applications, it is important to develop 3D MIB Galerkin methods for elliptic interface problems [86]. Additionally, high order interface methods are crucial to many problems involving high frequency waves. Therefore, fourth order MIB Galerkin methods are to be developed too. These problems are under our consideration.

Acknowledgments

This work was supported in part by NSF grants IIS-1302285 and DMS-1160352, and NIH grant R01GM-090208.

References

[1] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (1) (2000) 35–60.
[2] M. Francois, W. Shyy, Computations of drop dynamics with the immersed boundary method, part 2: drop impact and heat transfer, Numer. Heat Transfer B 44 (2003).
[3] M. Francois, E. Uzgoren, J. Jackson, W. Shyy, Multigrid computations with the immersed boundary technique for multiphase flows, Int. J. Numer. Methods Heat Fluid Flow 14 (2004) 98–115.
[4] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, Appl. Mech. Rev. 56 (2003) 331–347.
[5] M. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 236–261.
[6] Long Lee, Randall J. LeVeque, An immersed interface method for incompressible navier–stokes equations, SIAM J. Sci. Comput. 25 (3) (2003) 832–856.
[7] Zhilin Li, Sharon R. Lubkin, Numerical analysis of interfacial two-dimensional stokes flow with discontinuous viscosity and variable surface tension, Internat. J. Numer. Methods Fluids 37 (5) (2001) 525–540.
[8] A.T. Layton, Using integral equations and the immersed interface method to solve immersed boundary problems with stiff forces, Comput. Fluids 38 (2009) 266–272.
[9] G.R. Hadley, High-accuracy finite-difference equations for dielectric waveguide analysis i: uniform regions and dielectric interfaces, J. Lightwave Technol. 20 (2002) 1210–1218.
[10] J.S. Hesthaven, High-order accurate methods in time-domain computational electromagnetics. a review, Adv. Imaging Electron Phys. 127 (2003) 59–123.
[11] R. Kafafy, T. Lin, Y. Lin, J. Wang, Three-dimensional immersed finite element methods for electric field simulation in composite materials, Internat. J. Numer. Methods Engrg. 64 (2005) 940–972.

[12] Juri D. Kandilarov, Immersed interface method for a reaction–diffusion equation with a moving own concentrated source, in: NMA '02: Revised Papers from the 5th International Conference on Numerical Methods and Applications, Springer-Verlag, London, UK, 2003, pp. 506–513.
[13] Shan Zhao, G.W. Wei, High-order FDTD methods via derivative matching for Maxwell's equations with material interfaces, J. Comput. Phys. 200 (1) (2004) 60–103.
[14] S. Zhao, High order matched interface and boundary methods for the Helmholtz equation in media with arbitrarily curved interfaces, J. Comput. Phys. 229 (2010) 3155–3170.
[15] T.P. Horikis, W.L. Kath, Modal analysis of circular bragg fibers with arbitrary index profiles, Opt. Lett. 31 (2006) 3417–3419.
[16] T.Y. Hou, Z.L. Li, S. Osher, H.K. Zhao, A hybrid method for moving interface problems with application to the heleshaw flow, J. Comput. Phys. 134 (2) (1997) 236–252.
[17] W.K. Liu, Y. Liu, D. Farrell, L. Zhang, X. Wang, Y. Fukui, N. Patankar, Y. Zhang, C. Bajaj, X. Chen, H. Hsu, Immersed finite element method and its applications to biological systems, Comput. Methods Appl. Mech. Engrg. 195 (2006) 1722–1749.
[18] Y.C. Zhou, M. Feig, G.W. Wei, Highly accurate biomolecular electrostatics in continuum dielectric environments, J. Comput. Chem. 29 (2008) 87–97.
[19] S.N. Yu, W.H. Geng, G.W. Wei, Treatment of geometric singularities in implicit solvent models, J. Chem. Phys. 126 (2007) 244108.
[20] Weihua Geng, Sining Yu, G.W. Wei, Treatment of charge singularities in implicit solvent models, J. Chem. Phys. 127 (2007) 114106.
[21] Duan Chen, Zhan Chen, Changjun Chen, W.H. Geng, G.W. Wei, MIBPB: a software package for electrostatic analysis, J. Comput. Chem. 32 (2011) 657–670.
[22] Hend Ben Ameur, Martin Burger, Benjamin Hackl, Level set methods for geometric inverse problems in linear elasticity, Inverse Problems 20 (3) (2004) 673–696.
[23] Petter Andreas Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, J. Comput. Phys. 197 (1) (2004) 364–386.
[24] George Biros, Lexing Ying, Denis Zorin, A fast solver for the Stokes equations with distributed forces in complex geometries, J. Comput. Phys. 193 (1) (2004) 317–348.
[25] Miguel A. Dumett, James P. Keener, An immersed interface method for solving anisotropic elliptic boundary value problems in three dimensions, SIAM J. Sci. Comput. 25 (1) (2003) 348–367.
[26] Aaron L. Fogelson, James P. Keener, Immersed interface methods for neumann and related problems in two and three dimensions, SIAM J. Sci. Comput. 22 (5) (2001) 1630–1654.
[27] Frdric Gibou, Ronald Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. 202 (2) (2005) 577–601.
[28] Songming Hou, Xu-Dong Liu, A numerical method for solving variable coefficient elliptic equation with interfaces, J. Comput. Phys. 202 (2) (2005) 411–445.
[29] H. Huang, Z.L. Li, Convergence analysis of the immersed interface method, IMA J. Numer. Anal. 19 (4) (1999) 583–608.
[30] John K. Hunter, Z.L. Li, Hongkai Zhao, Reactive autophobic spreading of drops, J. Comput. Phys. 183 (2) (2002) 335–366.
[31] Shi Jin, Xuelei Wang, Robust numerical simulation of porosity evolution in chemical vapor infiltration: II. two-dimensional anisotropic fronts, J. Comput. Phys. 179 (2) (2002) 557–577.
[32] Hans Johansen, Phillip Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, J. Comput. Phys. 147 (1) (1998) 60–85.
[33] Zhilin Li, Wei-Cheng Wang, I-Liang Chern, Ming-Chih Lai, New formulations for interface problems in polar coordinates, SIAM J. Sci. Comput. 25 (1) (2003) 224–245.
[34] Mark N. Linnick, Hermann F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (1) (2005) 157–192.
[35] Bruno Lombard, Joël Piraux, How to incorporate the spring-mass conditions in finite-difference schemes, SIAM J. Sci. Comput. 24 (4) (2003) 1379–1407.
[36] Martin Schulz, Gerd Steinebach, Two-dimensional modelling of the river Rhine, J. Comput. Appl. Math. 145 (1) (2002) 11–20.
[37] J.A. Sethian, Evolution, implementation, and application of level set and fast marching methods for advancing fronts, J. Comput. Phys. 169 (2) (2001) 503–555.
[38] Anna-Karin Tornberg, Bjrn Engquist, Numerical approximations of singular source terms in differential equations, J. Comput. Phys. 200 (2) (2004) 462–488.
[39] J.John Vande Voorde, Jan Vierendeels, Erik Dick, Flow simulations in rotary volumetric pumps and compressors with the fictitious domain method, J. Comput. Appl. Math. 168 (1-2) (2004) 491–499.
[40] G. Morgenthal, J.H. Walther, An immersed interface method for the vortex-in-cell algorithm, Comput. Struct. 85 (11–14) (2007) 712–726.
[41] Andreas Wiegmann, Kenneth P. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (3) (2000) 827–862.
[42] W. Cai, S.Z. Deng, An upwinding embedded boundary method for Maxwell's equations in media with material interfaces: 2d case, J. Comput. Phys. 190 (2003) 159–183.
[43] J.T. Beale, A.T. Layton, On the accuracy of finite difference methods for elliptic problems with interfaces, Commun. Appl. Math. Comput. Sci. 1 (2006) 91–119.
[44] B.E. Griffith, C.S. Peskin, On the order of accuracy of the immersed boundary method: higher order convergence rates for sufficiently smooth problems, J. Comput. Phys. 208 (2005) 75–105.
[45] M.C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719.
[46] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252.
[47] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numer. Anal. 21 (1984) 285–299.
[48] A. McKenney, L. Greengard, A. Mayo, A fast Poisson solver for complex geometries, J. Comput. Phys. 118 (1995) 348–355.
[49] R.J. LeVeque, Z.L. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.
[50] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[51] X.D. Liu, R.P. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (2000) 151–178.
[52] T. Chen, J. Strain, Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems, J. Comput. Phys. 16 (2008) 7503–7542.
[53] J.L. Hellrung Jr., L.M. Wang, E. Sifakis, J.M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions, J. Comput. Phys. 231 (2012) 2015–2048.
[54] S.N. Yu, Y.C. Zhou, G.W. Wei, Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces, J. Comput. Phys. 224 (2) (2007) 729–756.
[55] S.N. Yu, G.W. Wei, Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities, J. Comput. Phys. 227 (2007) 602–632.
[56] Y.C. Zhou, Shan Zhao, Michael Feig, G.W. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys. 213 (1) (2006) 1–30.
[57] Y.C. Zhou, G.W. Wei, On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method, J. Comput. Phys. 219 (1) (2006) 228–246.
[58] K.L. Xia, Meng Zhan, Guo-Wei Wei, The matched interface and boundary (MIB) method for multi-domain elliptic interface problems, J. Comput. Phys. 230 (2011) 8231–8258.

[59] S. Zhao, Full-vectorial matched interface and boundary (MIB) method for the modal analysis of dielectric waveguides, IEEE/OSA J. Lightw. Technol. 26 (2008) 2251–2259.
[60] Y.C. Zhou, J.G. Liu, D.L. Harry, A matched interface and boundary method for solving multi-flow navier-stokes equations with applications to geodynamics, J. Comput. Phys. 231 (2012) 223–242.
[61] I. Babuška, The finite element method for elliptic equations with discontinuous coefficients, Computing 5 (1970) 207–213.
[62] R.E. Ewing, Z.L. Li, T. Lin, Y.P. Lin, The immersed finite volume element methods for the elliptic interface problems, Math. Comput. Simul. 50 (1999) 63–76.
[63] I. Ramiere, Convergence analysis of the q_1-finite element method for elliptic problems with non-boundary-fitted meshes, Internat. J. Numer. Methods Engrg. 75 (2008) 1007–1052.
[64] Xiaoming He, Tao Lin, Yanping Lin, Interior penalty bilinear IFE discontinuous Galerkin methods for elliptic equations with discontinuous coefficient, J. Syst. Sci. Complexity 23 (2010) 467–483.
[65] Zhiqiang Cai, Xiu Ye, Shun Zhang, Discontinuous Galerkin finite element methods for interface problems: a priori and a posteriori error estimations, SIAM J. Numer. Anal. 49 (2011) 1761–1787.
[66] M. Dryjaa, J. Galvisb, M. Sarkisb, BDDC methods for discontinuous Galerkin discretization of elliptic problems, J. Complexity 23 (2007) 715–739.
[67] J. Bramble, J. King, A finite element method for interface problems in domains with smooth boundaries and interfaces, Adv. Comput. Math. 6 (1996) 109–138.
[68] Erik Burman, Peter Hansbo, Interior-penalty-stabilized Lagrange multiplier methods for the finite-element solution of elliptic interface problems, IAM J. Numer. Anal. 30 (2010) 870–885.
[69] X. Sheldon Wang, L.T. Zhang, Liu Wing Kam, On computational issues of immersed finite element methods, J. Comput. Phys. 228 (2009) 2535–2551.
[70] A. Hansbo, P. Hansbo, An unfitted finite element method, Comput. Methods Appl. Mech. Engrg. 191 (2002) 5537–5552.
[71] S.M. Hou, W. Wang, L.Q. Wang, Numerical method for solving matrix coefficient elliptic equation with sharp-edged interfaces, J. Comput. Phys. 229 (2010) 7162–7179.
[72] I. Harari, J. Dolbow, Analysis of an efficient finite element method for embedded interface problems, Comput. Math. 46 (2010) 205–211.
[73] N. Sukumar, D.L. Chopp, N. Moes, T. Belytschko, Modeling holes and inclusions by level sets in the extended finite-element method, Comput. Methods Appl. Mech. Engrg. 190 (2001) 6180–6200.
[74] R. Hiptmair, J. Li, J. Zou, Convergence analysis of finite element methods for H(div;Omega)-elliptic interface problems, J. Numer. Math. 18 (2010) 187–218.
[75] D.A. Wang, R. Li, N.N. Yan, An edge-based anisotropic mesh refinement algorithm and its application to interface problems, Commun. Comput. Phys. 8 (2010) 511–540.
[76] J. Wang, X. Ye, A weak Galerkin finite element method for second-order elliptic problems, J. Comput. Appl. Math. 241 (2013) 103–115.
[77] L. Mu, J. Wang, G.W. Wei, X. Ye, S. Zhao, Weak Galerkin method for second order elliptic interface problems, J. Comput. Phys. 250 (2013) 106–125.
[78] K.L. Xia, Meng Zhan, D.C. Wan, G.W. Wei, Adaptively deformed mesh based matched interface and boundary (MIB) method for elliptic interface problems, J. Comput. Phys. 231 (2012) 1440–1461.
[79] W. Geng, G.W. Wei, Multiscale molecular dynamics using the matched interface and boundary method, J. Comput. Phys. 230 (2) (2011) 435–457.
[80] Qiong Zheng, Duan Chen, G.W. Wei, Second-order Poisson-Nernst-Planck solver for ion transport, J. Comput. Phys. 230 (2011) 5239–5262.
[81] D. Chen, G.W. Wei, Modeling and simulation of electronic structure, material interface and random doping in nano-electronic devices, J. Comput. Phys. 229 (2010) 4431–4460.
[82] G.W. Wei, Discrete singular convolution for the solution of the Fokker–Planck equations, J. Chem. Phys. 110 (1999) 8930–8942.
[83] G.W. Wei, Y.Q. Jia, Synchronization-based image edge detection, Europhys. Lett. 59 (6) (2002) 814–819.
[84] R. Glowinski, Tsorng-Whay Pan, Jacques Periaux, A fictitious domain method for Dirichlet problem and applications, Comput. Methods Appl. Mech. Engrg. 111 (1994) 283–303.
[85] Z.L. Li, T. Lin, X.H. Wu, New Cartesian grid methods for interface problems using the finite element formulation, Numer. Math. 96 (2003) 61–98.
[86] K.L. Xia, G.W. Wei, A galerkin formulation of the MIB method for three dimensional elliptic interface problems, Comput. Math. Appl. (2013), revised.