Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 213 (2006) 1–30
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources

Y.C. Zhou [a], Shan Zhao [a], Michael Feig [b,c], G.W. Wei [a,d,*]

[a] Department of Mathematics, Michigan State University, East Lansing, MI 48824-1027, USA
[b] Department of Chemistry, Michigan State University, East Lansing, MI 48824, USA
[c] Department of Biochemistry and Molecular Biology, Michigan State University, East Lansing, MI 48824, USA
[d] Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA

Received 4 April 2005; received in revised form 27 July 2005; accepted 28 July 2005
Available online 19 September 2005

Abstract

This paper introduces a novel high order interface scheme, the matched interface and boundary (MIB) method, for solving elliptic equations with discontinuous coefficients and singular sources on Cartesian grids. By appropriate use of auxiliary line and/or fictitious points, physical jump conditions are enforced at the interface. Unlike other existing interface schemes, the proposed method disassociates the enforcement of physical jump conditions from the discretization of the differential equation under study. To construct higher order interface schemes, the proposed MIB method bypasses the major challenge of implementing high order jump conditions by repeatedly enforcing the lowest order jump conditions. The proposed MIB method is of arbitrarily high order, in principle. In treating straight, regular interfaces we construct MIB schemes up to 16th-order. For more general elliptic problems with curved, irregular interfaces and boundary, up to 6th-order MIB schemes have been demonstrated. By employing the standard high-order finite difference schemes to discretize the Laplacian, the present MIB method automatically reduces to the standard central difference scheme when the interface is absent. The immersed interface method (IIM) is regenerated for a comparison study of the proposed method. The robustness of the MIB method is verified against the large magnitude of the jump discontinuity across the interface. The nature of high efficiency and low memory requirement of the MIB method is extensively validated via solving various elliptic immersed interface problems in two- and three-dimensions.

Keywords: Immersed boundary method; Immersed interface method; Ghost fluid method; Matched interface and boundary; Elliptic equations; High order method

* Corresponding author. Tel.: +1 517 3534689; fax: +1 517 4321562.
E-mail address: wei@math.msu.edu (G.W. Wei).

0021-9991/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2005.07.022

1. Introduction

In this paper, we propose a class of numerical methods for solving elliptic interface problems

$$\nabla \cdot (\beta \nabla u) - \kappa u(\mathbf{x}) = q(\mathbf{x}), \quad \mathbf{x} \in \Omega = \Omega^+ \cup \Omega^- \tag{1}$$

with a boundary condition on $\partial\Omega$ on regular Cartesian grids. For simplicity, Ω is assumed to be a regular domain, such as rectangle type in two-dimensions (2D) or cube like in three-dimensions (3D). Across the interface Γ in domain Ω, the coefficient function $\beta(\mathbf{x})$ of the elliptic equation is discontinuous, while the source term $q(\mathbf{x})$ may be even singular. Depending on the properties of the source term $q(\mathbf{x})$, we usually have jump conditions across the interface Γ:

$$[u] = u^+(\mathbf{X}(s)) - u^-(\mathbf{X}(s)) = \phi(s), \quad [\beta u_n] = \beta^+ u_n^+(\mathbf{X}(s)) - \beta^- u_n^-(\mathbf{X}(s)) = \psi(s), \tag{2}$$

where $\mathbf{X}(s)$ is a point on the interface Γ, s the arc-length parametrization of the interface Γ, and n the unit normal direction. The superscript, $-$ or $+$, denotes the limiting value of a function from one side or the other of the interface. Such an elliptic interface problem is often encountered in fluid dynamics, molecular biology [13], and material science, and its solution is usually non-smooth or even discontinuous across the interface. The standard numerical methods designed for smooth solutions usually perform poorly for these interface problems. Consequently, the development of accurate and efficient numerical approaches based on a Cartesian grid for the elliptic interface problems has attracted much attentions in the last decade [1,2,5,17,18,25,27–29,35,48,51].

It is noted that there also exist numerous studies in the literature about a quite relevant problem, an elliptic irregular domain problem [22,37,38]. One way to solve this problem is to embed the irregular domain in a larger regular computational domain, and reformulate the original boundary conditions as interface jump conditions. A simple Cartesian grid can then be adopted, so that various fast algebraic solvers developed in the literature can be utilized. This essentially converts an elliptic irregular domain problem into an elliptic interface problem. Due to this close relationship, the methods originally designed for one type of problems may be extended to solve another one. However, in immersed boundary problems, no solution is sought outside the domain boundary, whereas, in immersed interface problems, interface jump conditions couple solutions in the both sides of the interface. We will primarily focus on solving the elliptic immersed interface problems in the present study.

To solve an elliptic problem with an irregular domain or interface, a body-fitting grid can be employed [3,7,8]. Nevertheless, for certain geometrically complex domains, the construction of a good body-fitting mesh remains a nontrivial and time-consuming task, even though considerable progress has been made. Furthermore, a considerable increase of the computational difficulties will be encountered for moving interface problems, where a moving mesh method is required to regenerate or deform the grid during the simulation. Therefore, numerous modified finite difference (FD) methods that are based only on a simple Cartesian grid have been chosen to solve the elliptic interface-related problems in the literature [1,2,5,11,12,17,18,20,22,25,27–29,32,35,37–40,45,48,51]. One obvious advantage of Cartesian grid methods is that there is no computational cost for grid generation. A Cartesian grid also allows the use of simple data structures and standard FD stencil over a majority of the domain. Moreover, many contemporary software packages, such as fast Poisson solvers, multigrid, level set method, etc., are mainly developed for Cartesian grid, and thus could be taken advantage of. On the other hand, in order to properly maintain the accuracy at the interface, some extra numerical work needs to be done near the interface in a Cartesian FD method.

In the 1970–1980s, Peskin [32,39,40] developed the immersed boundary method (IBM) to model blood flow in the heart. In the IBM, the complicated time-varying geometric boundary is regarded as being immersed in the fluid, and the Navier–Stokes equation is solved on a simple Cartesian grid. The presence of the embedded boundary is modeled via a singular source on the interfaces. The IBM has had success in

modeling flow in complicated time-dependent geometry [11,45] due to its flexibility, efficiency, and robustness. Recently, Tornberg and Engquist [44] have proposed and analyzed a class of regularization methods for treating the singular source terms. An extensive review of the IBM for turbulent flow simulations can be found in [20]. However, the IBM is typically first order accurate in higher dimensions.

Recently, some high order immersed boundary schemes have been proposed in the literature [15,33]. Gibou and Fedkiw [15] constructed a fourth-order boundary scheme by using cubic extrapolation at the boundary. This scheme was applied to the Laplace and heat equations. The advantage of this method is its simplicity. Linnick [33] and Fasel developed a fourth-order immersed boundary scheme for flow past cylinders by using the compact finite difference scheme. These high order immersed boundary schemes are restricted to certain class of boundaries and have a potential to be generalized for treating elliptic equations with interface jump conditions, i.e., with discontinuous coefficients.

A major advance in this field was due to LeVeque and Li [25]. These authors proposed the immersed interface method (IIM) for solving elliptic equations with discontinuous coefficients and singular sources. In the IIM, local corrections of FD schemes on irregular grid points where the discretized Laplacian operator involves nodes from both sides of the interface, are pursued throughout the domain. This is achieved by incorporating interface jump conditions into local Taylor expansions of the operator on irregular points, from which FD schemes with genuinely first-order accurate truncation error can be derived. The resulting FD scheme is of second-order accuracy and preserves the jumps at the interface. For 2D or 3D problems, a local coordinate is typically required to offer a better representation of the jump conditions since they are usually given in the direction normal to the interface. The IIM is robust and efficient, and has been successfully applied to a variety of interface-related problems, such as moving interface problems [17], elliptic irregular domain problems [10], 3D interface problems [9], etc. The reader is referred to a recent review [28] and references therein for more details about these applications.

Various extensions and further improvements of the IIM have been considered in the literature [4–6,10,16–19,21,23,24,30,31,36,42,43,46,47,51]. The IIM formulations in polar coordinates [31] and using a finite element method [27] were presented. Hou and Liu considered elliptic problems with non-smooth interfaces [16]. Dumett and Keener proposed an IIM for solving anisotropic elliptic boundary value problems [10]. The problem of convergence and efficiency of the IIM has attracted much research interest. The original IIM [25] typically leads to a matrix of non-symmetric coefficients though the original problems are self-adjoint and strictly elliptic. This reduces the number of standard fast solvers that can be utilized with IIM and convergence may not be rigorously guaranteed [18]. To address this problem, a maximum principle preserving IIM was proposed by Li and Ito [29] to attain a diagonally dominant albeit still non-symmetric coefficient matrix. Specially designed multigrid solvers can then be employed to speed up the convergence of the maximum principle preserving IIM [1,2]. For interface problems with piecewise constant coefficient, a fast IIM was constructed [26] through introducing an unknown jump condition for $[u_n]$. Together with the elliptic equation, this unknown will also be solved numerically. The success of the fast IIM lies in the fact that the IIM models the jump conditions of $[u]$ and $[u_n]$ by using the standard FD scheme with a correction term in 2D. As a result, various standard fast Poisson solvers can be applied. Motivated by the fast IIM [26], an explicit jump IIM was developed by Wiegmann and Bube [51]. Recently, a decomposed IIM for elliptic equations with variable coefficients was proposed by Berthelsen [5]. The linear systems in two IIMs [51,5] are all symmetric and diagonally dominant, allowing the use of conventional fast Poisson solvers.

Another popular sharp interface scheme using the Cartesian grid is the ghost fluid method (GFM) originally developed for treating contact discontinuities in the inviscid Euler equations by Osher and his coworkers [12]. The GFM is typically first-order accurate for interface problems, including the elliptic one [35] and could be of second-order accuracy for elliptic irregular domain problems. In the flavor of the level set method which gives an implicit representation of the interface, the interface jump conditions are captured implicitly by extending values across the interface into a ghost fluid. On irregular grid points, when the FD discretized

Laplacian refers to a node from the other side of the interface, a ghost fluid value instead of the real one will be supplied. Such a jump condition capturing procedure is directly incorporated into the numerical discretization in a way that the symmetry of the FD coefficient matrix is maintained, allowing the use of standard fast solvers. In higher dimensions, the jump in the normal derivative is correctly captured through a projection to Cartesian coordinate directions in the GFM, while the jump in the tangential derivatives is neglected [35]. Then the GFM can be applied dimension by dimension. The GFM is very simple and robust, and its practical extensions to complex interface problems such as 3D moving interfaces or the multiphase Navier–Stokes equations are promising. Recently, an interesting jump condition capturing FD scheme was constructed by Wang [48] by using a body-fitting curvilinear coordinate system.

The idea of fictitious points or ghost domain is also used in the literature for solving elliptic irregular domain problems by Mayo [37,38] in 1980s. The ghost cell is introduced outside the (inner) domain as a fictitious domain. Similar ideas have been explored in [22,45], but the fictitious values are smoothly extended or extrapolated according to the given jump conditions. In the discrete singular convolution (DSC) algorithm [49], fictitious points are used to satisfy boundary conditions [54].

The objective of the present paper is to introduce a matched interface and boundary (MIB) method for solving elliptic equations with discontinuous coefficients and singular sources. The MIB method is formulated based on our previous interface scheme, the hierarchical derivative matching [52,53], originally designed for electromagnetic wave propagation and scattering in inhomogeneous media. It has achieved over 12th-order of accuracy for solving Maxwell's equations with discontinuities in material permittivity. This high order interface approach has also been applied as a boundary scheme for the treatment some general or complex boundary conditions, such as free edge supports in structural analysis [54]. Unlike the IIM and GFM, the MIB disassociates the discretization of partial differential equations (PDEs) from enforcing the jump conditions. Moreover, it repeatedly uses only the lowest order jump conditions in an iterative manner, so as to avoid the challenge of handling cross derivatives arisen from high order jump conditions. The MIB modeling is systematically carried out and can be made to arbitrarily high order, in principle. For straight interfaces, up to 16th-order MIB schemes are constructed. For practical curve interfaces, constructing schemes that exceed second-order is usually difficult on the Cartesian grid. In the present work, we demonstrate second-, fourth- and sixth-order MIB schemes for elliptic equations with irregular immersed interfaces and/or immersed boundary.

The rest of this paper is organized as follows. In Section 2, we first formulate the MIB scheme for elliptic equations with regular interfaces. The treatment of general, irregular interfaces is developed based the same set of ideas. The source of the local truncation error and its order is analyzed in details. Sections 3 and 4 are devoted to extensive numerical test and validation of the proposed MIB method for regular and irregular elliptic interface problems, respectively. To have a detailed comparison, the IIM of LeVeque and Li [25] is regenerated in the present study. A discussion on the similarity and difference between MIB method and other interface methods, i.e., the IIM and GFM, is given in Section 5. A conclusion remark ends this paper.

2. Theory and algorithm

In this section, we propose the matched interface and boundary (MIB) method for solving elliptic equations based on the MIB scheme originally constructed for time-domain Maxwell's equations. Regular and irregular interfaces are treated in the following two subsections, respectively.

2.1. Regular interfaces

For regular interfaces, i.e., straight lines in two-dimensional (2D) problems or planes in three-dimensional (3D) problems, which are aligned to a Cartesian coordinate direction, the proposed MIB approach

in principle can be made to be arbitrarily high order. To illustrate the idea, we start with a general one-dimensional (1D) model problem

$$(\beta u_x)_x - \kappa u = q. \tag{3}$$

The domain is the unit interval $[0,1]$ with the Dirichlet, Neumann or Robin boundary conditions. We assume that the coefficient β is a piecewise constant, i.e., $\beta = \beta^-$ if $x < \alpha$ and $\beta = \beta^+$ if $x > \alpha$, and κ and q may also be discontinuous, or even singular, at $x = \alpha$. Mathematically, these discontinuities can be accounted via two general jump conditions

$$[u] = u^+ - u^- = \phi, \quad [\beta u_x] = \beta^+ u_x^+ - \beta^- u_x^- = \psi. \tag{4}$$

A universal rule for all matched interface approaches is that to approximate a derivative on one side of an interface, one never directly refers to function values from the other side of the interface. Jump conditions must be embedded into such a derivative approximation, in order to avoid accuracy reduction.

With jump conditions (4), the model problem (3) can be stated alternatively to be

$$(\beta u_x)_x - \kappa u = q, \quad x \in (0,\alpha) \cup (\alpha,1). \tag{5}$$

It is noted that then in each subdomain excluding the interface α, the coefficients and source terms of the PDE are continuous. By treating the jump conditions (4) as boundary conditions, this alternative statement allows a domain decomposition treatment of the original problem, which motivates the development of the MIB approach.

In all matched interface approaches, one first distinguishes the irregular grid points from regular ones. The irregular grid points are those points on which a standard finite difference (FD) approximation to the derivative would refer to grid points not all from the same side of interface. Without the loss of generality, we assume the length of the differential stencil under consideration to be $2M+1$. Then there would be M irregular grid points on each side of the interface. In the MIB method, if the derivative approximation at these irregular grid points refers to points from the other side of the interface, the fictitious values in place of real ones will be supplied. Thus, there should be M fictitious points (FPs) on each side of the interface, or a total of $2M$ FPs. To determine fictitious values on these $2M$ FPs by using jump conditions (4), functions values on $2L$ original grid points on both sides of the interface will be involved in the MIB approach, see Fig. 1. We denote the function values of u at the original points and FPs as g_i and f_i, respectively. For simplicity, a uniform grid is assumed in the present discussion, albeit non-uniform grids can be used as well.

In the MIB method, the fictitious values are determined in an iterative manner, which involves repeated use of the lowest order jump conditions. In the first step, we seek to determine two fictitious values f_1 and f_2 as shown in Fig. 1, by discretizing two jump conditions (4)

$$\sum_{i=1}^{L} w_{0,i}^- g_i + w_{0,L+1}^- f_2 = w_{0,1}^+ f_1 + \sum_{i=2}^{L+1} w_{0,i}^+ g_{L+i-1} - [u], \tag{6}$$

$$\beta^- \left(\sum_{i=1}^{L} w_{1,i}^- g_i + w_{1,L+1}^- f_2 \right) = \beta^+ \left(w_{1,1}^+ f_1 + \sum_{i=2}^{L+1} w_{1,i}^+ g_{L+i-1} \right) - [\beta u_x], \tag{7}$$

where $w_{j,i}^-$ and $w_{j,i}^+$ for $i = 1,\ldots,L+1$ and $j = 0,1$ are one-sided FD weights for left and right subdomains. Here the subscript j represents (zeroth-order) interpolation or first-order derivative approximation, and i is for grid index. These weights can be conveniently generated through a call into a small subroutine presented in [14]. In general, in order to make sure that the whole scheme to be of pth order accuracy, the FD approximation in interface matching should be at least pth order accurate. Thus, in Eqs. (6) and (7), we employ one-sided FD weights with L grid points from the other side of the interface, as illustrated in Fig. 1, to achieve high order. Two unknowns f_1 and f_2 can be simply solved from the resulting two algebraic equations (6) and (7).

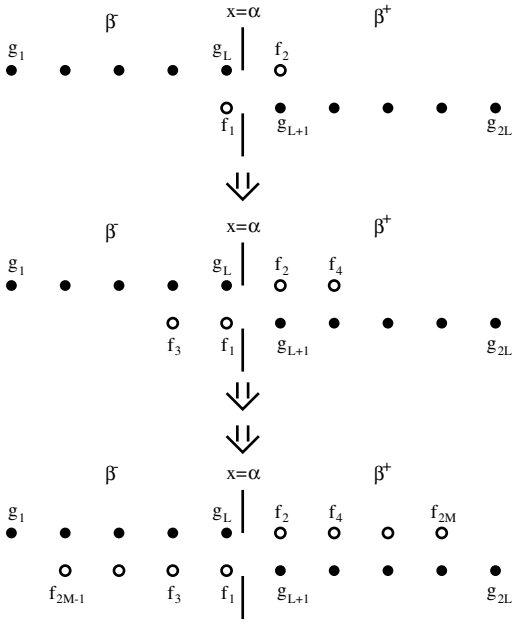

Fig. 1. Illustration of fictitious and original grid points used in the MIB approaches. The FPs are shown as open circles, while the original grid points are shown as filled circles.

It is clear that the MIB method does not explicitly employ the Taylor expansion to enforce jump conditions. Nevertheless, we note that in fact the standard FD approximation can be theoretically formulated based on the Taylor expansion. The advantage of the present matched interface scheme lies in that all computations including FD weight generations are carried out automatically via a Fortran code, without involving any manual derivation. Thus, it can be extended easily to as high order as one wishes by appropriately choosing M, L and corresponding FD schemes.

To achieve higher order, we can determine two more fictitious values by enforcing the previous two jump conditions (4) again, see the second stage of Fig. 1. It is noted that f_1 and f_2 have already been determined from the first step and are treated as knowns. The grid partition is chosen to still have L nodes from one side, but 2 FPs from the other side of the interface (Of course, one might fix the total length of grid partition as $L+1$ as well). Thus, this partition is independent of the previous one. By similarly discretizing jump conditions again, two new unknowns f_3 and f_4 can be determined via

$$\sum_{i=1}^{L} \tilde{w}^{-}_{0,i} g_i + \tilde{w}^{-}_{0,L+1} f_2 + \tilde{w}^{-}_{0,L+2} f_4 = \tilde{w}^{+}_{0,1} f_3 + \tilde{w}^{+}_{0,2} f_1 + \sum_{i=3}^{L+2} \tilde{w}^{+}_{0,i} g_{L+i-2} - [u], \tag{8}$$

$$\beta^{-}\left(\sum_{i=1}^{L} \tilde{w}^{-}_{1,i} g_i + \tilde{w}^{-}_{1,L+1} f_2 + \tilde{w}^{-}_{1,L+2} f_4\right) = \beta^{+}\left(\tilde{w}^{+}_{1,1} f_3 + \tilde{w}^{+}_{1,2} f_1 + \sum_{i=3}^{L+2} \tilde{w}^{+}_{1,i} g_{L+i-2}\right) - [\beta u_x]. \tag{9}$$

Note that the FD weights $\tilde{w}^{-}_{j,i}$ and $\tilde{w}^{+}_{j,i}$ here are different from those in Eqs. (6) and (7), and computationally, one needs to regenerate them.

This procedure can be easily extended to higher orders. All the $2M$ FPs can be generated iteratively, from inner to outer. See Fig. 1. Moreover, since the fictitious values are systematically solved, the present interface modeling can be made up to arbitrarily high order in principle. In general, the proposed MIB scheme can be applied to any higher order FD scheme to restore its full accuracy. In the present study,

the standard higher order central FD kernel will be employed as the basic differentiation kernel throughout the domain, albeit upwind schemes can also be used, if desirable. Thus, the present MIB method automatically reduces to the standard FD scheme when the material interfaces are absent.

The MIB method outlined above could be applied to an initial value problem [52], in which field values g_i [i.e., $u(x)$] are available. However, for solving boundary value or eigenvalue problems, an implicit version of the MIB has to be considered. To this end, we introduce a fundamental representation [52]

$$f_i = \mathbf{C}^i \cdot \mathbf{G}, \quad \text{for } i = 1, 2, \ldots, 2M, \tag{10}$$

where vector $\mathbf{G} = (g_1, \ldots, g_{2L}, [u], [\beta u_x])$ and the elements of vector $\mathbf{C}^i$ are the expansion coefficients of f_i with respect to $\mathbf{G}$. With this representation, instead of solving f_i, one needs to determine $\mathbf{C}^i$. The representation coefficients $\mathbf{C}^i$ will be determined from essentially the same procedure presented above for f_i. The only difference is that now one jump condition is discretized and spanned into $2L + 2$ algebraic equations, since a fictitious value f_i is represented via $2L + 2$ coefficients which are the $2L + 2$ elements of $\mathbf{C}^i$.

To better illustrate the MIB approach, we next present detailed MIB formulation for a fourth-order FD scheme with the half-length of the differential stencil being $M = 2$. Thus, there are a total of $2M = 4$ FPs. To ensure at least fourth-order accuracy in the MIB modeling, we choose $L = 4$. Consequently, $\mathbf{G} = (g_1, \ldots, g_8, [u], \beta u_x])$. By denoting $\mathbf{I}^i$ as a unit vector with its ith element being 1 and other $2L + 1$ elements being 0, we have

$$g_i = \mathbf{I}^i \cdot \mathbf{G}, \quad \text{for } i = 1, 2, \ldots, 2L, \quad [u] = \mathbf{I}^{2L+1} \cdot \mathbf{G}, \quad [\beta u_x] = \mathbf{I}^{2L+2} \cdot \mathbf{G}. \tag{11}$$

By using the representation (10) and (11), the discretized jump conditions (6) and (7) in the first MIB step are given as

$$\sum_{i=1}^{4} w_{0,i}^{-} \mathbf{I}^i + w_{0,5}^{-} \mathbf{C}^2 = w_{0,1}^{+} \mathbf{C}^1 + \sum_{i=2}^{5} w_{0,i}^{+} \mathbf{I}^{3+i} - \mathbf{I}^9 + \mathrm{O}(h^5), \tag{12}$$

$$\beta^{-} \left(\sum_{i=1}^{4} w_{1,i}^{-} \mathbf{I}^i + w_{1,5}^{-} \mathbf{C}^2 \right) = \beta^{+} \left(w_{1,1}^{+} \mathbf{C}^1 + \sum_{i=2}^{5} w_{1,i}^{+} \mathbf{I}^{3+i} \right) - \mathbf{I}^{10} + \mathrm{O}(h^4), \tag{13}$$

in which the common factor $\mathbf{G}$ has been cancelled. Two fictitious values can be solved as

$$\mathbf{C}^1 = \frac{\beta^{-} w_{1,5}^{-} \mathbf{K}_1 - w_{0,5}^{-} \mathbf{K}_2}{\beta^{+} w_{1,1}^{+} w_{0,5}^{-} - \beta^{-} w_{1,5}^{-} w_{0,1}^{+}}, \tag{14}$$

$$\mathbf{C}^2 = \frac{\beta^{+} w_{1,1}^{+} \mathbf{K}_1 - w_{0,1}^{+} \mathbf{K}_2}{\beta^{+} w_{1,1}^{+} w_{0,5}^{-} - \beta^{-} w_{1,5}^{-} w_{0,1}^{+}}, \tag{15}$$

where

$$\mathbf{K}_1 = \sum_{i=2}^{5} w_{0,i}^{+} \mathbf{I}^{3+i} - \sum_{i=1}^{4} w_{0,i}^{-} \mathbf{I}^i - \mathbf{I}^9,$$

$$\mathbf{K}_2 = \beta^{+} \sum_{i=2}^{5} w_{1,i}^{+} \mathbf{I}^{3+i} - \beta^{-} \sum_{i=1}^{4} w_{1,i}^{-} \mathbf{I}^i - \mathbf{I}^{10}.$$

Similarly, in the second (final) step, Eqs. (8) and (9) become

$$\sum_{i=1}^{4} \tilde{w}_{0,i}^{-} \mathbf{I}^i + \tilde{w}_{0,5}^{-} \mathbf{C}^2 + \tilde{w}_{0,6}^{-} \mathbf{C}^4 = \tilde{w}_{0,1}^{+} \mathbf{C}^3 + \tilde{w}_{0,2}^{+} \mathbf{C}^1 + \sum_{i=3}^{6} \tilde{w}_{0,i}^{+} \mathbf{I}^{2+i} - \mathbf{I}^9 + \mathrm{O}(h^5), \tag{16}$$

$$\beta^{-} \left(\sum_{i=1}^{4} \tilde{w}_{1,i}^{-} \mathbf{I}^i + \tilde{w}_{1,5}^{-} \mathbf{C}^2 + \tilde{w}_{1,6}^{-} \mathbf{C}^4 \right) = \beta^{+} \left(\tilde{w}_{1,1}^{+} \mathbf{C}^3 + \tilde{w}_{1,2}^{+} \mathbf{C}^1 + \sum_{i=3}^{6} \tilde{w}_{1,i}^{+} \mathbf{I}^{2+i} \right) - \mathbf{I}^{10} + \mathrm{O}(h^4). \tag{17}$$

The fictitious values f_3 and f_4 can be solved as

$$\mathbf{C}^3 = \frac{\beta^- \tilde{w}^-_{1,6}\tilde{\mathbf{K}}_1 - \tilde{w}^-_{0,6}\tilde{\mathbf{K}}_2}{\beta^+ \tilde{w}^+_{1,1}\tilde{w}^-_{0,6} - \beta^- \tilde{w}^-_{1,6}\tilde{w}^+_{0,1}}, \tag{18}$$

$$\mathbf{C}^4 = \frac{\beta^+ \tilde{w}^+_{1,1}\tilde{\mathbf{K}}_1 - \tilde{w}^+_{0,1}\tilde{\mathbf{K}}_2}{\beta^+ \tilde{w}^+_{1,1}\tilde{w}^-_{0,6} - \beta^- \tilde{w}^-_{1,6}\tilde{w}^+_{0,1}}, \tag{19}$$

where

$$\tilde{\mathbf{K}}_1 = \sum_{i=3}^{6} \tilde{w}^+_{0,i}\mathbf{I}^{2+i} - \sum_{i=1}^{4} \tilde{w}^-_{0,i}\mathbf{I}^i + \tilde{w}^+_{0,2}\mathbf{C}^1 - \tilde{w}^-_{0,5}\mathbf{C}^2 - \mathbf{I}^9,$$

$$\tilde{\mathbf{K}}_2 = \beta^+ \sum_{i=3}^{6} \tilde{w}^+_{1,i}\mathbf{I}^{2+i} - \beta^- \sum_{i=1}^{4} \tilde{w}^-_{1,i}\mathbf{I}^i + \beta^+ \tilde{w}^+_{1,2}\mathbf{C}^1 - \beta^- \tilde{w}^-_{1,5}\mathbf{C}^2 - \mathbf{I}^{10}.$$

For elliptic interface problems with regular interfaces, the extension of the MIB approach to higher dimensions is straightforward. For example, we consider the following 2D model problem

$$(\beta u_x)_x + (\beta u_y)_y - \beta\kappa u = -q, \quad (x, y) \in [0, 1] \times [0, 1] \tag{20}$$

with proper boundary conditions given on boundaries of the domain. Suppose along the y-direction the media are homogeneous, while along the x-direction, β could take different values

$$\beta = \begin{cases} \beta_1, & 0 \leqslant x < \alpha,\ 0 \leqslant y \leqslant 1, \\ \beta_2, & \alpha \leqslant x \leqslant 1,\ 0 \leqslant y \leqslant 1. \end{cases} \tag{21}$$

Since now the normal direction n is on the Cartesian coordinate directions, jump conditions read

$$[u] = \phi(\mathbf{x}), \quad [\beta u_n] = [\beta u_x] = \psi(\mathbf{x}), \tag{22}$$

which are essentially Eq. (4). Thus, the 1D MIB approaches can be directly applied to the present 2D model problem to restore the FD approximations along the x-direction, while the standard FD scheme will be applied along the y-direction. Obviously, for a 3D problem with cubic domain and discontinuities of the diffusion coefficient taking place only along the Cartesian directions, the MIB approach can also be directly employed to recover the full accuracy of the FD method.

2.2. Irregular interfaces

In this subsection, we formulate the MIB scheme for solving elliptic equations with irregular (curved) interfaces, while our previous MIB method was constructed only for regular interfaces. It is much more intricate to treat irregular interfaces because the local topology of irregular points may vary from point to point. Therefore, the procedure outlined for regular interfaces cannot be directly applied. To better illustrate the new MIB scheme, we mainly focus our description on 2D elliptic problems given in Eq. (3). The formalism of the proposed MIB approach for 3D problems and other elliptic equations is very similar. A second-order MIB method and its generalization to higher orders are described in the rest of this subsection.

2.2.1. A second-order scheme for irregular interfaces

Before proceeding to the construction of the MIB scheme for $(\beta u_x)_x$ or $(\beta u_y)_y$, we identify irregular grid points since only these points necessitate special care when the standard central difference scheme is applied to the whole domain. We define the irregular point as the one at which all the discretization points in a standard central FD scheme are not on the same side of an interface. For example, in a second-order

2D scheme, an irregular point has at least one of its 4 nearest neighbor grid points laying on the other side of the interface. Note that the number of irregular points increases when a higher order FD scheme is employed.

Assume as before that there are two given conditions associated with the interface, i.e.,

$$[u] = u^+ - u^- = \phi(x,y), \tag{23}$$

$$[\beta u_n] = \beta^+ u_n^+ - \beta^- u_n^- = \psi(x,y) \tag{24}$$

and also assume that both $\phi(x,y)$ and $\psi(x,y)$ are C^1 continuous along the interface Γ. When considering the interface which is not always aligned with the x- or y-mesh lines, as what shown by Fig. 3, one more interface condition can be attained by differentiating Eq. (24) along the tangential direction of the interface, i.e. $[u_\tau] = \phi_\tau(x,y) \triangleq \rho(x,y)$. Hence, for a point (x_o,y_o) on the interface, we have three jump conditions,

$$[u] = u^+ - u^- = \phi(x_o,y_o), \tag{25}$$

$$[u_\tau] = u_\tau^+ - u_\tau^- = \rho(x_o,y_o), \tag{26}$$

$$[\beta u_n] = \beta^+ u_n^+ - \beta^- u_n^- = \psi(x_o,y_o), \tag{27}$$

where normal vector of the interface $\vec{n} = (\cos\theta, \sin\theta)$ and u_τ is the derivative in the tangential direction $\vec{\tau} = (-\sin\theta, \cos\theta)$, while $0 \leqslant \theta < 2\pi$ is the angle between positive x-direction and the vector $\vec{n}$. Considering these relations, the above three local interface conditions can be reformulated as

$$[u] = u^+ - u^- = \phi(x_o,y_o), \tag{28}$$

$$[u_\tau] = (-u_x^+ \sin\theta + u_y^+ \cos\theta) - (-u_x^- \sin\theta + u_y^- \cos\theta) = \rho(x_o,y_o), \tag{29}$$

$$[\beta u_n] = \beta^+(u_x^+ \cos\theta + u_y^+ \sin\theta) - \beta^-(u_x^- \cos\theta + u_y^- \sin\theta) = \psi(x_o,y_o). \tag{30}$$

In the MIB approach, the implementation of jump conditions is disassociated with the discretization of the elliptic equation. Also $(\beta u_x)_x$ and $(\beta u_y)_y$ will be treated separately. Therefore, we only need to illustrate how to locally recover the second-order accuracy of the standard 3-point FD scheme for $(\beta u_x)_x$. The modeling for $(\beta u_y)_y$ can be achieved similarly. Considering an interface point $o = (x_o,y_o)$ which is located at the intersection of interface Γ and the x-mesh line, see Fig. 3, one fictitious point on each side of (x_o,y_o) is required in the present MIB modeling. To estimate these two fictitious values we will discretize u^+, u^-, u_x^+ and u_x^- involved in the jump conditions by using a 1D grid partition as in the regular interface cases. However, difficulty arise as how to deal with two derivatives u_y^+ and u_y^- in the jump conditions (29) and (30) at the interface point (x_o,y_o). We overcome this difficulty via essentially two steps: first, by using jump conditions (29) and (30), we eliminate one y derivative that is more difficult to discretized at the interface; second, we carry out the discretization for the remaining y-derivative by using a one-sided FD scheme on the auxiliary points whose values are obtained by interpolation.

This is always possible in practice. If u_y^+ is easier to be evaluated, one will cancel u_y^- from (29) and (30) to attain

$$[u] = u^+ - u^-, \quad \text{and} \quad [\beta u_n] - \beta^- \tan\theta [u_\tau] = C_x^+ u_x^+ - C_x^- u_x^- + C_y^+ u_y^+, \tag{31}$$

where $C_x^+ = \beta^+ \cos\theta + \beta^- \tan\theta \sin\theta$, $C_x^- = \beta^- \cos\theta + \beta^- \tan\theta \sin\theta$ and $C_y^+ = \beta^+ \sin\theta - \beta^- \sin\theta$. If otherwise u_y^- is easier to be calculated, one shall use the following jump conditions derived from Eqs. (28)–(30) by cancelling u_y^+:

$$[u] = u^+ - u^-, \quad \text{and} \quad [\beta u_n] - \beta^+ \tan\theta [u_\tau] = C_x^+ u_x^+ - C_x^- u_x^- - C_y^- u_y^-, \tag{32}$$

where $C_x^+ = \beta^+ \cos\theta + \beta^+ \tan\theta \sin\theta$, $C_x^- = \beta^- \cos\theta + \beta^+ \tan\theta \sin\theta$ and $C_y^- = \beta^- \sin\theta - \beta^+ \sin\theta$.

Cancellation of u_x^+ or u_x^- from Eqs. (29) and (30) is required when modeling $(\beta u_y)_y$, which leads to another two combinations of jump conditions. One combination is

$$[u] = u^+ - u^-, \quad \text{and} \quad [\beta u_n] + \beta^- \cot\theta [u_\tau] = C_x^+ u_x^+ + C_y^+ u_y^+ - C_y^- u_y^-, \tag{33}$$

where $C_x^+ = (\beta^+ - \beta^-)\cos\theta$, $C_y^+ = \beta^- \cos\theta \cot\theta + \beta^+ \sin\theta$ and $C_y^- = \beta^-(\cos\theta\cot\theta + \sin\theta)$; and the other is

$$[u] = u^+ - u^- \quad \text{and} \quad [\beta u_n] + \beta^+ \cot\theta [u_\tau] = C_x^- u_x^- + C_y^+ u_y^+ - C_y^- u_y^-. \tag{34}$$

It is easy to check that here $C_x^- = (\beta^+ - \beta^-)\cos\theta$, $C_y^- = \beta^+ \cos\theta\cot\theta + \beta^- \sin\theta$ and $C_y^+ = \beta^+(\cos\theta\cot\theta + \sin\theta)$. Either Eq. (33) or Eq. (34) can be chosen to formulate the discretization scheme for $(\beta u_y)_y$, depending on if u_x^+ or u_x^- is easier to be evaluated.

In the case that either $\tan\theta$ or $\cot\theta$ is undefined in choosing jump conditions, i.e., whenever the outer normal direction is aligned to x- or y-direction, the interface is locally perpendicular to the mesh line so that it can be treated as a straight interface locally. The procedure introduced in the preceding subsection for the straight interface can be directly employed to handle these irregular points.

From the above discussion, it can be seen that for irregular interfaces each irregular point should have its own local MIB representation because of different local topology, while for regular interface cases, in contrast, only one global MIB representation is needed throughout the whole domain. Some typical topology near irregular points is depicted in Fig. 2.

We first consider the detailed discretization of jump conditions (31) at the interface point (x_o, y_o) for the topology given in Fig. 3, where the interface Γ passes through two irregular points (i,j) and $(i+1,j)$ at (x_o, y_o). Four nodes along the jth mesh line, i.e., $(i-1,j)$, (i,j), $(i+1,j)$ and $(i+2,j)$, are required to approximate x-derivatives in jump conditions (31). In order to evaluate u_y^+ at $(x_o, y_o) = (x_o, y_j)$, we add an auxiliary y-mesh (dashed line) passing through (x_o, y_o). Three auxiliary grid points on this auxiliary line will be employed to approximate u_y^+. Two of these three points, $(o, j+1)$ and $(o, j+2)$ are on the positive side of the interface, while the third auxiliary point (o,j) is right on the interface.

Refer again to Fig. 3, where point $(i-1,j)$ and (i,j) are on the same side of the interface while $(i+1,j)$, $(i+2,j)$, $(o,j+1)$ and $(o,j+2)$ are on the other side. We deploy two fictitious points at both points (i,j) and $(i+1,j)$, the corresponding fictitious unknowns are denoted as $f_{i,j}$ and $f_{i+1,j}$, respectively. Two conditions in Eq. (31) are then approximated as

$$w_{0,i-1}^- u_{i-1,j} + w_{0,i}^- u_{i,j} + w_{0,i+1}^- f_{i+1,j} + [u] = w_{0,i}^+ f_{i,j} + w_{0,i+1}^+ u_{i+1,j} + w_{0,i+2}^+ u_{i+2,j} + \mathrm{O}(h^3), \tag{35}$$

$$\begin{aligned} &C_x^-(w_{1,i-1}^- u_{i-1,j} + w_{1,i}^- u_{i,j} + w_{1,i+1}^- f_{i+1,j}) + [\beta u_n] - \beta^- \tan\theta [u_\tau] \\ &\quad = C_x^+(w_{1,i}^+ f_{i,j} + w_{1,i+1}^+ u_{i+1,j} + w_{1,i+2}^+ u_{i+2,j}) + C_y^+(p_{1,j}^+ u_{o,j}^+ + p_{1,j+1}^+ u_{o,j+1} + p_{1,j+2}^+ u_{o,j+2}) + \mathrm{O}(h^2), \end{aligned} \tag{36}$$

where superscripts, $-$ and $+$, signify that the FD approximation is on the $-$ and $+$ side of the interface, respectively. Here w and p are FD weights for approximation along the x- and y-directions, respectively.

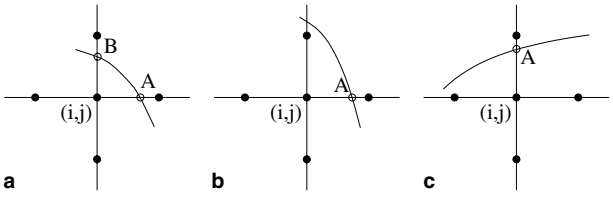

Fig. 2. Three typical situations for an interface crossing the mesh lines. In the left chart, the interface passes the x-mesh line at point A and y-mesh line at point B. The interface conditions are approximated at point A and point B to generate the difference scheme for $(\beta u_x)_x$ and $(\beta u_y)_y$, respectively. In the middle chart, the interface conditions are approximated at point A to obtain the difference scheme only for $(\beta u_x)_x$ and the regular central difference is used for $(\beta u_y)_y$. In the right chart, the difference scheme for $(\beta u_y)_y$ is built from the approximation of the interface conditions at point B while term $(\beta u_x)_x$ is treated with the regular central difference.

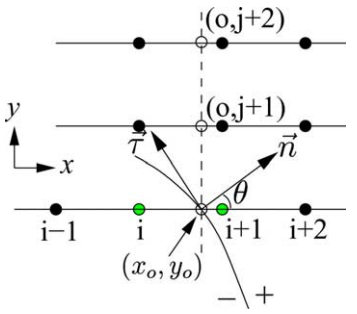

Fig. 3. Irregular point (i,j) and the interface. The interface crosses the x- mesh line at (x_o, y_o). The vertical dash line is the auxiliary line on which three auxiliary points (in empty circle) are defined: $(o,j+2)$,$(o,j+1)$ and (o,j) right at (x_o, y_o). The jumps $[u]$, $[\beta u_n]$ and $[u_\tau]$ are evaluated at (x_o, y_o).

Their first subscript (0 or 1) represents either interpolation or first-order derivative, while their second subscript is for node index. For example, $w^-_{0,i-1}$, $w^-_{0,i}$ and $w^-_{0,i+1}$ are the interpolation weights of $u^-(x_o, y_o)$ at grid points $(i-1,j)$, (i,j) and $(i+1,j)$.

As aforementioned, u^+_y shall be approximated on a one-sided finite difference stencil comprising of three auxiliary points, i.e., $u^+_{o,j}$, $u_{o,j+1}$ and $u_{o,j+2}$, which need further numerical treatment. To relate $u^+_{o,j}$ with function values at real grid points we adopt the relation

$$u^+_{o,j} = u^-_{o,j} + [u] = w^-_{0,i-1}u_{i-1,j} + w^-_{0,i}u_{i,j} + w^-_{0,i+1}f_{i+1,j} + [u] + \mathrm{O}(h^3). \tag{37}$$

The other two auxiliary values, $u_{o,j+1}$ and $u_{o,j+2}$ are interpolated with three grid points around them to ensure an accuracy of $\mathrm{O}(h^3)$. These interpolation points are chosen a priori, and should be on the same side of the interface. Here, we choose interpolation points to be $(i,j+1)$, $(i+1,j+1)$, $(i+2,j+1)$ and $(i,j+2)$, $(i+1,j+2)$, $(i+2,j+2)$ for $u_{o,j+1}$ and $u_{o,j+2}$, respectively. As a result, $u_{o,j+1}$ and $u_{o,j+2}$ become known values in jump condition (36).

There are only two unknowns in Eqs. (35) and (36), i.e., fictitious values $f_{i,j}$ and $f_{i+1,j}$. To solve for the representation of these fictitious values in terms of the real function values and the known jumps, we introduce two expansions

$$f_{i,j} = \mathbf{C}^i \cdot \mathbf{U}, \tag{38}$$

$$f_{i+1,j} = \mathbf{C}^{i+1} \cdot \mathbf{U}, \tag{39}$$

where $\mathbf{C}^i = (\mathbf{C}^i_1, \mathbf{C}^i_2, \ldots, \mathbf{C}^i_9)$ and $\mathbf{C}^{i+1} = (\mathbf{C}^{i+1}_1, \mathbf{C}^{i+1}_2, \ldots, \mathbf{C}^{i+1}_9)$ are the expansion coefficients of two fictitious unknowns with respect to 6 real unknowns and 3 jumps which are also given by the vector $\mathbf{U} = (u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, u_{o,j+1}, u_{o,j+2}, [u], [\beta u_n], [u_\tau])$. Insert these expansions of $f_{i,j}$ and $f_{i+1,j}$ to Eqs. (35) and (36), we end up with a small linear system for vectors $\mathbf{C}^i$ and $\mathbf{C}^{i+1}$:

$$w^-_{0,i+1}\mathbf{C}^{i+1} \cdot \mathbf{U} - w^+_{0,i}\mathbf{C}^i \cdot \mathbf{U} = \mathbf{K}_1 \cdot \mathbf{U}, \tag{40}$$

$$(C^-_x w^-_{1,i+1} - C^+_y p^+_{1,j} w^-_{0,i+1})\mathbf{C}^{i+1} \cdot \mathbf{U} - C^+_x w^+_{1,i}\mathbf{C}^i \cdot \mathbf{U} = \mathbf{K}_2 \cdot \mathbf{U}, \tag{41}$$

or further

$$w^-_{0,i+1}\mathbf{C}^{i+1} - w^+_{0,i}\mathbf{C}^i = \mathbf{K}_1, \tag{42}$$

$$(C^-_x w^-_{1,i+1} - C^+_y p^+_{1,j} w^-_{0,i+1})\mathbf{C}^{i+1} - C^+_x w^+_{1,i}\mathbf{C}^i = \mathbf{K}_2, \tag{43}$$

where vector

$$\mathbf{K}_1 = (-w^-_{0,i-1}, -w^-_{0,i}, w^+_{0,i+1}, w^+_{0,i+2}, 0, 0, -1, 0, 0),$$
$$\mathbf{K}_2 = (-C^-_x w^-_{1,i-1} + C^+_y p^+_{1,j} w^-_{0,i-1}, -C^-_x w^-_{1,i} + C^+_y p^+_{1,j} w^-_{0,i}, C^+_x w^+_{1,i+1}, C^+_x w^+_{1,i+2}, C^+_y p^+_{1,j+1},$$
$$C^+_y p^+_{1,j+2}, C^+_y p^+_{1,j}, -1, \beta^- \tan\theta).$$

The solution of this 2×2 linear system is readily available with Cramer's formula:

$$\mathbf{C}^i = \frac{w^-_{0,i+1}\mathbf{K}_2 - (C^-_x w^-_{1,i+1} C^+_y p^+_{1,j} w^-_{0,i+1})\mathbf{K}_1}{-C^+_x w^+_{1,i} w^-_{0,i+1} + w^+_{0,i}(C^-_x w^-_{1,i+1} C^+_y p^+_{1,j} w^-_{0,i+1})}, \tag{44}$$

$$\mathbf{C}^{i+1} = \frac{-C^+_x w^+_{1,i}\mathbf{K}_1 + w^+_{0,i}\mathbf{K}_2}{-C^+_x w^+_{1,i} w^-_{0,i+1} + w^+_{0,i}(C^-_x w^-_{1,i+1} C^+_y p^+_{1,j} w^-_{0,i+1})}. \tag{45}$$

Before finishing the seeking of the fictitious values, we need to distribute the expansion weights of $f_{i,j}$ and $f_{i+1,j}$ at auxiliary points $u_{o,j+1}$ and $u_{o,j+2}$ to their respective interpolation points defined above. Suppose $u_{o,j+1} = \mathbf{I}^{j+1}\cdot(u_{i,j+1}, u_{i+1,j+1}, u_{i+2,j+1}) + \mathrm{O}(h^3)$ and $u_{o,j+2} = \mathbf{I}^{j+2}\cdot(u_{i,j+2}, u_{i+1,j+2}, u_{i+2,j+2}) + \mathrm{O}(h^3)$, where $\mathbf{I}^{j+1} = (\mathbf{I}^{j+1}_1, \mathbf{I}^{j+1}_2, \mathbf{I}^{j+1}_3)$ and $\mathbf{I}^{j+2} = (\mathbf{I}^{j+2}_1, \mathbf{I}^{j+2}_2, \mathbf{I}^{j+2}_3)$ are the interpolation coefficients in the vector form. The expansion weights of $f_{i,j}$ and $f_{i+1,j}$ could then be distributed onto these interpolation points and the final expressions of $f_{i,j}$ and $f_{i+1,j}$ are:

$$\begin{aligned} f_{i,j} &= \mathbf{C}^i_1 u_{i-1,j} + \mathbf{C}^i_2 u_{i,j} + \mathbf{C}^i_3 u_{i+1,j} + \mathbf{C}^i_4 u_{i+2,j} + \mathbf{C}^i_5(\mathbf{I}^{j+1}_1 u_{i,j+1} + \mathbf{I}^{j+1}_2 u_{i+1,j+1} + \mathbf{I}^{j+1}_3 u_{i+2,j+1}) \\ &\quad + \mathbf{C}^i_6(\mathbf{I}^{j+2}_1 u_{i,j+2} + \mathbf{I}^{j+2}_2 u_{i+1,j+2} + \mathbf{I}^{j+2}_3 u_{i+2,j+2}) + \mathbf{C}^i_7[u] + \mathbf{C}^i_8[\beta u_n] + \mathbf{C}^i_9[u_\tau], \end{aligned} \tag{46}$$

$$\begin{aligned} f_{i+1,j} &= \mathbf{C}^{i+1}_1 u_{i-1,j} + \mathbf{C}^{i+1}_2 u_{i,j} + \mathbf{C}^{i+1}_3 u_{i+1,j} + \mathbf{C}^{i+1}_4 u_{i+2,j} + \mathbf{C}^{i+1}_5(\mathbf{I}^{j+1}_1 u_{i,j+1} + \mathbf{I}^{j+1}_2 u_{i+1,j+1} + \mathbf{I}^{j+1}_3 u_{i+2,j+1}) \\ &\quad + \mathbf{C}^{i+1}_6(\mathbf{I}^{j+2}_1 u_{i,j+2} + \mathbf{I}^{j+2}_2 u_{i+1,j+2} + \mathbf{I}^{j+2}_3 u_{i+2,j+2}) + \mathbf{C}^{i+1}_7[u] + \mathbf{C}^{i+1}_8[\beta u_n] + \mathbf{C}^{i+1}_9[u_\tau]. \end{aligned} \tag{47}$$

With these two expansions of $f_{i,j}$ and $f_{i+1,j}$, one could discretize $(\beta u_x)_x$ at irregular points (i,j) and $(i+1,j)$ as at the regular point:

$$(\beta u_x)_x = \frac{\beta^-_{i-\frac{1}{2},j}}{(\Delta x)^2} u_{i-1,j} - \frac{\beta^-_{i-\frac{1}{2},j} + \beta^-_{i+\frac{1}{2},j}}{(\Delta x)^2} u_{i,j} + \frac{\beta^-_{i+\frac{1}{2},j}}{(\Delta x)^2} f_{i+1,j} \quad \text{at } (i,j), \tag{48}$$

$$(\beta u_x)_x = \frac{\beta^+_{i+\frac{1}{2},j}}{(\Delta x)^2} f_{i,j} - \frac{\beta^+_{i+\frac{1}{2},j} + \beta^+_{i+\frac{3}{2},j}}{(\Delta x)^2} u_{i+1,j} + \frac{\beta^+_{i+\frac{3}{2},j}}{(\Delta x)^2} u_{i+2,j} \quad \text{at } (i+1,j), \tag{49}$$

by substituting the above expansions of $f_{i,j}$ and $f_{i+1,j}$. The known terms involving $[u]$, $[u_\tau]$ and $[\beta u_x]$ should be collected and contribute to the right-hand vector B in the final linear system $A\mathbf{x} = B$ from which the elliptic equation is eventually solved.

The difference scheme for $(\beta u_y)_y$ can be generated following the procedure similar to that for $(\beta u_x)_x$. Let (i,j) and $(i,j+1)$ be a pair of irregular points and between them the interface Γ intersects with the ith mesh line at (x_o, y_o), see Fig. 4. We need two fictitious points, $f_{i,j}$ at point (i,j) and $f_{i,j+1}$ at $(i,j+1)$, to facilitate the discretization of $(\beta u_y)_y$ at (i,j) and $(i,j+1)$. Jump conditions, Eq. (34), is chosen and is approximated at the intersect point (x_o, y_o). Considering the fact that we do not have real grid points to directly approximate u^-_x, a horizontal auxiliary mesh line passing (x_o, y_o) is introduced and three auxiliary points, $(i-2,o)$,$(i-1,o)$ and (i,o), are deployed on which a one-sided finite difference scheme for u^-_x is formulated. Jump conditions (34) are then replaced by the following approximate equations:

$$p^-_{0,j-1} u_{i,j-1} + p^-_{0,j} u_{i,j} + p^-_{0,j+1} f_{i,j+1} + [u] = p^+_{0,j} f_{i,j} + p^+_{0,j+1} u_{i,j+1} + p^+_{0,j+2} u_{i,j+2} + \mathrm{O}(h^3), \tag{50}$$

$$\begin{aligned} &C^-_y(p^-_{1,j-1} u_{i,j-1} + p^-_{1,j} u_{i,j} + p^-_{1,j+1} f_{i,j+1}) + [\beta u_n] + \beta^+ \cot\theta [u_\tau] \\ &\quad = C^+_y(p^+_{1,j} f_{i,j} + p^+_{1,j+1} u_{i,j+1} + p^+_{1,j+2} u_{i,j+2}) + C^-_x(w^-_{1,i-2} u_{i-2,o} + w^-_{1,i-1} u_{i-1,o} + w^-_{1,i} u^-_{i,o}) + \mathrm{O}(h^2). \end{aligned} \tag{51}$$

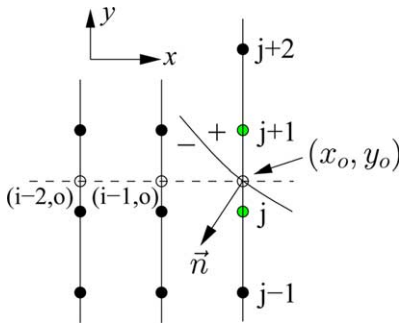

Fig. 4. Irregular point (i,j) and the interface. The interface crosses the y-mesh line at (x_o,y_o). The dash horizontal line is the auxiliary line on which three auxiliary points (in empty circle) are defined: $(i-2,o)$, $(i-1,o)$ and (i,o) right at (x_o,y_o). The jumps $[u]$,$[\beta u_n]$ and $[u_\tau]$ are evaluated at the intersect point (x_o,y_o).

The notations used here follow the same rule as that in Eqs. (35) and (36). The auxiliary point on the interface, $u^-_{i,o}$, is not at a grid point so we relate it with the unknowns at grid points by an interpolation:

$$u^-_{i,o} = u^+_{i,o} - [u] = p^+_{0,j} f_{i,j} + p^+_{0,j+1} u_{i,j+1} + p^+_{0,j+2} u_{i,j+2} - [u] + \mathrm{O}(h^3). \tag{52}$$

Nevertheless the interpolation on points $u_{i,j-1}$, $u_{i,j}$ and $f_{i,j+1}$ is also applicable. With this relation, Eq. (51) is changed into

$$\begin{aligned} &C^-_y (p^-_{1,j-1} u_{i,j-1} + p^-_{1,j} u_{i,j} + p^-_{1,j+1} f_{i,j+1}) + [\beta u_n] + \beta^+ \cot\theta [u_\tau] \\ &\quad = C^+_y (p^+_{1,j} f_{i,j} + p^+_{1,j+1} u_{i,j+1} + p^+_{1,j+2} u_{i,j+2}) + C^-_x [w^-_{1,i-2} u_{i-2,o} + w^-_{1,i-1} u_{i-1,o} \\ &\qquad + w^-_{1,i} (p^+_{0,j} f_{i,j} + p^+_{0,j+1} u_{i,j+1} + p^+_{0,j+2} u_{i,j+2} - [u])]. \end{aligned} \tag{53}$$

Also, we are interested in the representations of two fictitious values $f_{i,j}$ and $f_{i,j+1}$ with respect to the known jumps and real unknowns, i.e.,

$$f_{i,j} = \mathbf{C}^j \cdot \mathbf{U}, \tag{54}$$

$$f_{i,j+1} = \mathbf{C}^{j+1} \cdot \mathbf{U}, \tag{55}$$

where $\mathbf{U} = (u_{i,j-1}, u_{i,j}, u_{i,j+1}, u_{i,j+2}, u_{o,i-2}, u_{o,i-1}, [u], [\beta u_n], [u_\tau])$ and the elements of vector $\mathbf{C}^j$, $\mathbf{C}^{j+1}$ are the expansion coefficients of $f_{i,j}$ and $f_{i,j+1}$ with respect to $\mathbf{U}$, respectively. A 2×2 linear system for $\mathbf{C}^i$ and $\mathbf{C}^{j+1}$ can be obtained by replacing $f_{i,j}$ and $f_{i,j+1}$ in Eqs. (50) and (53) with above representations, as follows

$$p^-_{0,j+1} \mathbf{C}^{j+1} \cdot \mathbf{U} - p^+_{0,j} \mathbf{C}^j \cdot \mathbf{U} = \mathbf{K}_1 \cdot \mathbf{U}, \tag{56}$$

$$C^-_y p^-_{1,j+1} \mathbf{C}^{j+1} \cdot \mathbf{U} - (C^+_y p^+_{1,j} + C^-_x w^-_{1,i} p^+_{0,j}) \mathbf{C}^j \cdot \mathbf{U} = \mathbf{K}_2 \cdot \mathbf{U}. \tag{57}$$

It is easy to verify that

$$\begin{aligned} \mathbf{K}_1 &= (-p^-_{0,j-1}, -p^-_{0,j}, p^+_{0,j+1}, p^+_{0,j+2}, 0, 0, -1, 0, 0), \\ \mathbf{K}_2 &= (-C^-_y p^-_{1,j-1}, -C^-_y p^-_{1,j}, C^+_y p^+_{1,j+1} + C^-_x w^-_{1,i} p^+_{0,j+1}, C^+_y p^+_{1,j+2} + C^-_x w^-_{1,i} p^+_{0,j+2}, \\ &\quad C^-_x w^-_{1,i-2}, C^-_x w^-_{1,i-1}, -C^-_x w^-_{1,i}, -1, -\beta^+ \cot\theta). \end{aligned}$$

Once dropping the vector $\mathbf{U}$ from the above system, we obtain the solution of the system again by Cramer's formula:

$$\mathbf{C}^j = \frac{-(C_y^+ p_{1,j}^+ + C_x^- w_{1,i}^- p_{0,j}^+)\mathbf{K}_1 + p_{0,j}^+ \mathbf{K}_2}{C_y^- p_{1,j+1}^- p_{0,j}^+ - p_{0,j+1}^- (C_y^+ p_{1,j}^+ + C_x^- w_{1,i}^- p_{0,j}^+)}, \tag{58}$$

$$\mathbf{C}^{j+1} = \frac{-C_y^- p_{1,j+1}^- \mathbf{K}_1 + p_{0,j+1}^- \mathbf{K}_2}{C_y^- p_{1,j+1}^- p_{0,j}^+ - p_{0,j+1}^- (C_y^+ p_{1,j}^+ + C_x^- w_{1,i}^- p_{0,j}^+)}. \tag{59}$$

The expansions of $\mathbf{C}^j$ and $\mathbf{C}^{j+1}$ solved from this linear system involve the unknowns at auxiliary points $u_{o,i-2}$ and $u_{o,i-1}$, which are not the regular points so we also need to distribute the weights on these two points to the regular grid points. Considering the local topology we choose grid points $u_{i-1,j-1}$, $u_{i-1,j}$ and $u_{i-1,j+1}$ for the interpolation at auxiliary points $u_{o,i-1}$, and $u_{i-2,j-1}$,$u_{i-2,j}$ and $u_{i-2,j+1}$ for $u_{o,i-2}$. With these two interpolation relations:

$$u_{o,i-1} = \mathbf{I}_1^{i-1} u_{i-1,j-1} + \mathbf{I}_2^{i-1} u_{i-1,j} + \mathbf{I}_3^{i-1} u_{i-1,j+1}, \tag{60}$$

$$u_{o,i-2} = \mathbf{I}_1^{i-2} u_{i-2,j-1} + \mathbf{I}_2^{i-2} u_{i-2,j} + \mathbf{I}_3^{i-2} u_{i-2,j+1}, \tag{61}$$

it turns out that both $f_{i,j}$ and $f_{i,j+1}$ have the expansions in terms of unknowns at regular grid points and known jumps, as below

$$\begin{aligned} f_{i,j} = {} & \mathbf{C}_1^j u_{i,j-1} + \mathbf{C}_2^j u_{i,j} + \mathbf{C}_3^j u_{i,j+1} + \mathbf{C}_4^j u_{i,j+2} + \mathbf{C}_5^j (\mathbf{I}_1^{i-1} u_{i-1,j-1} + \mathbf{I}_2^{i-1} u_{i-1,j} + \mathbf{I}_3^{i-1} u_{i-1,j+1}) \\ & + \mathbf{C}_6^j (\mathbf{I}_1^{i-2} u_{i-2,j-1} + \mathbf{I}_2^{i-2} u_{i-2,j} + \mathbf{I}_3^{i-2} u_{i-2,j+1}) + \mathbf{C}_7^j [u] + \mathbf{C}_8^j [\beta u_n] + \mathbf{C}_9^j [u_\tau], \end{aligned} \tag{62}$$

$$\begin{aligned} f_{i,j+1} = {} & \mathbf{C}_1^{j+1} u_{i,j-1} + \mathbf{C}_2^{j+1} u_{i,j} + \mathbf{C}_3^{j+1} u_{i,j+1} + \mathbf{C}_4^{j+1} u_{i,j+2} + \mathbf{C}_5^{j+1} (\mathbf{I}_1^{i-1} u_{i-1,j-1} + \mathbf{I}_2^{i-1} u_{i-1,j} + \mathbf{I}_3^{i-1} u_{i-1,j+1}) \\ & + \mathbf{C}_6^{j+1} (\mathbf{I}_1^{i-2} u_{i-2,j-1} + \mathbf{I}_2^{i-2} u_{i-2,j} + \mathbf{I}_3^{i-2} u_{i-2,j+1}) + \mathbf{C}_7^{j+1} [u] + \mathbf{C}_8^{j+1} [\beta u_n] + \mathbf{C}_9^{j+1} [u_\tau]. \end{aligned} \tag{63}$$

These two fictitious values make the discretization of u_{yy} at point (i,j) and $(i,j+1)$ straightforward

$$(\beta u_y)_y = \frac{\beta_{i,j-\frac{1}{2}}^-}{(\Delta y)^2} u_{i,j-1} - \frac{\beta_{i,j+\frac{1}{2}}^- + \beta_{i,j-\frac{1}{2}}^-}{(\Delta y)^2} u_{i,j} + \frac{\beta_{i,j+\frac{1}{2}}^-}{(\Delta y)^2} f_{i,j+1} \quad \text{at } (i,j), \tag{64}$$

$$(\beta u_y)_y = \frac{\beta_{i,j+\frac{3}{2}}^+}{(\Delta y)^2} u_{i,j+2} - \frac{\beta_{i,j+\frac{3}{2}}^+ + \beta_{i,j+\frac{1}{2}}^+}{(\Delta y)^2} u_{i,j+1} + \frac{\beta_{i,j+\frac{1}{2}}^+}{(\Delta y)^2} f_{i,j} \quad \text{at } (i,j+1). \tag{65}$$

The known terms involving $[u]$, $[u_\tau]$ and $[\beta u_n]$ should also be collected and combined into the right-hand vector in the ultimate linear system $Ax = B$.

The finite difference schemes constructed above are applicable regardless whether the intersection point of the interface with the mesh line is right at the grid point. However, significant simplification can be obtained if the intersection point is at a grid point. When it comes to such a case, the difference scheme for $(\beta u_x)_x + (\beta u_y)_y$ could be generated in a single run rather than separately.

Fig. 5 shows such a situation where grid point (i,j) is an intersect point. Note that when $[u] \neq 0$ the function on the interface is not well defined but from the computational point of view the interface itself can be regarded either as in the interior of the interface or outside of it. Here we regard it as in the interior of the interface, i.e., $u(i,j) = u^-(i,j)$ if a grid point (i,j) is on the interface. In such a case, one only needs to take care of the difference scheme at point (i,j) since both $(i-1,j)$ and $(i,j-1)$ are now regular points while for $(i+1,j)$ and $(i,j+1)$ the jump in $[u]$ can be directly incorporated to generate the difference schemes:

$$(\beta u_x)_x = \frac{\beta_{i+\frac{3}{2},j}^+}{(\Delta x)^2} u_{i+2,j} - \frac{\beta_{i+\frac{3}{2},j}^+ + \beta_{i+\frac{1}{2},j}^+}{(\Delta x)^2} u_{i+1,j} + \frac{\beta_{i+\frac{1}{2},j}^+}{(\Delta x)^2} (u_{i,j} + [u]) \quad \text{at } (i+1,j), \tag{66}$$

$$(\beta u_y)_y = \frac{\beta_{i,j+\frac{3}{2}}^+}{(\Delta y)^2} u_{i,j+2} - \frac{\beta_{i,j+\frac{3}{2}}^+ + \beta_{i,j+\frac{1}{2}}^+}{(\Delta y)^2} u_{i,j+1} + \frac{\beta_{i,j+\frac{1}{2}}^+}{(\Delta y)^2} (u_{i,j} + [u]) \quad \text{at } (i,j+1). \tag{67}$$

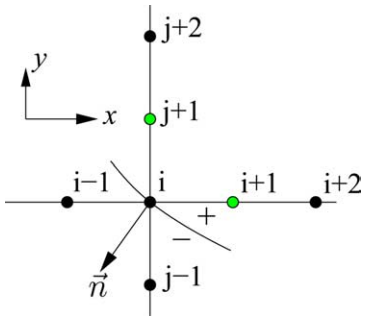

Fig. 5. Irregular point (i,j) and the interface. The interface passes the grid point (i,j). No auxiliary lines and auxiliary points are needed. The jumps $[u]$, $[\beta u_n]$ and $[u_\tau]$ are evaluated at the point (i,j).

For point (i,j) on the interface, however, two fictitious values, $f_{i,j+1}$ and $f_{i+1,j}$, are required in order to formulate the difference schemes for $(\beta u_x)_x + (\beta u_y)_y$ at point (i,j). The expansion coefficients of these two fictitious values with respect to the unknowns at surrounding grid points and the known jumps would also be solved from the approximation equations of the jump conditions at point (i,j). However, since the grid point is now on the interface, the interface conditions adopted are slightly different from what we used earlier. On the one hand, the interpolation representation of the jump relation for u, Eq. (28), does not involve both $f_{i+1,j}$ and $f_{i,j+1}$ any more so this jump condition does not provide an approximate equation for the fictitious values. On the other hand, two target fictitious values make the approximation possible for all derivatives, u_x^+, u_x^-, u_y^+ and u_y^-. Therefore the cancellation of one of Eqs. (29) and (30) is no longer necessary and these two conditions exactly provide two approximate equations for $f_{i,j+1}$ and $f_{i+1,j}$, as follows

$$\begin{aligned} &-(w^-_{1,i-1}u_{i-1,j} + w^-_{1,i}u_{i,j} + w^-_{1,i+1}f_{i+1,j})\sin\theta + (p^-_{1,j-1}u_{i,j-1} + p^-_{1,j}u_{i,j} + p^-_{1,j+1}f_{i,j+1})\cos\theta + [u_\tau] \\ &\quad = -[w^+_{1,i}(u_{i,j} + [u]) + w^+_{1,i+1}u_{i+1,j} + w^+_{1,i+2}u_{i+2,j}]\sin\theta + [p^+_{1,j}(u_{i,j} + [u]) + p^+_{1,j+1}u_{i,j+1} + p^+_{1,j+2}u_{i,j+2}]\cos\theta, \end{aligned} \tag{68}$$

$$\begin{aligned} &\beta^-(w^-_{1,i-1}u_{i-1,j} + w^-_{1,i}u_{i,j} + w^-_{1,i+1}f_{i+1,j})\cos\theta + \beta^-(p^-_{1,j-1}u_{i,j-1} + p^-_{1,j}u_{i,j} + p^-_{1,j+1}f_{i,j+1})\sin\theta + [\beta u_n] \\ &\quad = \beta^+(w^+_{1,i}(u_{i,j} + [u]) + w^+_{1,i+1}u_{i+1} + w^+_{1,i+2}u_{i+2})\cos\theta + \beta^+(p^+_{1,j}(u_{i,j} + [u]) + p^+_{1,j+1}u_{i,j+1} + p^+_{1,j+2}u_{i,j+2})\sin\theta. \end{aligned} \tag{69}$$

These two equations consist a 2×2 linear system for vectors $\mathbf{C}^{i+1}$ and $\mathbf{C}^{j+1}$, which are the respective expansion coefficients of fictitious values $f_{i+1,j}$ and $f_{i,j+1}$, i.e., $f_{i+1,j} = \mathbf{C}^{i+1}\cdot\mathbf{U}$, $f_{i,j+1} = \mathbf{C}^{j+1}\cdot\mathbf{U}$ with $\mathbf{U} = (u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, u_{i,j-1}, u_{i,j+1}, u_{i,j+2}, [u], [\beta u_n], [u_\tau])$. The solution of this linear system is

$$-w^-_{1,i+1}\sin\theta\,\mathbf{C}^{i+1} + p^-_{1,j+1}\cos\theta\,\mathbf{C}^{j+1} = \mathbf{K}_1, \tag{70}$$

$$\beta^- w^-_{1,i+1}\cos\theta\,\mathbf{C}^{i+1} + \beta^- p^-_{1,j+1}\sin\theta\,\mathbf{C}^{j+1} = \mathbf{K}_2, \tag{71}$$

where

$$\begin{aligned} \mathbf{K}_1 = (&w^-_{1,i-1}\sin\theta, (w^-_{1,i} - w^+_{1,i})\sin\theta + (p^+_{1,j} - p^-_{1,j})\cos\theta, -w^+_{1,i+1}\sin\theta, -w^+_{1,i+2}\sin\theta, -p^-_{1,j-1}\cos\theta \\ &p^+_{1,j+1}\cos\theta, p^+_{1,j+2}\cos\theta, p^+_{1,j}\cos\theta - w^+_{1,i}\sin\theta, 0, -1), \\ \mathbf{K}_2 = (&-\beta^- w^-_{1,i-1}\cos\theta, (\beta^+ w^+_{1,i} - \beta^- w^-_{1,i})\cos\theta + (\beta^+ p^+_{1,j} - \beta^- p^-_{1,j})\sin\theta, \beta^+ w^+_{1,i+1}\cos\theta, \\ &\beta^+ w^+_{1,i+2}\cos\theta, -\beta^- p^-_{1,j-1}\sin\theta, \beta^+ p^+_{1,j+1}\sin\theta, \beta^+ p^+_{1,j+2}\sin\theta, \beta^+(w^+_{1,i}\cos\theta + p^+_{1,j}\sin\theta), -1, 0), \end{aligned}$$

and

$$\mathbf{C}^{i+1} = -\frac{\beta^- p^-_{1,j+1}\sin\theta\ \mathbf{K}_1 - p^-_{1,j+1}\cos\theta\ \mathbf{K}_2}{\beta^- w^-_{1,i+1}p^-_{1,j+1}}, \tag{72}$$

$$\mathbf{C}^{j+1} = \frac{\beta^- w^-_{1,i+1}\cos\theta\ \mathbf{K}_1 + w^-_{1,i+1}\sin\theta\ \mathbf{K}_2}{\beta^- w^-_{1,i+1}p^-_{1,j+1}}. \tag{73}$$

Thus, we obtain the full representations of $f_{i+1,j}$ and $f_{i,j+1}$

$$\begin{aligned} f_{i+1,j} &= \mathbf{C}_1^{i+1}u_{i-1,j} + \mathbf{C}_2^{i+1}u_{i,j} + \mathbf{C}_3^{i+1}u_{i+1,j} + \mathbf{C}_4^{i+1}u_{i+2,j} + \mathbf{C}_5^{i+1}u_{i,j-1} + \mathbf{C}_6^{i+1}u_{i,j+1} \\ &\quad + \mathbf{C}_7^{i+1}u_{i,j+2} + \mathbf{C}_8^{i+1}[u] + \mathbf{C}_9^{i+1}[\beta u_n] + \mathbf{C}_{10}^{i+1}[u_\tau], \end{aligned} \tag{74}$$

$$\begin{aligned} f_{i,j+1} &= \mathbf{C}_1^{j+1}u_{i-1,j} + \mathbf{C}_2^{j+1}u_{i,j} + \mathbf{C}_3^{j+1}u_{i+1,j} + \mathbf{C}_4^{j+1}u_{i+2,j} + \mathbf{C}_5^{j+1}u_{i,j-1} + \mathbf{C}_6^{j+1}u_{i,j+1} \\ &\quad + \mathbf{C}_7^{j+1}u_{i,j+2} + \mathbf{C}_8^{j+1}[u] + \mathbf{C}_9^{j+1}[\beta u_n] + \mathbf{C}_{10}^{j+1}[u_\tau] \end{aligned} \tag{75}$$

and the difference scheme for $(\beta u_x)_x + (\beta u_y)_y$ at point (i,j) is

$$(\beta u_x)_x + (\beta u_y)_y = \frac{\beta^-_{i+\frac{1}{2},j}}{(\Delta x)^2}f_{i+1,j} - \frac{\beta^-_{i+\frac{1}{2},j} + \beta^-_{i-\frac{1}{2},j}}{(\Delta x)^2}u_{i,j} + \frac{\beta^-_{i-\frac{1}{2},j}}{(\Delta x)^2}u_{i-1,j} + \frac{\beta^-_{i,j+\frac{1}{2}}}{(\Delta y)^2}f_{i,j+1}, - \frac{\beta^-_{i,j+\frac{1}{2}} + \beta^-_{i,j-\frac{1}{2}}}{(\Delta y)^2}u_{i,j} + \frac{\beta^-_{i,j-\frac{1}{2}}}{(\Delta y)^2}u_{i,j-1}. \tag{76}$$

It is noted that the representation scheme described above has a local truncation error of $O(h^3)$ so that the central finite difference scheme for the second-order derivatives have a local truncation error of $O(h)$ at irregular points. As the number of irregular grid points are one-dimension lower than the number of total grid points, the approximation error of the proposed MIB method is of $O(h^2)$, which has been verified in our numerical tests.

2.2.2. Higher order MIB schemes for irregular interfaces

The generalization of the proposed MIB method to higher order convergence is quite straightforward. In the present approach, a high order scheme means the use of standard high order finite difference discretizations in the whole computational domain. In the vicinity of the interface, irregular grid points are first identified according to the finite difference discretization. Then fictitious points are created at the same locations of the irregular points. The function values at fictitious points are determined by enforcing the interface jump conditions. For high order schemes, the set of jump conditions is repeatedly used to determine the required set of fictitious values. At each irregular grid point, this procedure is exactly the same as what described for the straight interfaces.

Let us consider a fourth-order case. We start with the interface condition (31) and the stencil shown in Fig. 6 to derive a fourth-order scheme for u_{xx} and u_x by using following steps:

1. Use $u_{i-3,j},\ldots,u_{i,j}$ and $f_{i+1,j}$ as a stencil to approximate u^- and u^-_x. For u^+ and u^+_x we choose stencil $f_{i,j},u_{i+1,j},\ldots,u_{i+4,j}$. Here, u^+_y is discretized on auxiliary points $(o,j),\ldots,(o,j+4)$, to ensure an accuracy of $O(h^5)$.
2. Solve the 2×2 linear system resulting from Step 1 for the representations of $f_{i,j}$ and $f_{i+1,j}$.
3. Use $u_{i-3,j},\ldots,u_{i,j}$ and $f_{i+1,j}f_{i+2,j}$ to approximate u^- and u^-_x, and use $f_{i-1,j}f_{i,j}$ and $u_{i+1,j},\ldots,u_{i+4,j}$ to approximate u^+ and u^+_x. Use the same approximation for u^+_y as Step 1. Note that at this moment, both $f_{i,j}$ and $f_{i+1,j}$ are known.
4. Solve the 2×2 linear system resulting from Step 3 for the representations of $f_{i-1,j}$ and $f_{i+2,j}$.
5. Substitute appropriate terms of $f_{i-1,j}, f_{i,j}, f_{i+1,j}, f_{i+2,j}$ for the values at irregular points when the standard fourth-order central discretization of u_x or u_{xx} crosses the interface.

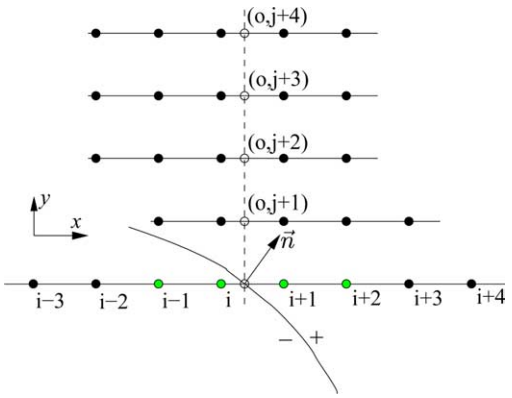

Fig. 6. A typical stencil used in constructing a fourth-order scheme for u_{xx} and u_x. There are two pairs of fictitious points in this case: $f_{i,j}$, $f_{i+1,\,j}$ and $f_{i-1,j}$, $f_{i+2,j}$.

This iterative procedure is systematic and is of arbitrarily high order provided there are enough regular points available on both side of the interface. For example, a sixth-order discretization of u_{xx} and u_x can be constructed by using the stencil illustrated in Fig. 7. In this case, we successively solve the representations for three pairs of fictitious values $(f_{i,j}, f_{i+1,j})$, $(f_{i-1,j}, f_{i+2,j})$ and $(f_{i-2,j}, f_{i+3,j})$. It is noted that in constructing high order MIB methods, one needs to ensure that the local truncation error at the irregular point is one order lower than the designed global order of accuracy. The nature of high order accuracy of these schemes can be appreciated from the numerical examples presented in Section 4.

It is noted that for a general interface the difference scheme generated by the MIB involves more nodes than the IIM and of course, more than the standard central difference scheme. The number of grid points and distribution of the MIB stencils vary with the underlying mesh and the local extension of the interface relative to the mesh. Also, the resulting linear system is no longer symmetric and diagonally dominant, as in the case of the IIM. Nevertheless, all the iterative solvers we tested, no matter the successive overrelaxation (SOR) or the preconditioned biconjugate gradient (PBCG), always yield the solution with favorable convergence rate. Detailed comparison in terms of accuracy and CPU time is given in Section 4.

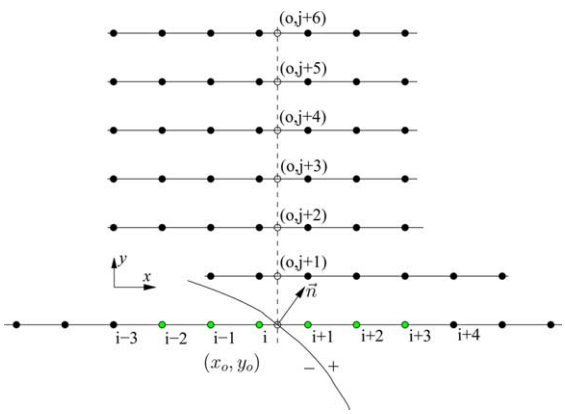

Fig. 7. A typical stencil used in constructing a sixth-order scheme for u_{xx} and u_x. Now there are three pairs of fictitious points in this case: $f_{i,j}$,$f_{i+1,j}$ and $f_{i-1,j}$, $f_{i+2,j}$ as well as $f_{i-2,j}$, $f_{i+3,j}$.

3. Numerical experiments on regular interfaces

In this section, we test the performance of the proposed MIB method for regular elliptic interface problems. Benchmarked with analytical solution, both the order of accuracy and the computational efficiency are investigated for the MIB method through 1D, 2D, and 3D studies. However, only 3D results are reported. The standard central FD scheme is used for a comparison. The standard preconditioned biconjugate gradient (PBCG) method [41] with the preconditioner being the diagonal part of the discrete matrix is employed to solve algebraic systems obtained from both the MIB and FD discretizations. In all studied cases, the interfaces are not laid on the grid nodes.

Consider the 3D elliptic equation

$$(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z - \beta k^2 u = q, \quad (x,y,z) \in \Omega = [0,1] \times [0,1] \times [0,1], \tag{77}$$

with Dirichlet boundary conditions along the x- and z-directions and Neumann boundary conditions along the y-direction. Here k is a constant. Suppose that the media are only inhomogeneous along the x direction, i.e.,

$$\beta = \begin{cases} \beta_1, & \Omega_1 = \{(x,y,z) | 0 \leqslant x < \alpha,\ 0 \leqslant y \leqslant 1,\ 0 \leqslant z \leqslant 1\}, \\ \beta_2, & \Omega_2 = \{(x,y,z) | \alpha \leqslant x \leqslant 1,\ 0 \leqslant y \leqslant 1,\ 0 \leqslant z \leqslant 1\}. \end{cases} \tag{78}$$

We consider a sinusoidal source

$$q(x,y,z) = \begin{cases} C_1 \sin(n\pi x)\cos(m\pi y)\sin(j\pi z), & (x,y,z) \in \Omega_1, \\ C_1 \sin(n\pi x)\cos(m\pi y)\sin(j\pi z) - C_2\beta_2 \cos(m\pi y)\sin(j\pi z), & (x,y,z) \in \Omega_2, \end{cases} \tag{79}$$

where n, m and j are integers and $C_1 = -n^2\pi^2 - m^2\pi^2 - j^2\pi^2 - k^2$, $C_3 = \sin(n\pi\alpha)(1/\beta_1 - 1/\beta_2)$, and $C_2 = C_3(m^2\pi^2 + j^2\pi^2 + k^2)$. With such a source term, the analytical solution of Eq. (77) reads

$$u(x,y,z) = \begin{cases} \frac{1}{\beta_1} \sin(n\pi x)\cos(m\pi y)\sin(j\pi z), & (x,y,z) \in \Omega_1, \\ \frac{1}{\beta_2} \sin(n\pi x)\cos(m\pi y)\sin(j\pi z) + C_3 \cos(m\pi y)\sin(j\pi z), & (x,y,z) \in \Omega_2. \end{cases} \tag{80}$$

The constant C_3 ensures that solution $u(x,y,z)$ is continuous across the interface $x = \alpha$, and the jump conditions are given by

$$[u] = 0, \quad \text{and} \quad [\beta u_x] = 0. \tag{81}$$

It can be derived from the analytical solution (80) that, the boundary conditions are $u(0,y,z) = 0$, $u(1,y,z) = C_3 \cos(m\pi y)\sin(j\pi z)$, $\frac{\partial u}{\partial y}|_{y=0} = 0$, $\frac{\partial u}{\partial y}|_{y=1} = 0$, $u(x,y,0) = 0$, and $u(x,y,1) = 0$.

We first examine the numerical convergence orders of the FD and MIB methods. A uniform mesh is used with size $N = N_x = N_y = N_z$. The maximum error and the overall accuracy of the approximation are measured, respectively, in terms of the standard L_∞ norm,

$$L_\infty = \max_{i=1,\ldots,N^3} |u_i - \tilde{u}_i|,$$

and the standard L_2 norm,

$$L_2 = \sqrt{\frac{1}{N^3}\sum_{i=1}^{N^3}(u_i - \tilde{u}_i)^2},$$

where u_i is the analytical solution and $\tilde{u}_i$ is the numerical solution. We are interested in examining the performance of the proposed scheme for both low β contrast and high β contrast across the interface. These results are presented in Tables 1 and 2, respectively. In comparing the MIB results in both tables, it can be

Table 1
Numerical convergence tests of the 3D elliptic equation with low contrast β

Scheme	N^3	L_2			L_∞		
		30^3	60^3	120^3	30^3	60^3	120^3
FD	Error	9.33(−2)	2.24(−2)	7.62(−3)	3.87(−1)	1.65(−1)	1.17(−1)
	Order		2.06	1.56		1.23	0.49
2nd-order MIB	Error	9.81(−2)	2.05(−2)	4.88(−3)	3.86(−1)	8.01(−2)	1.90(−2)
	Order		2.26	2.07		2.27	2.07
4th-order MIB	Error	3.14(−2)	2.19(−3)	1.41(−4)	1.29(−1)	8.71(−3)	5.55(−4)
	Order		3.84	3.96		3.89	3.97
8th-order MIB	Error	9.99(−3)	4.61(−5)	2.03(−7)	5.94(−2)	1.83(−4)	8.31(−7)
	Order		7.76	7.83		8.34	7.79
16th-order MIB	Error	4.81(−2)	1.27(−6)	8.54(−12)	4.77(−1)	1.60(−5)	1.66(−10)
	Order		15.20	17.19		14.86	16.56

The model parameters are chosen as $k = 1$, $n = 18$, $m = 18$, $j = 18$, $\alpha = 0.5$, $\beta_1 = 4$ and $\beta_2 = 1$. For the 2nd-, 4th-, 8th-, and 16th-order MIB, (M,L) is set to be (1,1), (2,4), (4,9), and (8,15), respectively.

Table 2
Numerical convergence tests of the 3D elliptic equation with high contrast β

Scheme	N^3	L_2			L_∞		
		30^3	60^3	120^3	30^3	60^3	120^3
FD	Error	1.12(−1)	3.40(−2)	1.25(−2)	7.17(−1)	4.35(−1)	2.27(−1)
	Order		1.72	1.45		0.72	0.94
2nd-order MIB	Error	9.52(−2)	1.99(−2)	4.74(−3)	3.86(−1)	8.01(−2)	1.90(−2)
	Order		2.26	2.07		2.27	2.07
4th-order MIB	Error	3.05(−2)	2.13(−3)	1.37(−4)	1.29(−1)	8.71(−3)	5.55(−4)
	Order		3.84	3.96		3.89	3.97
8th-order MIB	Error	9.70(−3)	4.47(−5)	1.97(−7)	5.94(−2)	1.83(−4)	8.31(−7)
	Order		7.76	7.83		8.34	7.79
16th-order MIB	Error	4.66(−2)	1.24(−6)	8.30(−12)	4.77(−1)	1.60(−5)	1.67(−10)
	Order		15.20	17.18		14.86	16.55

The model parameters are chosen as $k = 1$, $n = 18$, $m = 18$, $j = 18$, $\alpha = 0.5$, $\beta_1 = 80$ and $\beta_2 = 1$. For the 2nd-, 4th-, 8th-, and 16th-order MIB, (M,L) is set to be (1,1), (2,4), (4,9), and (8,15), respectively.

observed that the L_∞ errors and orders of both low and high contrast cases are exactly the same. Due to the large variation in magnitude of the solutions in two contrast cases, the L_2 errors of the MIB in two tables are slightly different. But the L_2 errors and their orders are still very similar. This suggests that the MIB approach yields essentially the same result, no matter how large the jump in solution is. This further means that the numerical error of the MIB is induced solely by the oscillatory nature of the solution, rather than by the interface. Therefore, the accuracy of the high order FD schemes is fully restored by the MIB method in these studies. Up to 16th orders are achieved numerically by the MIB approach for this 3D elliptic interface problem. However, without the proper interface modeling, the FD accuracy is at most of first-order in L_∞ norm measure, and becomes worse when the contrast in β is higher. The projection of the MIB solution for the 3D elliptic equation at $z = 1/64$ are shown in Fig. 8.

We next present an efficiency study which clearly illustrates the computational efficiency gained by using the higher order MIB approach, in place of low order schemes, see Table 3. Successive mesh refinements are considered towards the accuracy level $L_\infty \approx 0.1\%$ for the FD and MIB methods. This desired level

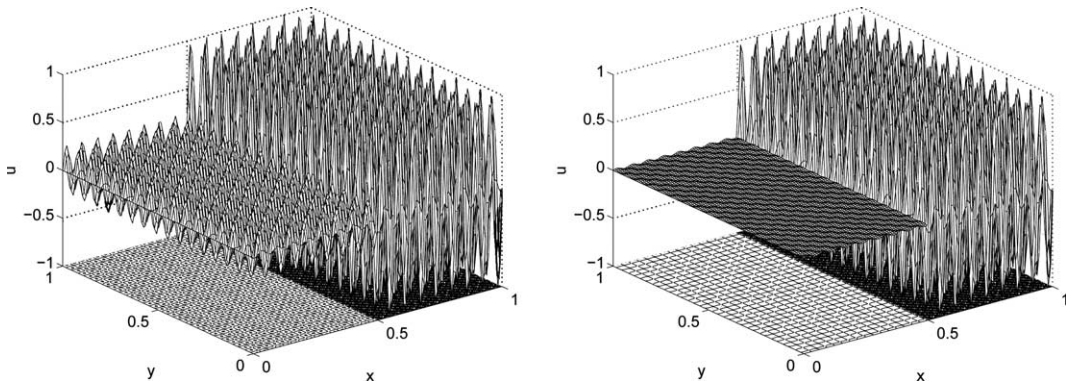

Fig. 8. The projection of the MIB solution of the 3D elliptic equation at $z = 1/64$, with $(M,L) = (8,17)$, $N^3 = 120^3$, $k = 1$, $n = 32$, $m = 32$, $j = 32$ and $\alpha = 0.5$. Left: Low contrast case ($\beta_1 = 4$ and $\beta_2 = 1$); Right: High contrast case ($\beta_1 = 80$ and $\beta_2 = 1$).

Table 3
Numerical efficiency tests of the 3D elliptic equation

N^3	FD			2nd-order MIB			12th-order MIB
	60^3	120^3	240^3	60^3	120^3	240^3	30^3
L_2	2.21(−2)	9.02(−3)	4.07(−3)	8.61(−3)	2.09(−3)	5.18(−4)	1.71(−4)
Order		1.29	1.15		2.04	2.01	
L_∞	3.01(−1)	1.53(−1)	7.66(−2)	3.47(−2)	8.40(−3)	2.08(−3)	1.83(−3)
Order		0.98	1.00		2.05	2.02	
CPU	3.14	185.08	4731.63	0.75	9.66	93.13	0.28

The model parameters are chosen as $k = 1$, $n = 12$, $m = 12$, $j = 12$, $\alpha = 0.5$, $\beta_1 = 80$ and $\beta_2 = 1$. For the 2nd and 12th-order MIB, (M,L) is set to be (1,1) and (6,7), respectively. CPU time in second is reported.

of accuracy is actually very challenging for low order methods in 3D, because the memory requirement goes quickly beyond the limit of our computer resource after only a few refinements. In particular, a 3D computation with a huge mesh size of 480^3 impractically requires over 2 Gbytes memory, so that the last three mesh refinements that are feasible are just until 240^3, as shown in Table 3. It is noted that in the PBCG iterative solver [41], only non-zero elements are stored for the large sparse matrix resulting from the PDE discretization. We also note that the FD scheme does not require the PBCG iterative solver. A small reduction in the CPU time can be gained for the FD scheme if a simpler iterative solver is used. It is clear from Table 3 that the 12th-order MIB delivers the desired accuracy by using only a very small mesh size of 30^3 and 0.28 sec CPU time. The 16th-order MIB can also achieve the same goal by using the same mesh and slightly more CPU time. However, to almost reach the same accuracy, the second-order MIB scheme requires a large mesh size of 240^3 and 93.13 s computing time. In other words, our 12th-order approach is about at least 332 times more efficient in terms of CPU time than a second-order method, such as a second-order MIB or the IIM. By using a huge mesh size of 240^3 and 4731.63 s execution time, the FD accuracy is still far away from the target one. It is really hopeless to achieve that goal by using such a slowly converged method. Moreover, it is interesting to note that based on the same mesh size, and using stencils of the same length, the execution time of the second-order MIB is usually much smaller than that of the standard FD method, although the same standard PBCG solver [41] is employed in both methods. In fact, similar behavior has been observed in our 1D and 2D studies, too. This suggests that the presence of the

interface will not only degrade all numerical scheme to first-order, but also slow down the convergence speed of the iterative solver, whereas both impairments can be recovered after using the MIB method.

4. Numerical experiments on irregular interfaces

In this section, we examine the performance of the proposed MIB scheme for 2D irregular interface problems by considering six case studies with different boundary and interface geometry. We will focus on the standard second-order MIB method in the first 4 cases, while the last two cases are devoted to the validation of high order MIB schemes. Numerical results are compared to the analytical solutions of the equation, in terms of both numerical accuracy and computational efficiency. The IIM of LeVeque and Li [25] is regenerated for a comparison in some test cases. The performance of our IIM code has been verified with that in the literature [25] and is found to be similar to a later version of the IIM given by Li and Ito [29]. The PBCG solver [41] is exclusively adopted to solve the linear system due to its efficiency and the simplicity in implementation. The standard L_∞ norm error measurement is employed in this section.

Case 1. We are interested in the 2D Poisson equation

$$(\beta u_x)_x + (\beta u_y)_y = q(x,y) \tag{82}$$

defined in a square $[-1,1]\times[-1,1]$ with a circular interface $r^2 \equiv x^2 + y^2 = \frac{1}{4}$ inside. Following [25], the exact solution is designed to be

$$u(x,y) = \begin{cases} x^2 + y^2, & r \leqslant 0.5, \\ \frac{1}{4}\left(1 - \frac{1}{8b} - \frac{1}{b}\right) + \left(\frac{r^4}{2} + r^2\right)/b, & \text{otherwise}, \end{cases} \tag{83}$$

with the diffusion coefficient

$$\beta(x,y) = \begin{cases} 2, & r \leqslant 0.5, \\ b, & \text{otherwise}. \end{cases} \tag{84}$$

Such a designated solution forces the discontinuous inhomogeneous term $q(x,y)$ to be

$$q(x,y) = \begin{cases} 8.0, & r \leqslant 0.5, \\ 8(x^2 + y^2) + 4.0, & \text{otherwise}. \end{cases} \tag{85}$$

Let $b = 10$ such that $u(x,y)$ is continuous throughout the domain and $[\beta u_n] = -0.75$ on the interface. The computed result with a 20×20 mesh is plotted in Fig. 9. Table 4 lists the computed error of the second-order MIB scheme in a comparison with the results of the IIM. Both methods have very clear second-order accuracy. The MIB delivers a slightly more accurate result than that of the IIM. It is found that the CPU times used for generating the local immersed grids are almost the same for both methods although the underlying algorithms are different. The CPU times used for solving the linear algebraic equation systems are not compared because the IIM method has its own optimal solver [29], while no such solver is available for the MIB yet. It is expected the IIM is faster since it involves fewer irregular nodes.

Case 2. In this case, we solve the Laplace equation $u_{xx} + u_{yy} = 0$, which is defined in square $[-1,1]\times[-1,1]$ and has the following analytical solution as designed:

$$u(x,y) = \begin{cases} e^x \cos(y), & r \leqslant 0.5, \\ 0, & \text{otherwise}, \end{cases} \tag{86}$$

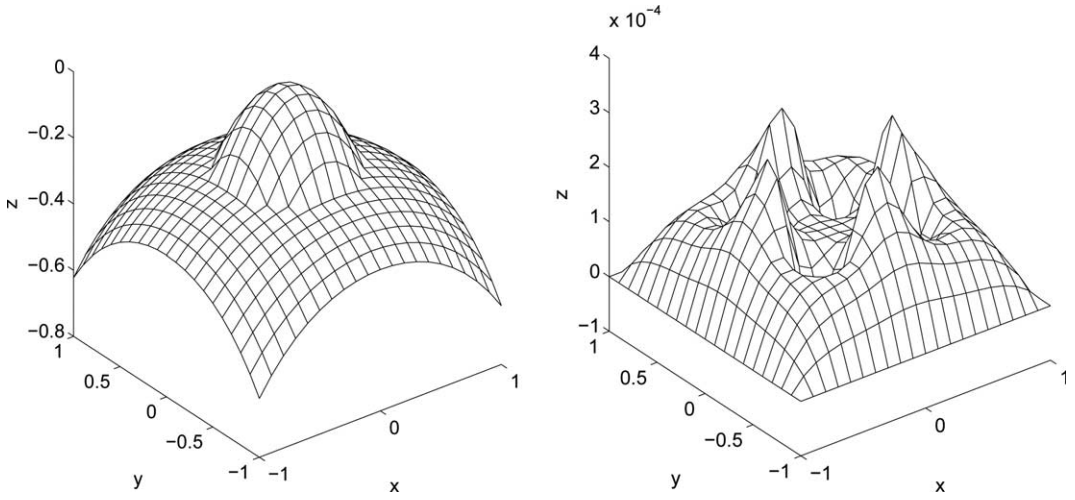

Fig. 9. Computed solution (left) and the error (right) for the 2D Poisson equation (Case 1). $\beta = 10$, $[u] = 0$, $[\beta u_n] = -0.75$.

Table 4
Numerical efficiency tests of the 2D Poisson equation (Case 1)

$n_x \times n_y$	Second -rder MIB		IIM	
	L_∞	Order	L_∞	Order
20×20	2.852(−4)		2.167(−3)	
40×40	7.707(−5)	1.9	5.000(−4)	2.1
80×80	2.069(−5)	1.9	1.131(−4)	2.1
160×160	5.131(−6)	2.0	2.748(−5)	2.0
320×320	1.257(−6)	2.0	6.781(−6)	2.0

$b = 10$, $[u] = 0$, $[\beta u_n] = -0.75$.

The jumps in u and u_n along the interface $r = \frac{1}{2}$ can be evaluated from this solution. Note that here we have unit diffusion coefficient throughout the whole domain.

Table 5 gives the computational results of the second-order MIB and the IIM for a comparison. Again, as expected, a steady second-order convergence is validated for both methods. Moreover, MIB scheme is slightly favored due to the smaller numerical error on all the five successively refined meshes. The computed result in Fig. 10 sharply features designated discontinuity along the circular interface.

Case 3. Here, we solve the Poisson equation with elliptical interface Γ given as

$$\left(\frac{x}{18/27}\right)^2 + \left(\frac{y}{10/27}\right)^2 = 1. \tag{87}$$

It admits the exact solution

$$u(x,y) = \begin{cases} e^x \cos(y), & \text{inside } \Gamma, \\ 5\exp(-x^2 - \frac{y^2}{2}), & \text{otherwise.} \end{cases} \tag{88}$$

We leave the diffusion coefficient to be tuned to produce different contrast ratios. Outside the interface Γ, we fix the diffusion coefficient to be 1. In the first experiment we choose $\beta = 10$ for the inside whilst a much large $\beta = 1000$ is chosen in the second experiment. The computed solution and error for the large contrast

Table 5
Numerical efficiency tests of the 2D Laplace equation (Case 2)

$n_x \times n_y$	Second-order MIB		IIM	
	L_∞	Order	L_∞	Order
20×20	1.015(−4)		4.389(−4)	
40×40	2.511(−5)	2.0	1.079(−4)	2.0
80×80	6.369(−6)	2.0	2.778(−5)	2.0
160×160	1.608(−6)	2.0	7.500(−6)	1.9
320×320	3.714(−7)	2.1	1.740(−6)	2.1

$\beta \equiv 1$.

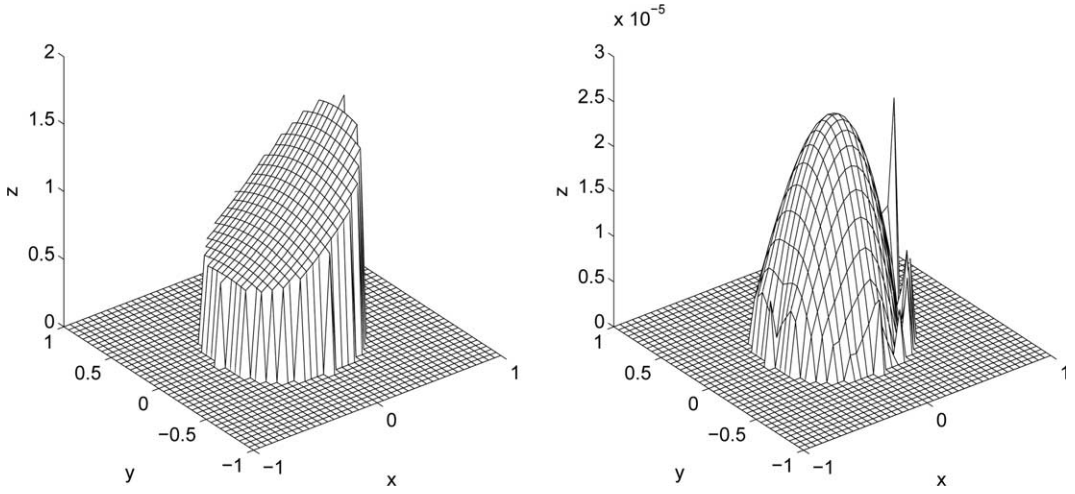

Fig. 10. Computed solution (left) and the error (right) for the 2D Laplace equation (Case 2). $\beta \equiv 1$.

Table 6
Accuracy assessment of the MIB method for the Poisson equation with elliptical interface (Case 3)

Second-order MIB			JCCS		
$n_x \times n_y$	L_∞	Order	$n_r \times n_\theta$	L_∞	Order
20×20	2.659(−2)		23×32	1.755(−2)	
40×40	5.206(−3)	2.4	41×64	4.961(−3)	1.8
80×80	1.487(−3)	1.8	77×128	1.352(−3)	1.9
160×160	3.746(−4)	2.0	149×256	3.548(−4)	1.9
320×320	7.803(−5)	2.3	293×512	9.096(−5)	2.0

$\beta = 10$ inside of Γ; $\beta = 1$ outside of Γ. For the JCCS, a polar coordinate is used and the elliptical interface in the physical domain is mapped onto a circular interface in the computational domain. n_r or n_θ is the number of mesh points in respective direction.

ratio are plotted in Fig. 11. The numerical errors are also given in Tables 6 and 7 in a comparison with the results of the jump condition capturing scheme (JCCS) obtained by Wang [48]. The JCCS is constructed based on the GFM.

For the case with $\beta = 10$ in the inside, two methods produce results with very close errors. For $\beta = 1000$ in the inside, however, the results of the MIB are much more accurate than those of the JCCS, particularly, a significant increasing was found in the numerical error of the JCCS accompanying this improvement of the ratio of diffusion coefficients. This indicates that the MIB is much less sensitive to the difference of β on

Table 7
Accuracy assessment of the MIB method for the Poisson equation with elliptical interface (Case 3)

Second-order MIB			JCCS		
$n_x \times n_y$	L_∞	Order	$n_r \times n_\theta$	L_∞	Order
22×22	9.130(−2)		23×32	2.803	
40×40	2.764(−2)	1.9	41×64	7.543(−1)	1.9
80×80	7.524(−3)	1.9	77×128	1.940(−1)	2.0
160×160	2.169(−3)	2.0	149×256	4.906(−2)	2.0
320×320	4.841(−4)	2.2	293×512	1.232(−2)	2.0

$\beta = 1000$ inside of Γ; $\beta = 1$ outside of Γ.

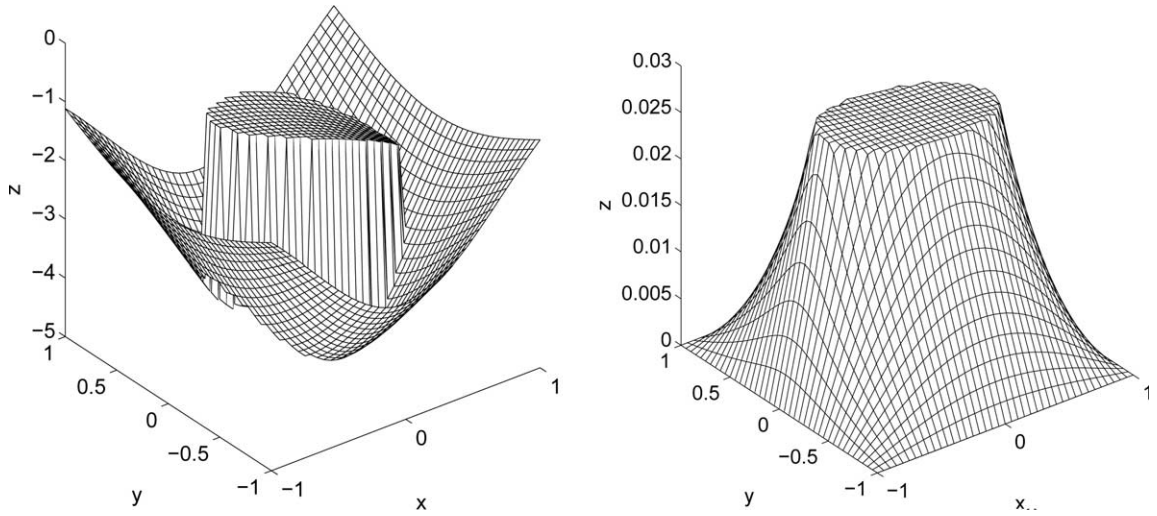

Fig. 11. Computed solution (left) and the error (right) for the 2D Poisson equation with elliptical interface Γ (Case 3). $\beta = 1000$ inside of Γ; $\beta = 1$ outside of Γ.

the two sides of the interface thus would be more robust and accurate for the problem with a large jump in the diffusion coefficient.

Case 4. As in the preceding example, we illustrate the usefulness of the second-order MIB method for an elliptic problem with a more complicated interface geometry. In this case, the treatment of the physical irregular domain is similar to that in Case 2, i.e., by embedding the physical irregular domain into a slightly larger rectangular domain and treating the physical irregular boundary as an interface inside the new domain. Thus one obtains an elliptic interface problem favoring interface methods. The necessary interface conditions can be derived from the given boundary conditions of the original problem. In our case, the physical irregular boundary Γ is defined to be

$$r = 0.5(1 + 0.5\sin(6\theta)) \tag{89}$$

and the analytical solution of the Poisson equation in the extended domain $[-1,1]\times[-1,1]$ is

$$u(x,y) = \begin{cases} \frac{1}{4} + \sin(x)\sin(y), & \text{inside } \Gamma, \\ 0, & \text{outside } \Gamma, \end{cases} \tag{90}$$

with $\beta = 1$.

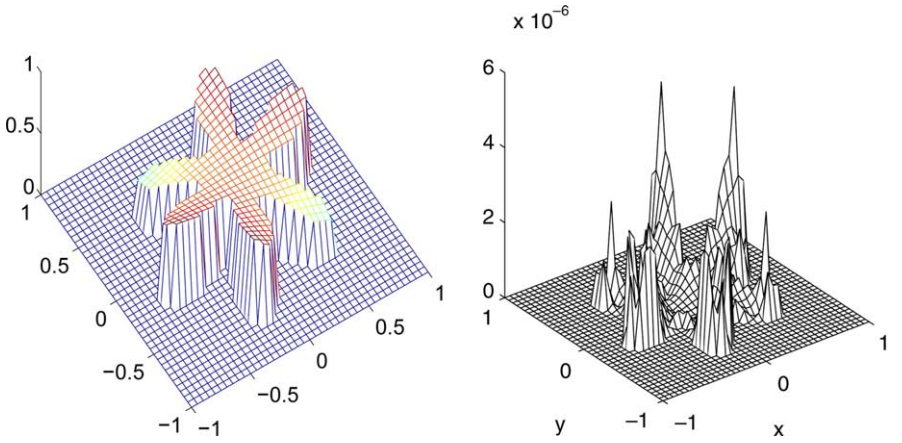

Fig. 12. Computed solution and the error of Case 4.

In computation, the value of $u(x,y)$ outside of Γ is set to be zero as it is already known. The computed error for the accuracy assessment is very similar to earlier cases so its table is omitted here. The numerical solution on a 40×40 mesh is plotted in Fig. 12, together with the plot of the absolute error whose maximum is about 4.7×10^{-6}.

Case 5. In this case, we demonstrate the second- and fourth-order MIB schemes. The interface is given as $\Gamma(x) = 0.2\tanh(\exp(1/(1-x)) - \exp(1/(1+x)))$. The exact solution of the Poisson equation is

$$u(x,y) = \begin{cases} \sin((y-\Gamma(x)))\cos(x), & y \geqslant \Gamma(x), \\ \sin(4(y-\Gamma(x)))\cos(x)+1, & \text{otherwise.} \end{cases} \tag{91}$$

The value of β is prescribed to be 5 above the $\Gamma(x)$ and 1 below, which gives a non-vanishing flux discontinuity at the interface. Fig. 13 (left) depicts the computed solution on a 40×40 grid. The L_∞ norm errors of both the second- and fourth-order schemes are tabulated in Table 8.

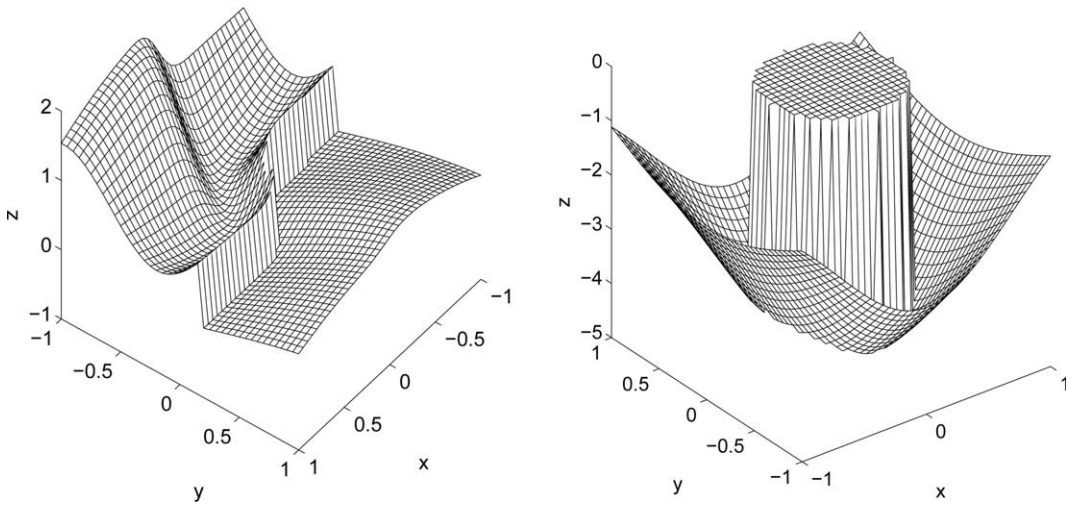

Fig. 13. The solution of the high order MIB method for the 2D Poisson equation. Left: Case 5; Right: Case 6, $-u(x,y)$ is plotted.

Table 8
Accuracy assessment of the second- and fourth-order MIB schemes for the Poisson equation in Case 5

$n_x \times n_y$	Second-order MIB		Fourth-order MIB	
	L_∞	Order	L_∞	Order
40×40	7.44(−3)		1.40(−3)	
80×80	1.68(−3)	2.1	5.76(−5)	4.6
160×160	4.99(−4)	1.8	3.82(−6)	3.9
320×320	1.37(−4)	1.9	2.19(−7)	4.1

Table 9
Comparison of the second-, fourth- and sixth-order MIB schemes for the Poisson equation (Case 6)

$n_x \times n_y$	Second-order MIB			Fourth-order MIB			Sixth-order MIB		
	L_∞	Order	CPU	L_∞	Order	CPU	L_∞	Order	CPU
20×20	1.20(−3)		0.155 (0.008)	2.47(−5)		0.552 (0.02)			
40×40	2.85(−4)	2.07	1.45 (0.026)	1.51(−6)	4.03	3.714 (0.07)	3.48(−8)		10.94 (0.15)
80×80	6.55(−5)	2.12	11.18 (0.089)	6.16(−8)	4.61	25.85 (0.24)	3.42(−10)	6.64	90.67 (0.55)
160×160	1.67(−5)	1.97	90.26 (0.227)	2.88(−9)	4.41	249.8 (0.89)	4.62(−12)	6.21	480.9 (1.90)
320×320	4.10(−6)	2.03	595.0 (0.503)	1.33(−10)	4.44	2760 (1.65)			
640×640	1.02(−6)	2.01	5174 (2.15)						

The numbers in the parenthesis are the CPU times in second for the generation of local difference schemes.

Case 6. As the last example, we illustrate further the proposed high order MIB method. In this, we design our scheme to be of fourth- and sixth-order convergence. We note that the present method is particularly favorable if the immersed inner boundary is convex. In such cases, there is always adequate number of grid points to support a given high order approximation of the interface conditions. Here we choose the immersed inner boundary as a circle of radius $\frac{1}{2}$ with its center located at the origin. The computational domain is $[-1,1]\times[-1,1]$, excluding the inner circle. The exact solution of the Poisson equation is set to be $u(x,y) = 5\exp(-x^2-y^2/2)$ on the computational domain. We plot the numerical solution, computed on a 40×40 grid, in Fig. 13 (right) with a negative sign (i.e., $-u(x,y)$) to illustrate the immersed inner boundary. The accuracy and convergence order of our MIB method are presented in Table 9. For high order method, it takes more time in the generation of the local difference schemes because of the iterative nature of current algorithm, as given in Table 9. However, it is still negligible compared to the time spent on the solution of the main linear system. It is seen that the result obtained by using the sixth order scheme at the grid of 40×40 is about 30 times more accurate than that obtained by using the second-order scheme at the grid of 640×640, while consuming only a fraction (1/473) of CPU time. In order for the second method to reach the same accuracy of the sixth-order scheme at the grid of 40×40, a grid larger than 2600×2600 has to been used, which approximately costs 300,000 s CPU time. Therefore, the sixth-order scheme is about 28,000 times more efficient than the second-order scheme in terms of CPU time. It is noted that the CPU time used for the generation of the local finite difference scheme (i.e, the immersed grids) is very small compared to that for solving the linear algebraic equation system.

5. Discussion

In this section, we give a comparison between the proposed MIB scheme and other well established methods, such as the immersed interface method (IIM) and the ghost fluid method (GFM), so as to have a better understanding of these methods.

First, all the aforementioned three interface methods require an explicit identification of the interface and associated irregular grid points. However, the choice of irregular points differs very much in three methods. An irregular point in the IIM might be a regular point in the MIB method, vice versa. In each method, irregular points are determined by the interface location and the selection of discretization schemes. In general, higher order schemes involve more irregular points.

Second, in the MIB approach, fictitious points (FPs) are created so that standard higher order central finite difference (FD) schemes can be directly employed throughout the domain, including the vicinity of the interface. When a FD scheme refers to grid point values from the other side of the interface, fictitious function values instead of real ones will be supplied. Due to the use of FPs, a uniform discretization scheme can be used over the whole domain. Nevertheless, one may choose different discretization schemes at any given location in the MIB, if it is desirable. Whereas, in the IIM and GFM, discretization schemes in the vicinity of the interface have to vary from point to point.

Third, in the MIB scheme, enforcing jump conditions is disassociated with the discretization of the governing PDE, while in the IIM, the jump conditions are satisfied in special discretization schemes along the interface. Therefore, in the MIB method, interface modeling is virtually independent of the governing PDE and the technique developed in solving one PDE can be applied to a vast range of interface problems without too much modification. In the GFM, it is only the key features of the jump conditions rather than original conditions themselves that are directly incorporated into the discretized Laplacian. For example, in higher dimensions, by means of a projection to Cartesian coordinate directions, the jump condition in the normal derivative is captured, while the less important tangential jump is omitted in the GFM. Such a modeling is in consistent with its overall first-order accuracy. The advantage of this modeling is that the resulting matrix is symmetric, while the sharp solution is still maintained at the interface. However, a change in philosophy is required in order to construct fully second-order and higher order GFMs.

Fourth, jump conditions are *repeatedly* enforced in the MIB method, while each jump condition is utilized only once in the IIM and GFM. Since jump conditions are reused in the MIB method, only the lowest order ones are required. Whereas, higher order jump conditions are inevitably required to construct a high order interface scheme, if each condition is applied only once.

Fifth, in the MIB method, higher order interface modeling is equivalent to the determination of FP values up to higher order via iteratively enforcing the jump conditions. Whereas, albeit the GFM uses ghost domains, it does not enforce all the jump conditions. Therefore, a change in philosophy is required for the GFM to achieve higher order accuracy. In the IIM, to achieve high order accuracy, entirely new discretization schemes that involve additional grid point(s) across the interface are to be constructed via incorporating jump conditions to the Taylor expansion. Thus, higher order interface modeling is equivalent to creating more jump conditions for higher order Taylor expansions. Such an extension to higher order is possible in 1D as explored in solving Maxwell's equations [52,53], but will be extremely difficult, if not impossible, for 2D or 3D interface problems for the following reasons: (i) Higher order jump conditions are usually quite complicated and involve high order derivatives. For example, the second order jump condition used in the IIM, which is derived from the PDE and low order jump conditions, typically involves $u^-, u^+, u_n^-, u_n^+, u_\tau^-, u_\tau^+, u_{nn}^-, u_{nn}^+, u_{\tau\tau}^-$, and $u_{\tau\tau}^+$. (ii) For a given order of accuracy, the discretization of a higher order derivative jump condition requires a larger stencil or involves more FPs, which in turn requires more higher order jump conditions. (iii) For many elliptic equations, such as the Helmholtz equation, cross derivatives will be involved, whose discretization requires wide stencils or many FPs in both the x- and y-directions. Consequently, the growth of the number of jump conditions can never match the growth of stencils or the number of FPs.

Sixth, for regular interfaces, the interface modeling can be carried out in a systematic and automatic manner in the MIB method. As a consequence, one MIB code will be able to generate schemes of different orders by simply adjusting a parameter. We can choose a desirable order of accuracy for a given problem under study. For irregular interfaces, though the MIB procedure proposed in the present work is systematic, the MIB, GFM and IIM all require subtle treatment of irregular grid points locally.

Seventh, due to the use of FPs and the separation of enforcing jump conditions from the discretization of the PDE, the MIB can be incorporated with other existing high-order methods for hyperbolic conservation laws, such as weighted essentially non oscillatory (WENO) schemes [34] and the conjugate oscillation reduction scheme (CFOR) [50,55]. We expect that such MIB based shock capturing methods would return sharp shock fronts.

Finally, it is noted that the IIM has been developed and significantly improved over the past decade by a number of researchers [4–6,10,16–19,21,23,24,30,31,36,42,43,46,47,51]. It is by far one of the most powerful existing methods for the solution of elliptic equations involving interfaces and/or singular sources. Using only one FPs on each side of the interface to capture the essence of the jump condition, the GFM is very simple and easy to realize for complex problems [12,35,48]. Adopting the FPs idea from the GFM and overcoming the difficulty of generating higher order schemes in the IIM, the proposed MIB can be regarded as a generalization of both the GFM and IIM.

6. Conclusion

In this work we have formulated novel high order schemes for solving elliptic equations with discontinuous diffusion coefficients and/or singular source terms, based on the matched interface and boundary (MIB) method originally proposed for time-domain electromagnetic wave scattering and propagation [52,53]. Two significant generalizations are made to the previous MIB method. First, the construction of high order MIB schemes was extended from Maxwell's equations to elliptic equations. Second, the formulation of the MIB scheme was extended from regular, straight interfaces to general, irregular interfaces, which endows the MIB method the applicability to a vast range of practical problems. The extension to curved interfaces was made possible by adding auxiliary line and fictitious points (FPs) such that one can strictly enforce physical jump conditions near the interface. Extensive numerical experiments are carried out to verify the high order nature of the proposed MIB scheme for both regular interfaces and curved interfaces. The MIB works well on the regular Cartesian grid and is robust regardless the magnitude of the jump discontinuity across the interface. In addition, the MIB does not require high order interface jump conditions as well as the high regularity of the interface. In general, the MIB method is more efficient and requires less CPU time at a given level of accuracy than other existing methods. Comparison is made to the immersed interface method (IIM) and the ghost fluid method (GFM) both numerically and theoretically. The main new features of the proposed MIB method are the follows: (1) the disassociation between enforcing jump conditions and discretizing PDEs via FPs; (2) the repeated use of the lowest order jump conditions via an iterative scheme. For solving elliptic equations, the proposed MIB method can be regarded as a generalization of the IIM and GFM. The generalization of the MIB method for treating Lipschitz continuous interfaces, for tracing moving interfaces, and for electromagnetic scattering are currently under our investigation.

Acknowledgements

This work was supported in part by NSF Grant IIS-0430987 and IRGP grant 71-4834. Zhou acknowledges Michigan State University for a Quantitative Biology Interdisciplinary Research Award.

References

[1] L. Adams, Z.L. Li, The immersed interface/multigrid methods for interface problems, SIAM J. Sci. Comput. 24 (2002) 463–479.
[2] L. Adams, T.P. Chartier, New geometric immersed interface multigrid solvers, SIAM J. Sci. Comput. 25 (2004) 1516–1533.
[3] I. Babuška, The finite element method for elliptic equations with discontinuous coefficients, Computing 5 (1970) 207–213.
[4] H. Ben Ameur, M. Burger, B. Hackl, Level set methods for geometric inverse problems in linear elasticity, Inverse Problems 20 (2004) 673–696.
[5] P.A. Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, J. Comput. Phys. 197 (2004) 364–386.
[6] G. Biros, L.X. Ying, D. Zorin, A fast solver for the Stokes equations with distributed forces in complex geometries, J. Comput. Phys. 193 (2004) 317–348.
[7] 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.
[8] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math. 79 (1998) 175–202.
[9] S.Z. Deng, K. Ito, Z.L. Li, Three-dimensional elliptic solvers for interface problems and applications, J. Comput. Phys. 184 (2003) 215–243.
[10] M.A. Dumett, J.P. Keener, An immersed interface method for solving anisotropic elliptic boundary value problems in three dimensions, SIAM J. Sci. Comput. 25 (2003) 348–367.
[11] 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 (2000) 30–60.
[12] 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.
[13] M. Feig, C.L. Brooks III, Recent advances in the development and application of implicit solvent models in biomolecule simulations, Curr. Opin. Struct. Bio. 14 (2004) 217–224.
[14] B. Fornberg, Calculation of weights in finite difference formulas, SIAM Rev. 40 (1998) 685–691.
[15] F. Gibou, R.P. 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 (2005) 577–601.
[16] S. Hou, X.-D. Liu, A numerical method for solving variable coefficient elliptic equation with interfaces, J. Comput. Phys. 202 (2005) 411–445.
[17] T.Y. Hou, Z.L. Li, S. Osher, H. Zhao, A hybrid method for moving interface problems with application to the Hele–Shaw flow, J. Comput. Phys. 134 (1997) 236–252.
[18] H. Huang, Z.L. Li, Convergence analysis of the immersed interface method, IMA J. Numer. Anal. 19 (1999) 583–608.
[19] J.K. Hunter, Z.L. Li, H. Zhao, Reactive autophobic spreading of drops, J. Comput. Phys. 183 (2002) 335–366.
[20] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, Appl. Mech. Rev. 56 (2003) 331–347.
[21] S. Jin, X.L. Wang, Robust numerical simulation of porosity evolution in chemical vapor infiltration II. Two-dimensional anisotropic fronts, J. Comput. Phys. 179 (2002) 557–577.
[22] H. Johansen, P. Colella, A Cartesian grid embedding boundary method for Poisson's equation on irregular domains, J. Comput. Phys. 147 (1998) 60–85.
[23] J.D. Kandilarov, Immersed interface method for a reaction-diffusion equation with a moving own concentrated source, Lecture Notes Comput. Sci. 2542 (2003) 506–513.
[24] L. Lee, R.J. LeVeque, An immersed interface method for incompressible Navier-Stokes equations, SIAM J. Sci. Comput. 25 (2003) 832–856.
[25] 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.
[26] Z.L. Li, A fast iterative algorithm for elliptic interface problems, SIAM J. Numer. Anal. 35 (1998) 230–254.
[27] Z.L. Li, The immersed interface method using a finite element formulation, Appl. Numer. Math. 27 (1998) 253–267.
[28] Z.L. Li, An overview of the immersed interface method and its applications, Taiwanese J. Math. 7 (2003) 1–49.
[29] Z.L. Li, K. Ito, Maximum principle preserving schemes for interface problems with discontinuous coefficients, SIAM J. Sci. Comput. 23 (2001) 339–361.
[30] Z.L. Li, S.R. Lubkin, Numerical analysis of interfacial two-dimensional Stokes flow with discontinuous viscosity and variable surface tension, Int. J. Numer. Meth. Fluid. 37 (2001) 525–540.
[31] Z.L. Li, W.-C. Wang, I.-L. Chern, M.-C. Lai, New formulations for interface problems in polar coordinates, SIAM J. Sci. Comput. 25 (2003) 224–245.
[32] S. Lim, C.S. Peskin, Simulations of the whirling instability by the immersed boundary method, SIAM J. Sci. Comput. 25 (2004) 2066–2083.

[33] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157–192.
[34] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[35] 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.
[36] B. Lombard, J. Piraux, How to incorporate the spring-mass conditions in finite-difference schemes, SIAM J. Sci. Comput. 24 (2003) 1379–1407.
[37] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numer. Anal. 21 (1984) 285–299.
[38] A. Mckenney, L. Greengard, A. Mayo, A fast Poisson solver for complex geometries, J. Comput. Phys. 118 (1995) 348–355.
[39] C.S. Peskin, Numerical analysis of blood flow in heart, J. Comput. Phys. 25 (1977) 220–252.
[40] C.S. Peskin, Lectures on mathematical aspects of physiology, Lect. Appl. Math. 19 (1981) 69–107.
[41] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran: the Art of Scientific Computing, Cambridge University Press, Cambridge, 1999.
[42] M. Schulz, G. Steinebach, Two-dimensional modelling of the river Rhine, J. Comput. Appl. Math. 145 (2002) 11–20.
[43] J.A. Sethian, A. Wiegmann, Structural boundary design via level set and immersed interface methods, J. Comput. Phys. 163 (2000) 489–528.
[44] A.K. Tornberg, B. Engquist, Numerical approximations of singular source terms in differential equations, J. Comput. Phys. 200 (2004) 462–488.
[45] Y.-H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593–623.
[46] J.V. Voorde, J. Vierendeels, E. Dick, Flow simulations in rotary volumetric pumps and compressors with the fictitious domain method, J. Comput. Appl. Math. 168 (2004) 491–499.
[47] J.H. Walther, G. Morgenthal, An immersed interface method for the vortex-in-cell algorithm, J. Turbulence 3 (039) (2002).
[48] W.-C. Wang, A jump condition capturing finite difference scheme for elliptic interface problems, SIAM J. Sci. Comput. 25 (2004) 1479–1496.
[49] G.W. Wei, Discrete singular convolution for the solution of the Fokker–Planck equations, J. Chem. Phys. 110 (1999) 8930–8942.
[50] G.W. Wei, Y. Gu, Conjugated filter approach for solving Burgers' equation, J. Comput. Appl. Math. 149 (2002) 439–456.
[51] A. Wiegmann, K.P. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (2000) 827–862.
[52] S. Zhao, G.W. Wei, High order FDTD methods via derivative matching for Maxwell's equations with material interfaces, J. Comput. Phys. 200 (2004) 60–103.
[53] S. Zhao, G.W. Wei, Tensor product derivative matching for wave propagation in inhomogeneous media, Microwave and Optical Technology Letters 43 (2004) 69–77.
[54] S. Zhao, G.W. Wei, Y. Xiang, DSC analysis of free-edged beams by an iteratively matched boundary method, J Sound Vibrat 284 (2005) 487–493.
[55] Y.C. Zhou, G.W. Wei, High-resolution conjugate filters for the simulation of flows, J. Comput. Phys. 189 (2003) 150–179.