MIB method for elliptic equations with multi-material interfaces

Kelin Xia\textsuperscript{a,b}, Meng Zhan\textsuperscript{a}, Guo-Wei Wei\textsuperscript{b,c,*}

\textsuperscript{a} Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Wuhan 430071, China
\textsuperscript{b} Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA
\textsuperscript{c} Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, USA

**A R T I C L E  I N F O**

Article history:
Received 22 October 2010
Received in revised form 15 February 2011
Accepted 22 February 2011
Available online 2 March 2011

Keywords:
Immersed boundary method
Immersed interface method
Ghost fluid method
Matched interface and boundary
Elliptic equations
Multiple material interfaces
Triple-junctions

**A B S T R A C T**

Elliptic partial differential equations (PDEs) are widely used to model real-world problems. Due to the heterogeneous characteristics of many naturally occurring materials and man-made structures, devices, and equipments, one frequently needs to solve elliptic PDEs with discontinuous coefficients and singular sources. The development of high-order elliptic interface schemes has been an active research field for decades. However, challenges remain in the construction of high-order schemes and particularly, for nonsmooth interfaces, i.e., interfaces with geometric singularities. The challenge of geometric singularities is amplified when they are originated from two or more material interfaces joining together or crossing each other. High-order methods for elliptic equations with multi-material interfaces have not been reported in the literature to our knowledge. The present work develops matched interface and boundary (MIB) method based schemes for solving two-dimensional (2D) elliptic PDEs with geometric singularities of multi-material interfaces. A number of new MIB schemes are constructed to account for all possible topological variations due to two-material interfaces. The geometric singularities of three-material interfaces are significantly more difficult to handle. Three new MIB schemes are designed to handle a variety of geometric situations and topological variations, although not all of them. The performance of the proposed new MIB schemes is validated by numerical experiments with a wide range of coefficient contrasts, geometric singularities, and solution types. Extensive numerical studies confirm the designed second order accuracy of the MIB method for multi-material interfaces, including a case where the derivative of the solution diverges.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Mathematical modeling of material interfaces often leads to elliptic partial differential equations (PDEs) with discontinuous coefficients and singular sources. Solutions to such PDEs have attracted much attention\cite{2,4,5,11,14,19,27,25,28,29,31,32,34,37,39,40,42,45,56–59,50,62,6,3} since Peskin pioneered the immersed boundary method (IBM) in 1977\cite{20,35,53,54,52}. Interface techniques have demonstrated an increasing importance in a number of areas, including fluid dynamics\cite{12,15,16,30,49,37,36}, electromagnetic wave propagation\cite{22,23,33,34,68,67}, materials science,\cite{24,27} and biological systems\cite{43,70,64,17,7}. A major advance in the field after Peskin’s contribution was the remarkable second order sharp interface scheme, the immersed interface method (IIM) constructed by LeVeque and Li\cite{38,1,10,41}. Many other elegant methods have been proposed in the past decade, including the ghost fluid method (GFM) proposed by Fedkiw, Osher...
Elliptic interface methods were originally developed for or motivated by problems in computational fluid dynamics. In fact, time-dependent problems, matrix properties are still important [68, 69].

The reader is referred to the portable, extensible toolkit for scientific computation (PETSc) package (http://www.mcs.anl.gov/petsc/petsc-as/) for the application of advanced linear algebraic equation solvers. However, for Krylov subspace techniques. The non-symmetric and non-diagonally dominant matrices resulting from the discretization of the governing PDE. In fact, the interface modeling is virtually independent of the governing PDE and thus the technique developed for solving one PDE can be applied to a vast range of other PDEs without too much modification. A comparison of the MIB, GFM, and IIM is given in our earlier work [72, 71].

Most of earlier elliptic interface methods are developed for smooth interfaces, perhaps partially due to the fact that many elliptic interface methods were originally developed for or motivated by problems in computational fluid dynamics. In fact, in most other physical problems, one frequently encounters elliptic problems with nonsmooth interfaces or interfaces with Lipschitz continuity [9, 25, 26, 65, 64]. Nonsmooth interfaces are also called geometric singularities. This class of problems is significantly more challenging than ones with smooth interfaces. The first known second order accurate schemes for 2D nonsmooth interfaces were constructed by us using the MIB method in 2007 [66]. Since then, a couple of other interesting methods have been reported for this class of problems in 2D [9, 26]. It is worth mentioning that the PPD developed by Chen and Strain has demonstrated its utility in handling exotically complex interface topology [9]. It will be interesting to see the further development of the PPD for 3D elliptic interface problems. For arbitrarily nonsmooth interfaces in 2D, the best results reported are all limited to the second order accuracy [9, 26, 66]. The construction of 2D schemes of numerical orders higher than two for arbitrarily nonsmooth interfaces is another open problem, although it may be possible to do so for some special nonsmooth interface geometries. For practical applications, there is a pressing need to develop high order schemes for non-smooth interfaces in 3D domains as most realistic problems are in 3D domains. The MIB method [65] provides second order schemes for 3D elliptic PDEs with arbitrarily non-smooth interfaces or geometric singularities. It has found numerous applications in biomolecular systems [64, 17, 18] and biomedical imaging [8]. However, currently fourth order schemes for 3D elliptic PDEs with nonsmooth interfaces have been developed only for a few special interface geometries [65]. It appears truly challenging to develop fourth order schemes for arbitrarily non-smooth interfaces in 3D domains, which contributes to another open problem in the field as posed in 2007 [65].

Moreover, it is fairly easy to construct a second order scheme for elliptic interface problems and demonstrate the scheme on some special and simple geometries. However, it is extremely difficult to ensure that the designed accuracy is achievable for all possible geometric situations. So far, there has been very little literature to address this issue in the computational mathematics community. Unfortunately, for real-world problems, one encounters extremely complicated geometries. Consequently, most mathematical schemes do not keep their designed promises. This is particularly true for biomolecular systems – molecular surfaces [55] of proteins have geometric singularities, i.e., cusps, sharp tips and self-intersecting surfaces. The MIB method based Poisson–Boltzmann solver, the MIBPB [64, 17, 18], is able to derive second order accuracy in solving the Poisson–Boltzmann equation with primitive interface surface of proteins. Unfortunately, at present, the MIBPB still cannot guarantee its designed second order accuracy for tens of thousands of protein geometries – it occasionally encounters...
the accuracy reduction for some special protein molecular surfaces. Therefore, there is a pressing need in the field of elliptic interface problems to develop systematical procedures to ensure the designed accuracy for all possible geometric and topological variations.

What concerns the present work is a new class of interface problems, the construction of high order numerical schemes for elliptic equations with discontinuous coefficients from modeling multi-material interfaces, as briefly reported elsewhere [63]. This class of problems is referred to as multi-material interace problems and is omnipresent in science, engineering and daily life. The solution to this class of problems becomes exceptionally challenging when more than two heterogeneous materials join at one point of the space and form a geometric singularity. For instant, two different internal organs and the internal liquid can form a three-material singularity (or triple junction) [8]. Note that although each organ is very smooth, the resulting geometric singularity can be very sharp. To our knowledge, no second order elliptic interface scheme has been constructed for this class of problems before our poster in 2010 [63]. The primary objective of the present work is to construct a second order method to solve 2D elliptic equations with discontinuous coefficients associated with three-material interfaces. The MIB method is developed for solving this class of challenging problems. The secondary objective of the present work is to removed an acute angle restriction in our earlier MIB schemes for 2D elliptic PDEs with two-material interfaces [66] by utilizing two sets of interface jump conditions. Additionally, we propose new MIB schemes to deal with singular interface geometries due to exotic two-material interface topologies.

The rest of this paper is organized as follows. Section 2 is devoted to the theory and algorithm. We start with the construction of a number of new MIB algorithms for two-material nonsmooth interfaces. These algorithms resolve the restriction of the minimum angle in our previous 2D MIB schemes [66] by using interface jump conditions at two intersecting points between the interface and a mesh line. This idea, originally proposed in our 3D scheme [65] is very useful for us to construct the present MIB scheme for multi-material interfaces. We also provide a full consideration of all the possible geometric and topological configurations for an interface to locate exactly on a grid node in two-material nonsmooth interfaces, which completes our previous 2D MIB schemes for two-material nonsmooth interfaces and provides a systematical strategy for the present MIB scheme in dealing with multi-material interfaces. Algorithms for three-material interfaces are presented in Section 3. As indicated before, the main challenges of three-material interfaces still come from the geometric singularities when three-materials join at one point in the space. Otherwise, the three-material problems can be handled by previous two-material schemes. We categorize the three-material interface singularities into three types of distinguished topologies and appropriate MIB schemes are constructed for these topological variations. In Section 4, we carry out intensive numerical experiment to validate the order of accuracy and demonstrate the performance of the proposed MIB schemes for two- and three-material nonsmooth interfaces. A variety of geometric morphologies are considered in the present work. Problems with a wide range of coefficient magnitudes are designed to test the robustness of the MIB method. The designed second order accuracy is confirmed in our numerical studies. This paper ends with a conclusion.

2. Theory and algorithm for two-material interfaces

In this section, we develop new systematical schemes to deal with all the possible geometric and topological configurations in two-material nonsmooth interfaces. These schemes also provide some of the basic technical preparation for us to handle three-material interfaces in Section 3.

Consider an open bounded domain $\Omega \subseteq \mathbb{R}^2$. A given interface $\Gamma$ divides $\Omega$ into two subdomains, $\Omega^a$ and $\Omega^b$, hence $\Omega = \Omega^a \cup \Omega^b$ and $\Gamma = \partial \Omega^a \cap \partial \Omega^b$. We assumed that the boundary $\partial \Omega$ and interfaces $\Gamma$ are Lipschitz continuous and there is a piecewise smooth level-set function $\phi$ on $\Omega$, such that $\Gamma = \{x,y : \phi = 0; x,y \in \Omega\}$. $\Omega^a = \{x,y : \phi \geq 0; x,y \in \Omega\}$ and $\Omega^b = \{x,y : \phi \leq 0; x,y \in \Omega\}$. We solve the following two dimensional (2D) elliptic interface problem

\begin{equation}
\begin{aligned}
&\beta(x,y) u_x(x,y) + (\beta(x,y) u_y(x,y))_y = g(x,y), \\
&\text{with variable coefficients } \beta(x,y), \text{ which may have jumps at the interface } \Gamma.
\end{aligned}
\end{equation}

To make the problem well defined, there are two given jump conditions associated with the interface, i.e.,

\begin{equation}
|u| = u^a - u^b = \Psi_1,
\end{equation}

\begin{equation}
|\beta u_n| = \beta^a u^a_n - \beta^b u^b_n = \Psi_2,
\end{equation}

where superscripts $a$ and $b$ denote the limiting value of a function from $\Omega^a$ and $\Omega^b$ sides of the interface $\Gamma$ respectively. We assume that both $\Psi_1(x,y)$ and $\Psi_2(x,y)$ are $C^1$ continuous. In considering the interface which is not always aligned with the $x$- or $y$-mesh lines, one more interface condition can be attained by differentiating Eq. (2) along the tangential direction of the interface, $|u| = u^a - u^b$. Hence for a point on the interface, if we define unit normal vector $\hat{n} = (\cos \theta, \sin \theta)$, the unit tangential vector is $\hat{t} = (-\sin \theta, \cos \theta)$. Thus, we have three jump conditions

\begin{equation}
|u| = u^a - u^b,
\end{equation}

\begin{equation}
|u| = (-u^a_s \sin \theta + u^a_t \cos \theta) - (-u^b_s \sin \theta + u^b_t \cos \theta),
\end{equation}

\begin{equation}
|\beta u_n| = \beta^a (u^a_s \cos \theta + u^a_t \sin \theta) - \beta^b (u^b_s \cos \theta + u^b_t \sin \theta).
\end{equation}
In the MIB approach, the implementation of jump conditions is disassociated from the discretization of the elliptic equation. Here, \( \beta u_o \) and \( \beta u_j \) are discretized separately with the second order central finite difference scheme \([71]\). Therefore, a direct calculation of \( \beta u_o \) at point \((i,j)\) only involves grid points \( u_{i-1,j}, u_{i,j} \) and \( u_{i+1,j} \). When the interface intersects the \( j \)th mesh line at a point between \((i,j)\) and \((i+1,j)\), the \( u_i \) and \( u_{i+1} \) are located in different subdomains. In order to avoid the reduction of the convergence order, we replace \( u_{i+1,j} \) with a fictitious value \( f_{i+1,j} \), which can be regarded as a smooth extension of the function value from \( u_{i+1,j} \) side. To estimate the fictitious value, we enforce the jump conditions by discretizing some of \( u_j \), \( u_p \), \( u_j \), \( u_k \), and \( u_j \) based on the geometry. We can then construct two linear equations involving two fictitious values.

It is always possible in practice to find two fictitious values in a 2D domain. If \( u_{i+1} \) is difficult to evaluate, we can eliminate it from Eqs. (5) and (6) to attain

\[
[u] = u^e - u^b, \quad \text{and} \quad [\beta u_o - \beta^a \tan \theta | u_i |] = C^x u^x - C^b u^b + C^y u^y,
\]

where \( C^x = \beta^a \cos \theta + \beta^b \tan \theta \sin \theta \). Here, \( C^b = \beta^a \cos \theta + \beta^b \tan \theta \sin \theta \) and \( C^y = \beta^a \sin \theta - \beta^b \sin \theta \). Based on the geometry, we can choose to eliminate one of \( u^x \), \( u^y \), and \( u^z \) and attain two linear equations. If one eliminates \( u^x \),

\[
[u] = u^e - u^b, \quad \text{and} \quad [\beta u_o - \beta^a \tan \theta | u_i |] = C^b u^b - C^y u^y,
\]

where \( C^b = \beta^a \cos \theta + \beta^b \tan \theta \sin \theta \), \( C^y = \beta^a \cos \theta + \beta^b \tan \theta \sin \theta \) and \( C^y = \beta^a \sin \theta - \beta^b \sin \theta \). If one eliminates \( u^y \),

\[
[u] = u^e - u^b, \quad \text{and} \quad [\beta u_o + \beta^a \cot \theta | u_i |] = C^x u^x + C^b u^b - C^y u^y,
\]

where \( C^x = \beta^a \cos \theta + \beta^b \cot \theta \sin \theta \), \( C^y = \beta^a \cos \theta + \beta^b \sin \theta \) and \( C^y = \beta^a \sin \theta - \beta^b \sin \theta \). If one eliminates \( u^x \),

\[
[u] = u^e - u^b, \quad \text{and} \quad [\beta u_o + \beta^a \cot \theta | u_i |] = C^x u^x + C^b u^b - C^y u^y,
\]

where \( C^x = \beta^a \cos \theta + \beta^b \cot \theta \sin \theta \), \( C^y = \beta^a \cos \theta + \beta^b \sin \theta \) and \( C^y = \beta^a \sin \theta - \beta^b \sin \theta \).

In our earlier work, we classified irregular points as off-interface type and on-interface type, and proposed different schemes to deal with them \([66]\). For Off-interface schemes 2, which is designed to do with the situation when interface has large curvature, sharp edge, sharp wedge and tip, we introduced the concept of secondary fictitious point and achieved second order accuracy for nearly all the situations except when the acute angle is smaller than the critical angle of \( 2 \tan^{-1}(1/3) \) \([66]\). Here, we present a scheme to remove this limitation in 2D domain. The resulting MIB algorithm has no critical angle restriction at all. The detail of this scheme is described in Section 2.1.

On-interface schemes are created for the situations where the interface intersects a node of the Cartesian grid \([66]\). We introduced some basic schemes to study general on-interface situations and achieved second order accuracy. While, as the geometry becomes more and more complicated and some unusual on-interface situations may occur. In order to make our MIB method general and robust, we category all the possible on-interface situations from a topological point of view. We propose three new on-interface schemes. By combining our disassociation technique \([66]\) with six on-interface schemes, we are able to provide solutions for all possible on-interface situations.

### 2.1. Scheme for two-material interfaces

The MIB approach splits a 2D problem into 1D ones, and disassociates the enforcement of jump conditions from the discretization of the PDE under study. At an interface, such a disassociation is made possible by appropriate use of the auxiliary points and fictitious points. For example, when \( u_i \) and \( u_{i+1} \) are in different regions, two fictitious values \( f_{i,j} \) and \( f_{i+1,j} \) and two auxiliary values \( u_{i-1} \) and \( u_{i+2} \) are needed to construct the central difference scheme at point \((i,j)\), which involves values \( u_{i-1}, u_{i+1}, u_{i+1,j} \). Similarly, at point \((i+1,j)\), \( f_{i,j} \), \( u_{i+1} \), and \( u_{i+2} \) are involved. However, when there are sharp tips or sharp wedges at the interface, the auxiliary values are not available. This situation is already considered by us for second order 3D MIB schemes \([65]\). Here, we give a detailed description to resolve this difficulty in the present 2D scheme, and thus remove the critical angle restriction in our earlier MIB scheme in 2D \([66]\).

Consider the situation depicted in Fig. 1, two interfaces near irregular point \((i,j)\) intersect the grid meshline at points \( o \) and \( o' \). In a common MIB approach, the enforcement of the jump conditions at point \( o \) to solve the fictitious values \( f_{i-1,j} \) and \( f_{i,j} \) requires that the auxiliary points \( u_{i+1} \) and \( u_j \) are in the same domain. In order to overcome this geometric restriction, we replace \( u_{i+1} \) with fictitious value \( f_{i+1,j} \). Therefore, together with the original two fictitious values, we can discretize jump condition (2) at points \( o \) and \( o' \), and two sets of equations can be attained. As we now have three fictitious values to be resolved, we just need one more equation. This equation, however, can be easily attained by using the normal and tangential direction jump conditions at point \( o \) or \( o' \). The detail of this scheme is described below.

In Fig. 1, the interface divides the whole domain into two subdomains, denoted as \( \Omega^o \) and \( \Omega^{o'} \). In this paper, \( \Omega^o \) is marked with yellow color, while \( \Omega^{o'} \) is marked with green color. To make it clear, we denote \( u_{o'} \), the limiting value of function \( u \) at point \( o \) of \( \Omega^o \) and \( u_{o'}^{\otimes} \), the limiting value of function \( u \) at point \( o \) from \( \Omega^{o'} \). The derivative of \( u \) with respect to \( x \) at point \( o \) from \( \Omega^o \) is represented as \( u_{o,x}^o \), and from domain \( b \) as \( u_{o,x}^{\otimes} \). We use a set of similar notations for point \( o' \). For example, \( u_{o'}^{\otimes} \) stands for the limiting value of function \( u \) from \( \Omega^{o'} \) at point \( o' \).

Therefore, for point \( o \), using \( u_{i-1,j}, u_{i-1,j}, u_{i+1,j} \) and the fictitious values \( f_{i-1,j} \), \( f_{i,j} \) and \( f_{i+1,j} \), we can discretize \( u_{o,x}^o, u_{o,x}^{\otimes}, u_{o,x}^{\otimes}, u_{o,x}^{\otimes} \) explicitly as follows...
Scheme for two-material interfaces. In this figure as well as all other figures \( \Omega^o \) is marked with yellow color and \( \Omega^b \) is denoted by green color. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

\[
\begin{align*}
\mathbf{u}^0 &= \begin{pmatrix}
\mathbf{w}_{C0}^0_{i-1,j} & \mathbf{w}_{C0}^0_{i,j} & \mathbf{w}_{C0}^0_{i+1,j}
\end{pmatrix}^T, \\
\mathbf{u}^b &= \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-1,j} & \mathbf{w}_{C0}^b_{i,j} & \mathbf{w}_{C0}^b_{i+1,j}
\end{pmatrix}^T, \\
\mathbf{u}_x^0 &= \begin{pmatrix}
\mathbf{w}_{C0}^0_{i-1,j} & \mathbf{w}_{C0}^0_{i,j} & \mathbf{w}_{C0}^0_{i+1,j}
\end{pmatrix}^T, \\
\mathbf{u}_x^b &= \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-1,j} & \mathbf{w}_{C0}^b_{i,j} & \mathbf{w}_{C0}^b_{i+1,j}
\end{pmatrix}^T, \\
\mathbf{u}_y^0 &= \begin{pmatrix}
\mathbf{w}_{C0}^0_{i-1,j} & \mathbf{w}_{C0}^0_{i,j} & \mathbf{w}_{C0}^0_{i+1,j}
\end{pmatrix}^T.
\end{align*}
\]

where \( w_{m,n} \) are the standard finite difference weights computed from Lagrange polynomials. Subscript \( n = 0 \) represents the interpolation and subscript \( n = 1 \) the first order derivative. Here, subscript \( m \) stands for the node index and superscript \( l \) is the position of the interpolation, for example, \( a_0 \) stands for the limiting value at point \( o \) from \( \Omega^o \).

According to Fig. 1, it is difficult for us to find suitable auxiliary points to discretize \( u^0_n \) as there are not enough points in \( \Omega^o \) around point \( o \). We choose to discretize \( u^b_n \). In order to do so, we need three \( u \) values along the \( y \)-direction in \( \Omega^b \). Here we denote the coordinate of point \( o \) as \((i,j)\). We can choose these three values at point \((i,j)\), \((i,j+1)\) and \((i,j+2)\). Having already obtained \( u^b_o \), the other two values can be approximated by auxiliary values \( u_{i-2,j} \), \( u_{i-2,j+1} \), \( u_{i-1,j+1} \), \( u_{i-1,j+2} \), \( u_{i-2,j+2} \), \( u_{i-1,j+2} \) and \( u_{i,j+2} \). Here, \( u^b_y \) can be represented as

\[
\begin{align*}
\mathbf{u}^b_y &= \begin{pmatrix}
\mathbf{w}_{C0}^b_{i,j} & \mathbf{w}_{C0}^b_{i-1,j} & \mathbf{w}_{C0}^b_{i+1,j} & \mathbf{w}_{C0}^b_{i-1,j+1} & \mathbf{w}_{C0}^b_{i+1,j+1} & \mathbf{w}_{C0}^b_{i-1,j+2} & \mathbf{w}_{C0}^b_{i+1,j+2} & \mathbf{w}_{C0}^b_{i-1,j+3} & \mathbf{w}_{C0}^b_{i+1,j+3}
\end{pmatrix}^T, \\
\end{align*}
\]

where, \( w_1 \) and \( w_2 \) are finite difference coefficients for the points at two auxiliary lines. Based on the local geometry, we eliminate \( u^b_\alpha \), and two equations can be attained by substituting Eqs. (11) and (12) into Eq. (8)

\[
\begin{align*}
[u]_o &= \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-1,j} & \mathbf{w}_{C0}^b_{i,j} & \mathbf{w}_{C0}^b_{i+1,j}
\end{pmatrix}^T \cdot \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-1,j} & \mathbf{w}_{C0}^b_{i,j} & \mathbf{w}_{C0}^b_{i+1,j}
\end{pmatrix}^T - \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-2,j} & \mathbf{w}_{C0}^b_{i-1,j+1} & \mathbf{w}_{C0}^b_{i+1,j+1}
\end{pmatrix}^T,
\end{align*}
\]

where, \( w_1 \) and \( w_2 \) are finite difference coefficients for the points at two auxiliary lines. Based on the local geometry, we eliminate \( u^b_\alpha \), and two equations can be attained by substituting Eqs. (11) and (12) into Eq. (8)

\[
\begin{align*}
\beta [u]_o - [u]_o - \beta^2 \tan \theta [u]_o &= C_{\alpha}^b \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-1,j} & \mathbf{w}_{C0}^b_{i,j} & \mathbf{w}_{C0}^b_{i+1,j}
\end{pmatrix}^T \cdot \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-1,j} & \mathbf{w}_{C0}^b_{i,j} & \mathbf{w}_{C0}^b_{i+1,j}
\end{pmatrix}^T - \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-2,j} & \mathbf{w}_{C0}^b_{i-1,j+1} & \mathbf{w}_{C0}^b_{i+1,j+1}
\end{pmatrix}^T,
\end{align*}
\]

where, \( w_1 \) and \( w_2 \) are finite difference coefficients for the points at two auxiliary lines. Based on the local geometry, we eliminate \( u^b_\alpha \), and two equations can be attained by substituting Eqs. (11) and (12) into Eq. (8)

\[
\begin{align*}
\beta [u]_o - \beta^2 \tan \theta [u]_o &= C_{\alpha}^b \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-1,j} & \mathbf{w}_{C0}^b_{i,j} & \mathbf{w}_{C0}^b_{i+1,j}
\end{pmatrix}^T \cdot \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-1,j} & \mathbf{w}_{C0}^b_{i,j} & \mathbf{w}_{C0}^b_{i+1,j}
\end{pmatrix}^T - \begin{pmatrix}
\mathbf{w}_{C0}^b_{i-2,j} & \mathbf{w}_{C0}^b_{i-1,j+1} & \mathbf{w}_{C0}^b_{i+1,j+1}
\end{pmatrix}^T,
\end{align*}
\]
Here the unit normal vector is \( \bar{n} = (\cos \theta, \sin \theta) \). \( C_x = \beta^c \cos \theta + \beta^a \tan \theta \sin \theta \) and \( C_x' = \beta^b \sin \theta - \beta^b \sin \theta \). Therefore, we obtain two equations from the interface information at point \( o \). We can use the same method to attain two similar equations by using jump conditions at point \( o' \). There are only three fictitious values which are to be represented by node values and jump conditions. Therefore, we just need to discretize \( u_{ao}^o \) and \( u_{ao}^{o'} \), and enforce Eq. (4) as follows

\[
\begin{align*}
\begin{array}{c}
u_{ao}^o = \begin{pmatrix}
W_{01}^{ao} & W_{02}^{ao} & W_{03}^{ao}
\end{pmatrix} \cdot (f_{i-1,j}, u_{i,j}, f_{i+1,j})^T, \\
u_{ao}^{o'} = \begin{pmatrix}
W_{01}^{ao} & W_{02}^{ao} & W_{03}^{ao}
\end{pmatrix} \cdot (u_{i-1,j}, f_{i,j}, u_{i+1,j})^T.
\end{array}
\end{align*}
\]

(16)

Substituting Eq. (16) into jump condition (4), one has

\[
[u]_o = (W_{01}^{ao}, W_{02}^{ao}, W_{03}^{ao}) \cdot (f_{i-1,j}, u_{i,j}, f_{i+1,j})^T - (W_{01}^{ao}, W_{02}^{ao}, W_{03}^{ao}) \cdot (u_{i-1,j}, f_{i,j}, u_{i+1,j})^T.
\]

(17)

To obtain fictitious values, we solve Eqs. (13), (14) and (17) together. The fictitious values can be represented by node values and jump conditions from interface intersecting points \( o \) and \( o' \)

\[
\begin{pmatrix}
f_{i-1,j} \\
f_{i,j} \\
f_{i+1,j}
\end{pmatrix} = \begin{pmatrix}
C_1 & \cdots & C_{14}
\end{pmatrix} \cdot \begin{pmatrix}
U_{14x1}
\end{pmatrix}.
\]

(18)

where vector \( \{U\}_{14x1} \) consists of 10 function values and 4 jump conditions as follows

\[
\begin{pmatrix}
U_{14x1}
\end{pmatrix} = \begin{pmatrix}
u_{i-2,j}, u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, u_{i-1,j+1}, u_{i,j+1}, u_{i+1,j+1}, [u]_o, [u]_{o'}, [\beta u]_o, [u]_{o'}
\end{pmatrix}^T.
\]

(19)

Here, \( \{C\}_{3x14} \) is a coefficient matrix and its components are the combination of weights and trigonometric functions of the normal vectors at points \( o \) and \( o' \). Thus, once the locations of \( o \) and \( o' \) are given, one can easily calculate the matrix and then the expression of fictitious values. Finally, one discretizes \( \beta u \) at irregular points \( (i-1,j), (i,j) \) and \( (i+1,j) \) as follows

\[
\begin{align*}
(\beta u)_{i,j} = \frac{1}{\Delta x^2} & \left( \beta^b \frac{1}{2} \frac{f_{i,j}}{f_{i,j}} - \beta^b \frac{1}{2} \frac{f_{i,j}}{f_{i,j}} \right) \cdot (u_{i,j}, u_{i,j})^T, \\
(\beta u)_{i,j} = \frac{1}{\Delta x^2} & \left( \beta^b \frac{1}{2} \frac{f_{i,j}}{f_{i,j}} - \beta^b \frac{1}{2} \frac{f_{i,j}}{f_{i,j}} \right) \cdot (f_{i-1,j}, u_{i,j})^T, \\
(\beta u)_{i,j} = \frac{1}{\Delta x^2} & \left( \beta^b \frac{1}{2} \frac{f_{i,j}}{f_{i,j}} - \beta^b \frac{1}{2} \frac{f_{i,j}}{f_{i,j}} \right) \cdot (f_{i,j}, u_{i,j})^T.
\end{align*}
\]

(20)

To solve the singularity problem of sharp tips and sharp wedges, we enforce jump conditions at two interface intersecting points together. It is seen that not all the information of two jump conditions is required in the present scheme. In fact, at point \( o' \), we only enforce Eq. (4) by discretizing \( u_{ao}^o \) and \( u_{ao}^{o'} \). We can however discretize \( u_{ao}^{o'} \) and \( u_{ao}^{o'} \), and find suitable auxiliary points to approximate \( u_{ao}^{o'} \) or \( u_{ao}^{o'} \) based on the geometry. Therefore, we can construct another equation by enforcing Eq. (8). In fact, this is the main idea to deal with geometric singularities in three-material interfaces.

2.2. On-interface schemes

In our previous work, the situation of an interface passing through a grid node was studied and three “On-interface schemes” were constructed for dealing with some of these “on-interface” situations [66]. The main idea in these schemes is that when the interface intersects with a grid node, this special node can be treated as serving dual functions, one as a node inside the interface and the other as a node outside the interface. State differently, the special node can have a left-limiting value and a right-limiting value. These two values are related by the jump condition (4). We then utilize both values on this special node and other auxiliary points to approximate \( u_{ao}^{o'} \) or \( u_{ao}^{o'} \) based on the geometry. Therefore, we can construct another equation by enforcing Eq. (8).

Although many situations are considered in our earlier three on-interface schemes [66], there are still some exotic situations that have not been accounted for. As the interface geometry becomes more and more sophisticated, more on-interface schemes are needed to ensure the robustness of the MIB method. In the present work, we summarize all the possible topological situations of the interface intersecting with a grid node and construct corresponding schemes to resolve these situations.

Three on-interface schemes were discussed in our earlier work [66] and their topologies are depicted in Fig. 2. The red dots and green dots indicate grid points in different domains separated by the interface. To obtain fictitious values, we focus on two jump equations

\[
[u]_z = (\sin \theta, -\cos \theta, \sin \theta, -\cos \theta) \cdot \begin{pmatrix}
u_x, v_x, u_y, u_y
\end{pmatrix}^T,
\]

(21)

\[
[\beta u]_z = (\beta^b \cos \theta, \beta^b \sin \theta, -\beta^b \cos \theta, -\beta^b \sin \theta) \cdot \begin{pmatrix}
u_x, v_x, u_y, u_y
\end{pmatrix}^T.
\]

(22)
For node point on the interface, if we regard it as in \( X_a \), we can use the jump condition \[ u \] = \( u_a / C_0 \) \( u_b \) to obtain \( u_b \). Therefore we can have two limiting values at this grid point. Combining them with the function value from auxiliary points, we can discretize \( u_a \) \( x \); \( u_b \) \( x \); \( u_a \) \( y \); and \( u_b \) \( y \) and substitute them into two jump conditions (21) and (22) to solve two fictitious values. Otherwise we can just eliminate one of the four derivatives by using the above two equations and substitute the discretization expressions of other three derivatives into the equation we attained to solve one fictitious value. The second approach is similar to the common method of calculating the fictitious value in our earlier MIB, except that the derivative is calculated on a grid node rather than off the grid. Thus, the discretization is much simpler.

2.2.1. On-interface scheme 4

As shown in Fig. 3 (On-interface scheme 4), the interface passes through node \( (i, j) \). All the green dots are on the same side of the interface and all the red dots are on the other side of the interface. For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$
\begin{align*}
\text{(On-interface scheme 1)} & & \text{(On-interface scheme 2)} & & \text{(On-interface scheme 3)} \\
\text{(On-interface scheme 4)} & & \text{(On-interface scheme 5)} & & \text{(On-interface scheme 6)}
\end{align*}
$$

For node point on the interface, if we regard it as in \( Q^a \), we can use the jump condition \[ |u| = u^a - u^b \] to obtain \( u^b \). Therefore we can have two limiting values at this grid point. Combining them with the function value from auxiliary points, we can discretize \( u^a \); \( u^b \); \( u^c \); and \( u^d \) and substitute them into two jump conditions (21) and (22) to solve two fictitious values. Otherwise we can just eliminate one of the four derivatives by using the above two equations and substitute the discretization expressions of other three derivatives into the equation we attained to solve one fictitious value. The second approach is similar to the common method of calculating the fictitious value in our earlier MIB, except that the derivative is calculated on a grid node rather than off the grid. Thus, the discretization is much simpler.

2.2.1. On-interface scheme 4

As shown in Fig. 3 (On-interface scheme 4), the interface passes through node \( (i, j) \). All the green dots are on the same side of the interface and all the red dots are on the other side of the interface. For this scheme and all the schemes below, we assume that all the green points belong to \( \Omega^b \), and all the red points belong to \( \Omega^a \). Here for the situation in Fig. 3 (On-interface scheme 4), the on-interface grid node is treated as a \( \Omega^b \) point. With the relation \( u_{ij} = u_{ij}^b + |u| \), we can discretize three derivatives as

$$
\begin{align*}
\bar{u}_x &= (w_{1,i-1}, w_i, w_{1,i+1}) \cdot (u_{i-1,j}, u_{ij}, u_{i+1,j})^T, \\
\bar{u}_y &= (w_{1,j-1}, w_{1,j}, w_{1,j+1}) \cdot (u_{ij-1}, u_{ij}, u_{ij+1})^T, \\
\bar{u}_f &= (w_{ij-1}, w_{ij}, w_{ij+1}) \cdot (u_{ij-1}, u_{ij}, f_{ij+1})^T.
\end{align*}
$$

Substituting above relations into Eq. (10), we can easily calculate \( f_{ij+1} \).

2.2.2. On-interface scheme 5

The geometry is shown in Fig. 3 (On-interface scheme 5). We treat the on-interface node as a point in \( \Omega^b \), and note that \( u_{ij} = u_{ij}^b + |u| \). The two fictitious values are \( f_{i-1,j} \) and \( f_{ij+1} \). Four derivatives can be discretized as
By substituting these equations into Eqs. (21) and (22), we solve \( f_{i-1,j} \) and \( f_{i,j+1} \).

### 2.2.3. On-interface scheme 6

As shown in Fig. 3 (On-interface scheme 6), we treat the on-interface node as a point in \( X_a \), and use the relation

\[
ub^x_i = \frac{w_{1,i-1} + w_{1,j} + [u_i, u_{i+2,j}]}{C_1}
\]

Two fictitious values are \( f_{i-1,j} \) and \( f_{i,j+1} \). We obtain the explicit discretization equations of four derivatives as

\[
ua^x = \left( w_{1,i} + w_{1,j} + [u_i, u_{i+2,j}] \right)^T,
\]

\[
ub^x = \left( w_{0,i} + w_{0,j} + [u_i, u_{i+2,j}] \right)^T.
\]
\[
\mathbf{u}_y^w = (w_{1,j-1}, w_{1,j}, w_{1,j+1}) \cdot (u_{i,j-1}, u_{i,j}, f_{ij-1})^T, \\
\mathbf{u}_y^b = (w_{1,j-2}, w_{1,j}, w_{1,j+1}) \cdot (u_{i,j-2}, u_{i,j} - |t|, u_{i,j+1})^T.
\]

We compute \(f_{i-1,j}\) and \(f_{i,j+1}\) by substituting above equations into Eqs. (21) and (22).

### 2.2.4. On-interface topological variations

An important issue in the construction of elliptic interface schemes is to make sure that the designed scheme is robust for all possible geometric situations and topological variations. As 2D schemes provide some of the necessary cornerstones for 3D schemes, it is important to systematically examine all possible topological variations. Here, we develop MIB schemes for two-material interfaces with all possible 2D on-interface topological variations. For each scheme, if we denote points in \(\Omega^a\) as inside the interface and in \(\Omega^b\) as outside the interface, the basic topological relation remains the same when we switch all the points inside and outside. Similarly, the basic topological relation remains the same if we rotate the axis clockwisely by \(\frac{1}{4}\pi\), \(\pi\) or \(\frac{3}{4}\pi\). Therefore, our MIB scheme for this particular topology should work for all of these variations. For this reason, we can summarize all the possible on-interface situations in a systematical manner.

For all the on-interface schemes, it is easy to see that two mesh lines intersect with the interface at one node point. Apart from this special node point, all likely involved points in our schemes are eight adjacent points which all locate on these two meshlines. On each side of the interface, if two adjacent grid points are in the same domain, we call these two points a pair. We can have at most four pairs in a 2D domain. Patterns with three or four pairs are obviously easy to resolve. We classify all possible topological variations which have less than three pairs into three categories. In the first category, we have exactly two pairs on meshlines near the special node, see Fig. 4. Additionally, in the second category, we have only one pair of on meshline grid points near the special node, see Fig. 5. Finally, in the last category, there is no pair of on meshline grid points near the special node, see Fig. 6. In all categories, if the fictitious values in a grid point pattern can be solved by using one of

![Fig. 5](image1)

![Fig. 6](image2)
aforementioned six on-interface schemes, we encircle the required grid points in such a pattern by manganviolett dash lines. No manganviolett dash lines are presented if the fictitious values in a grid point pattern cannot be solved by using one of the aforementioned six on-interface schemes. In this situation, we need to utilize additional disassociation strategy, which was discussed in our earlier work [71]. Essentially, in case one cannot find required auxiliary points in constructing an interface scheme from a given direction, one may make use of the fictitious values obtained either from an interface scheme in another direction or from an interface scheme at another nearby location.

For the first category, there are four different types. The first type has two pairs of grid points which are on the same side of the interface but on different meshlines as depicted in Fig. 4(A). Then, depending on the interface morphology, there can be three different topological variations as shown in Fig. 4(A1)–(A3). Among these three variations, the case of Fig. 4(A1) can be resolved without using an interface scheme. While fictitious values in pattern Fig. 4(A3) can be solved by using one of the aforementioned six on-interface schemes. Only the fictitious values in pattern Fig. 4(A2) needs the disassociation strategy [71].

The second type of the first category has two pairs of grid points which are on the different sides of the interface and on different meshlines as depicted in Fig. 4(B). Depending on the interface morphology, there can be three different topological variations as shown in Fig. 4(B1)–(B3). Among these three variations, the cases of Fig. 4(B1) and (B3) can be solved by using one of the aforementioned six on-interface schemes. However, the fictitious values in pattern Fig. 4(B2) requires the use of the disassociation strategy [71].

In the third type of the first category, two pairs of grid points are on the same side of the interface and on the same meshline as depicted in Fig. 4(C). There are three different topological variations as shown in Fig. 4(C1)–(C3). Among them, the case of Fig. 4(C1) can be solved by using one of the aforementioned six on-interface schemes. Cases in Fig. 4(C3) can be resolved without using an interface scheme. Once again, we use the disassociation strategy to solve fictitious values in pattern Fig. 4(C3) [71].

Finally, the last type of the first category has two pairs of grid points which are on the different sides of the interface but on the same meshline as depicted in Fig. 4(D). One can see that there are three different topological variations as shown in Fig. 4(D1)–(D3). All of these cases can be resolved by using one of the aforementioned six on-interface schemes.

For the second category, there are six different topological variations for a pair in the computational domain, as shown in Fig. 5(E1)–(E6). The pattern in Fig. 5(E1) can be solved without the use of an interface scheme. Cases in Fig. 5(E2), (E4) and (E6) can be easily solved by using the aforementioned six on-interface schemes. The other two cases, Fig. 5(E3) and (E5) are readily resolved by the disassociation strategy [71].

In the last category, we do not have any pair in the computational domain as illustrated in Fig. 6. Interface schemes are required to resolve the first three distinguished situations. Cases in Fig. 6(F1) and (F2) can be solved by using the aforementioned six on-interface schemes. The third case as depicted in Fig. 6(F3) can be resolved by our disassociation strategy [71]. No interface scheme is needed in the case of Fig. 6(F4).

3. Theory and algorithm for three-material interfaces

This section presents new MIB methods for solving elliptic equations with discontinuous coefficients in three-material subdomains. These equations are solved in a 2D setting for the first time.

The three-material interface problem is that we have two interfaces $I^1$ and $I^2$ to divide the open bounded domain $\Omega \in \mathbb{R}^2$ into three subdomains $\Omega^a$, $\Omega^b$ and $\Omega^c$, such that $\Omega = \Omega^a \cup \Omega^b \cup \Omega^c$. We assume that the boundary $\partial \Omega$ and interfaces $I^1$ and $I^2$ are Lipschitz continuous and there are two piecewise smooth level-set functions $\Phi_1$ and $\Phi_2$ on $\Omega$ such that $I^1 = \{(x,y) | \Phi_1(x,y) = 0, \forall x,y \in \Omega \}$, $I^2 = \{(x,y) | \Phi_2(x,y) = 0, \forall x,y \in \Omega \}$, and $\partial \Omega = \{(x,y) | \Phi_1(x,y) = 0, \Phi_2(x,y) = 0, \forall x,y \in \Omega \}$. We seek the solution of the 2D elliptical equation with variable coefficient $\beta(x,y)$

$$
\left\{ \begin{array}{ll}
\beta(x,y) u_{x} u_{x} + \beta(x,y) u_{y} u_{y} = g(x,y), & x,y \in \Omega \\
\beta(x,y) u_{x} u_{x} + \beta(x,y) u_{y} u_{y} = g(x,y), & x,y \in \Omega \\
\beta(x,y) u_{x} u_{x} + \beta(x,y) u_{y} u_{y} = g(x,y), & x,y \in \Omega \\
\beta(x,y) u_{x} u_{x} + \beta(x,y) u_{y} u_{y} = g(x,y), & x,y \in \Omega \\
\beta(x,y) u_{x} u_{x} + \beta(x,y) u_{y} u_{y} = g(x,y), & x,y \in \Omega \\
\beta(x,y) u_{x} u_{x} + \beta(x,y) u_{y} u_{y} = g(x,y), & x,y \in \Omega
\end{array} \right.
$$

(34)

where $\beta(x,y)$ may be discontinuous at interfaces $I^1$ and $I^2$. We denoted function $\beta(x,y)$ in three different domains $\Omega^a$, $\Omega^b$ and $\Omega^c$ as $\beta^a(x,y)$, $\beta^b(x,y)$ and $\beta^c(x,y)$, respectively. All the points on $I^1$ are classified into following three categories.

**Category 1.** The points in this category belong to those parts of $I^1$ that do not intersect or touch $I^2$. These points are denoted as $O_1$. Then $\exists \epsilon_1 > 0$, $\forall x < \epsilon_1$ and $\epsilon > 0$, there exists an open disk $O(\epsilon_1, \epsilon)$, such that $O(\epsilon_1, \epsilon) \in \Omega^a \cup \Omega^b$. Jump conditions are given as

$$
[u]_{O_1} = u^a - u^b,
$$

(35)

$$
[\beta u]_{O_1} = \beta^a u^a - \beta^b u^b.
$$

(36)

Here, superscripts, $a$ and $c$, denote the limiting values of a function at point $O_1$ from $\Omega^a$ and $\Omega^c$, respectively.

**Category 2.** The points in this category locate in the parts of $I^1$ that overlap with $I^2$ and do not include the critical initial points where $I^1$ just intersects $I^2$. These points are denoted as $O_2$. Then $\exists \epsilon_1 > 0$, $\forall x < \epsilon_1$, and $\epsilon > 0$, there is a disk $O(\epsilon_2, \epsilon)$, such that $O(\epsilon_2, \epsilon) \in \Omega^a \cup \Omega^b$. Jump conditions are given as

$$
[u]_{O_2} = u^a - u^b,
$$

(37)

$$
[\beta u]_{O_2} = \beta^a u^a - \beta^b u^b.
$$

(38)
**Category 3.** The points in this category are the initial contacting points when $C^1$ and $C^2$ intersect, cross or meet each other. In other words, these points are the junction points of three subdomains. These points are denoted as $o_3$. Then $\forall \epsilon > 0, \exists o_3 \in \Omega^i$, $(i = a, b, c)$, such that $o_3 \in O(o_3, \epsilon)$. For points in Category 3, we may have six jump conditions, they define the interface relations between $\Omega^a$ and $\Omega^b$, between $\Omega^a$ and $\Omega^c$, and between $\Omega^b$ and $\Omega^c$. To make it clear, we use notation $[u]_{o_3}^{ij}$, $[\beta u_{n_{o_3}}]_{o_3}^{ij}$ ($i \neq j; i,j = a,b,c$). Jump conditions are expressed as

\[ [u]_{o_3}^{ab} = u^a - u^b, \]  
\[ [u]_{o_3}^{ac} = u^a - u^c, \]  
\[ [u]_{o_3}^{bc} = u^b - u^c, \]  
\[ \beta [u_{n_{o_3}}]_{o_3}^{ab} = \beta^a u_{n_{o_3}}^a - \beta^b u_{n_{o_3}}^b, \]  
\[ \beta [u_{n_{o_3}}]_{o_3}^{ac} = \beta^a u_{n_{o_3}}^a - \beta^c u_{n_{o_3}}^c, \]  
\[ \beta [u_{n_{o_3}}]_{o_3}^{bc} = \beta^b u_{n_{o_3}}^b - \beta^c u_{n_{o_3}}^c. \]  

There is no suitable definition for the unit normal vector (or vectors) at these junction points. It is necessary to have consistent jump conditions at junction points. However, choices of the unit normal vectors in the jump conditions are not important. Therefore, in our scheme, we just denote unit normal vectors between different domains as $n_1, n_2, n_3$. 

For all the three categories above, we can differentiate the jump condition $[u] = u^i - u^j$ along the tangential direction, here $i, j$ represent different domains ($i \neq j; i,j = a,b,c$). If the unit normal vector is defined as $\vec{n} = (\cos \theta, \sin \theta)$, then the unit tangential vector is $\vec{s} = (\sin \theta, -\cos \theta)$. We can have the third set of jump conditions

\[ [u]_{s_{o_3}}^{ij} = u_s^i \sin \theta + u_s^j \cos \theta - u_s^j \sin \theta + u_s^i \cos \theta. \]  

The above discussions are for points on $C^1$. For points on $C^2$, we can carry out similar classifications and provide other sets of jump conditions.

In this work, $\Omega^a$, $\Omega^b$, $\Omega^c$ are colored in yellow, green and red, respectively. Real challenges are due to geometric singularities, such as multiple junctions, sharp edges and sharp tips. Moreover, two interfaces may cross one meshline simultaneously within a grid spacing. In order to solve different situations encountered in three domain problems, we propose three different schemes.

### 3.1. Three-material interface scheme 1

For the three-material interface problems, if irregular points and auxiliary points are all in adjacent two domains, we just need to use the corresponding two domain schemes based on the geometry property. However, when irregular points and
auxiliary points locate in three different domains, there is no existing method to deal with these problems yet. We develop new MIB schemes for these situations. In Fig. 7, the singular point \((i, j)\) is in \(O^c\) and the adjacent points \((i-1, j)\) and \((i+1, j)\) locate in another two different domains, \(O^a\) and \(O^b\). We find that there are two interfaces beside point \((i, j)\) and they intersect with the grid meshline at point \(o\) and \(o'\). If we solve the problem by using the central difference scheme, we need four fictitious values \(f_{i-1,j}^{o^a}, f_{i,j}^{o^a}, f_{i+1,j}^{o^a}\), and \(f_{i,j+1}^{o^a}\). The superscripts here denote the directions of the extension. For example, \(f_{i-1,j}^{o^a}\) denotes the extension of the function value from \(O^c\) through point \(o\) to point \((i-1, j)\) and \(f_{i,j+1}^{o^a}\) denotes extension of the function value from \(O^b\) through point \(o'\) to point \((i, j+1)\). Here \(f_{i}^{o^a}\) and \(f_{i}^{o^b}\) are two different fictitious values. In Section 2.1, we have constructed three linear equations by enforcing two sets of interface jump conditions, and one more linear equation is left behind. Therefore, we just need to incorporate this equation and solve four fictitious values together. We give a detailed description below.

For point \(o\), using \(u_{i-2,j}, u_{i-1,j}, u_i, u_{i+1,j}\) and fictitious values \(f_{i-1,j}^{o^a}, f_{i,j}^{o^a}\), we can discretize the \(u_o^a, u_o^b, u_o^a\) and \(u_o^b\) explicitly as

\[
\begin{align*}
 u_o^a &= \begin{bmatrix} w_{0,1}^{o^a} & w_{0,1}^{o^a} & w_{0,1}^{o^a} \end{bmatrix}^T \cdot \begin{bmatrix} u_{i-2,j}, u_{i-1,j}, f_{i,j}^{o^a} \end{bmatrix}^T, \\
 u_o^b &= \begin{bmatrix} w_{0,1}^{o^b} & w_{0,1}^{o^b} \end{bmatrix} \cdot \begin{bmatrix} f_{i-1,j}, u_{i,j}, f_{i,j}^{o^b} \end{bmatrix}^T, \\
 u_x^a &= \begin{bmatrix} w_{1,0}^{o^a} & w_{1,0}^{o^a} \end{bmatrix} \cdot \begin{bmatrix} f_{i-1,j}, u_{i,j}, f_{i,j}^{o^a} \end{bmatrix}^T, \\
 u_x^b &= \begin{bmatrix} w_{1,0}^{o^b} & w_{1,0}^{o^b} \end{bmatrix} \cdot \begin{bmatrix} f_{i-1,j}, u_{i,j}, f_{i,j}^{o^b} \end{bmatrix}^T.
\end{align*}
\]

According to Fig. 7, \(u_o^a\) is to be discretized. We need six auxiliary points \(u_{i-3,j+1}, u_{i-2,j+1}, u_{i-1,j+1}, u_{i-2,j+2}, u_{i-2,j+2}\) and \(u_{i-1,j+2}\). The value of \(u_y^a\) can be represented as

\[
 u_y^a = \begin{bmatrix} w_{0,2}^{o^a} & w_{0,1}^{o^a} & w_{0,0}^{o^a} \\
 w_{1,1}^{o^a} & w_{1,0}^{o^a} & w_{0,0}^{o^a} \\
 w_{1,0}^{o^a} & w_{1,0}^{o^a} & w_{0,0}^{o^a} \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0 \end{bmatrix}^T + \begin{bmatrix} 0 & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} w_{0,0}^{o^a}, w_{0,0}^{o^a}, w_{0,0}^{o^a} \\
 w_{0,0}^{o^a}, w_{0,0}^{o^a}, w_{0,0}^{o^a} \\
 w_{0,0}^{o^a}, w_{0,0}^{o^a}, w_{0,0}^{o^a} \end{bmatrix} \cdot \begin{bmatrix} u_{i-2,j}, u_{i-1,j}, f_{i,j}^{o^a} \end{bmatrix}^T.
\]

Substituting Eqs. (46) and (47) into Eq. (7), we attain two different linear equations as

\[
\begin{aligned}
 \begin{bmatrix} u_o \end{bmatrix} &= \begin{bmatrix} w_{0,2}^{o^a} & w_{0,1}^{o^a} & w_{0,0}^{o^a} \\
 w_{1,1}^{o^a} & w_{1,0}^{o^a} & w_{0,0}^{o^a} \\
 w_{1,0}^{o^a} & w_{1,0}^{o^a} & w_{0,0}^{o^a} \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0 \end{bmatrix}^T + \begin{bmatrix} 0 & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} w_{0,0}^{o^a}, w_{0,0}^{o^a}, w_{0,0}^{o^a} \\
 w_{0,0}^{o^a}, w_{0,0}^{o^a}, w_{0,0}^{o^a} \\
 w_{0,0}^{o^a}, w_{0,0}^{o^a}, w_{0,0}^{o^a} \end{bmatrix} \cdot \begin{bmatrix} f_{i-1,j}, u_{i,j}, f_{i,j}^{o^a} \end{bmatrix}^T, \\
 \begin{bmatrix} \beta u_o \end{bmatrix} &= \begin{bmatrix} \beta \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} f_{i-1,j}, u_{i,j}, f_{i,j}^{o^a} \end{bmatrix}^T,
\end{aligned}
\]

Here the unit normal vector at point \(o\) is \(\bar{n} = (\cos \theta_o, \sin \theta_o)\), \(C_o^a = \beta^o \cos \theta_o + \beta^f \tan \theta_o \sin \theta_o\), \(C_o^b = \beta^o \cos \theta_o + \beta^f \tan \theta_o \sin \theta_o\), and \(C_o^c = \beta^o \sin \theta_o - \beta^f \sin \theta_o\). Therefore, we obtain two linear equations by using the jump conditions of interface intersecting point \(o\). For point \(o\), we can implement the same idea, and discretize \(u_o^b, u_o^f, u_o^b\) and \(u_o^f\) explicitly as

\[
\begin{aligned}
 u_o^b &= \begin{bmatrix} w_{0,1}^{o^b} & w_{0,0}^{o^b} \end{bmatrix} \cdot \begin{bmatrix} f_{i-1,j}, u_{i,j}, u_{i+1,j} \end{bmatrix}^T, \\
 u_o^f &= \begin{bmatrix} w_{0,1}^{o^f} & w_{0,0}^{o^f} \end{bmatrix} \cdot \begin{bmatrix} f_{i-1,j}, u_{i,j}, u_{i+1,j} \end{bmatrix}^T, \\
 u_x^b &= \begin{bmatrix} w_{1,1}^{o^b} & w_{1,0}^{o^b} \end{bmatrix} \cdot \begin{bmatrix} f_{i-1,j}, u_{i,j}, u_{i+1,j} \end{bmatrix}^T, \\
 u_x^f &= \begin{bmatrix} w_{1,1}^{o^f} & w_{1,0}^{o^f} \end{bmatrix} \cdot \begin{bmatrix} f_{i-1,j}, u_{i,j}, u_{i+1,j} \end{bmatrix}^T.
\end{aligned}
\]

According to Fig. 7, \(u_o^b\) is to be discretized. We need other six auxiliary points \(u_{i+1,j+1}, u_{i+2,j+1}, u_{i+3,j+1}, u_{i+1,j+2}, u_{i+2,j+2}\) and \(u_{i+3,j+2}\). Then \(u_o^b\) can be represented as

\[
 u_o^b = \begin{bmatrix} w_{0,1}^{o^b} & w_{0,0}^{o^b} \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0 \end{bmatrix}^T + \begin{bmatrix} 0 & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} w_{0,0}^{o^b}, w_{0,0}^{o^b}, w_{0,0}^{o^b} \\
 w_{0,0}^{o^b}, w_{0,0}^{o^b}, w_{0,0}^{o^b} \\
 w_{0,0}^{o^b}, w_{0,0}^{o^b}, w_{0,0}^{o^b} \end{bmatrix} \cdot \begin{bmatrix} f_{i-1,j}, u_{i,j}, f_{i,j}^{o^b} \end{bmatrix}^T.
\]
Substituting Eqs. (50) and (51) into Eq. (7), we attain other different two linear equations as
\[
[u]_{i,o} = \left( w_{0,i+1,i}^{bo} \right)^T \cdot (f_{i,j}^{bo}, u_{i+1,j}, u_{i+2,j}) - \left( w_{0,i-1,i}^{bo} \right)^T \cdot (f_{i,j}^{bo}, u_{i,j}, f_{i+1,j})^T
\] (52)
\[
[\beta u^o_{i,o}] = [C]_{i,o}^{bo} \cdot \left( \{U\}_{3,23} \right)^T
\] (53)

Here the unit normal vector at point \( \gamma \) is \( \hat{n} = (\cos \theta, \sin \phi) \). Interpretations for \( [C]_{i,o}^{bo} \), \( w_{0,i+1,i}^{bo} \), \( w_{0,i-1,i}^{bo} \), etc. are very similar to their counterparts in earlier descriptions and are omitted here. Therefore, we have four equations, Eqs. (48), (49), (52) and (53). We can solve four fictitious values from these four equations and attain the expressions of node values and jump conditions as
\[
\begin{align*}
\{f_{i,j}^{bo}\} &= \left( [C]_{i,o}^{bo} \right)^{-1} \cdot \left[ [u]_{i,o}, [\beta u^o_{i,o}]^{bo}, \{U\}_{23,1} \right]^T, \\
\{x\} &= \left( [C]_{i,o}^{bo} \right)^{-1} \cdot \left[ [u]_{i,o}, [\beta u^o_{i,o}]^{bo}, \{U\}_{23,1} \right]^T.
\end{align*}
\] (54)

where vector \( \{U\}_{23,1} \) consists of 17 function values and 6 jump conditions
\[
\{U\}_{23,1} = (u_{i-2,j}, u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, u_{i+3,j+1}, u_{i+2,j+1}, u_{i+1,j+1}, u_{i,j+1}, u_{i-1,j+1}, u_{i-2,j+1}, u_{i-3,j+2}, u_{i-2,j+2}, u_{i-1,j+2})
\] (55)

Here, \( [C]_{3,23} \) is the coefficient matrix and its components are the combination of weights and trigonometric functions of unit normal vectors at points \( o \) and \( \gamma \). Finally, one can discretize \((\beta u^o)_{x}\) at irregular points \((i-1,j), (i,j), (i+1,j)\) as
\[
\begin{align*}
(\beta u^o)_{x} &= \frac{1}{\Delta x} \left( \beta_{i-1/2,j} - \beta_{i+1/2,j} - \beta_{i-1/2,j} - \beta_{i+1/2,j} \right) \cdot (u_{i-1,j} - u_{i,j} - f_{i,j})^T \quad \text{at} \quad (i-1,j), \\
(\beta u^o)_{x} &= \frac{1}{\Delta x} \left( \beta_{i+1/2,j} - \beta_{i-1/2,j} - \beta_{i+1/2,j} - \beta_{i-1/2,j} \right) \cdot (f_{i,j}^{bo}, u_{i,j} - f_{i+1,j})^T \quad \text{at} \quad (i,j), \\
(\beta u^o)_{x} &= \frac{1}{\Delta x} \left( \beta_{i+1/2,j} - \beta_{i-1/2,j} - \beta_{i+1/2,j} - \beta_{i-1/2,j} \right) \cdot (f_{i,j}^{bo}, u_{i+1,j} - u_{i,j})^T \quad \text{at} \quad (i+1,j).
\end{align*}
\] (56)

The main idea of Three-material interface scheme 1 is to use the information at two interface intersecting points together to extend the function values beyond each individual subdomain and attain fictitious values. More attention should be paid to fictitious values at the central point like \((i,j)\) in Fig. 7 as two fictitious values at this point are different. We need to be careful to use correct fictitious values to discretize \((\beta u^o)_{x}\).

### 3.2. Three-material interface scheme 2

In three-material interface problem, when interfaces intersect with the grid mesh, we can make use of the disassociation strategy [71] and/or on-interface schemes [66], depending on the geometry. The disassociation strategy can be used to solve fictitious values for discretization no matter how they are calculated. Therefore, if we can compute the fictitious values by using interface conditions at nearby locations, we can directly implement the central difference scheme. For example, for situations in Fig. 8(a) and (b), we just need to treat the interface intersecting node as a node in \( \Omega^e \) and make use of the disassociation strategy at points \((i-1,j)\) and \((i+1,j)\). However, for situations in Fig. 9(a) and (b), the simple disassociation approach does not work. To solve this kind of problems, a disassociation approach should be implemented at points \((i,j+1)\) and \((i,j-1)\). Two fictitious values \((i,j+1)\) and \((i,j-1)\) are then treated as secondary auxiliary points. Additionally, On-interface scheme 4 is used to attain the fictitious values we need. We give a detailed description below. In Fig. 9(b), if we treat the interface intersecting node \((i,j)\) as a point in \( \Omega^e \), that is \( u_{i,j} = u_{i,j}^{bo} \), then using the jump condition between \( \Omega^e \) and \( \Omega^p \), we can obtain \( u_{i,j}^{bo} = u_{i,j}^{bo} - u_{i,j}^{bo} \). The \( u_{i,j}^{bo} \) here means the jump value between \( \Omega^e \) and \( \Omega^p \) at point \( o \). When we introduce a fictitious value \( f_{i,j} \), which is the extension of function in \( \Omega^p \) to point \((i+1,j)\), we can discretize \( u_{i,j}^{bo} \) and \( u_{i,j}^{bo} \) explicitly as
\[
\begin{align*}
u_{i,j}^{bo} &= \left( \left( w_{0,i+1,j}^{bo}, w_{0,i+2,j}, w_{0,i+3,j+1}^{bo} \right)^T \cdot (u_{i-1,j}, u_{i,j}, f_{i+1,j}) \right)^T, \\
u_{i,j}^{bo} &= \left( \left( w_{0,i+1,j}^{bo}, w_{0,i+2,j}, w_{0,i+3,j+1}^{bo} \right)^T \cdot (u_{i-1,j} - u_{i,j}^{bo}, u_{i+1,j}, u_{i+2,j}) \right)^T.
\end{align*}
\] (57)
For points \((i, j + 1)\) and \((i, j - 1)\), we can calculate their fictitious values from both sides of the interface using two sets of jump conditions, as we have mentioned in Three-material interface scheme 1. We only choose the fictitious values from one side, so that \(f_{i,j-1}\) and \(f_{i,j+1}\) are the extensions of the function values from the same domain. Here we choose two fictitious values from \(\Omega^2\). As a result, it is easy for us to obtain an explicit representation of \(u^{(0)}_{j}\) in terms of \(f_{i,j-1}, f_{i,j+1}\) and \(u_{ij}\) as

\[
u^{(0)}_{j} = \left( w^{(0)}_{i,j-1}, w^{(0)}_{i,j}, w^{(0)}_{i,j+1} \right) \cdot \left( f_{i,j-1}, u_{ij}, f_{i,j+1} \right)^{T}.
\]

By substituting Eqs. (57) and (58) into Eq. (7), we have

\[
\begin{align*}
(\beta u_{ij})^{ab} - \beta^{b} \tan \theta_{ij} |u_{ij}^{ab}| &= C^{\alpha}_{x} \begin{bmatrix} w^{(0)}_{i,j-1} & w^{(0)}_{i,j} & w^{(0)}_{i,j+1} \end{bmatrix}^{T} \begin{bmatrix} u_{i,j-1} & u_{i,j} & u_{i,j+1} \end{bmatrix} - C^{\alpha}_{y} \begin{bmatrix} w^{(0)}_{i-1,j} & w^{(0)}_{i+1,j} & w^{(0)}_{i+2,j} \end{bmatrix} \begin{bmatrix} u_{i-1,j} & u_{i+1,j} & u_{i+2,j} \end{bmatrix} + C^{\alpha}_{y} \begin{bmatrix} w^{(0)}_{i-1,j} & w^{(0)}_{i+1,j} & w^{(0)}_{i+2,j} \end{bmatrix} \begin{bmatrix} f_{i,j-1} & f_{i,j+1} & f_{i,j+1} \end{bmatrix},
\end{align*}
\]

here \(C^{\alpha}_{x} = \beta^{a} \cos \theta + \beta^{b} \tan \theta \sin \theta\), \(C^{b}_{x} = \beta^{a} \cos \theta - \beta^{b} \tan \theta \sin \theta\) and \(C^{\alpha}_{y} = \beta^{a} \sin \theta - \beta^{b} \sin \theta\). We can calculate \(f_{i+1,j}\) from Eq. (59) as

\[
\begin{align*}
f_{i+1,j} &= \{ C \} \cdot \left( u_{i,j-1}, u_{i,j}, u_{i,j+1}; [\beta u_{i,j}]^{ab}, [u_{i,j}]^{ab}, f_{i,j-1}, f_{i,j+1} \right)^{T} \\
&= \{ C_{1} \} \cdot \left( u_{i,j-1}, u_{i,j}, u_{i,j+1}; [\beta u_{i,j}]^{ab}, [u_{i,j}]^{ab} \right)^{T} + \{ C_{2} \} \cdot f_{i,j-1} + \{ C_{3} \} \cdot f_{i,j+1} \\
&= \{ C_{1} \} \cdot \left( u_{i,j-1}, u_{i,j}, u_{i,j+1}; [\beta u_{i,j}]^{ab}, [u_{i,j}]^{ab} \right)^{T} + \{ C_{2} \} \cdot \{ C_{1} \}^{j+1}_{1,23} \cdot \{ U \}^{j+1}_{23} + \{ C_{3} \} \cdot \{ C_{1} \}^{j+1}_{1,23} \cdot \{ U \}^{j+1}_{23} \\
&= \{ C \}^{j+1}_{1,23} \cdot \{ U \}^{j+1}_{23}.
\end{align*}
\]

Fig. 8. Three-material interface scheme 2. The left yellow area is \(\Omega^a\), the right green area is \(\Omega^b\), the middle red area is \(\Omega^c\). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 9. Three-material interface scheme 2. The left yellow area is \(\Omega^a\), the right green area is \(\Omega^b\), the middle red area is \(\Omega^c\). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
where \( \{C\} = \{\{C_1\}, \{C_2\}, \{C_3\}\} \) and \( f_{i,j} = \{(U)^{i,j-1}_{23}\} \cdot \{(U)^{i,j-1}_{23}\} \cdot \{(U)^{i,j-1}_{23}\} \). Although we write \((U)\) as a vector \((U)_{52,1}\) including 38 function values and 15 jump conditions, some of the function values may have been repeated counted, as we have different ways of choosing auxiliary points for fictitious value \( f_{i,j} \) and \( f_{i,j+1} \). Therefore, the number of independent components is usually less than 53. Moreover, if we pay more attention to Fig. 9(b), we can find out that if the unit normal vector at the point \( o \) is defined as \( \vec{n} = (\cos \theta, \sin \theta) = (1, 0) \), Eq. (59) can be simplified as:

\[
\left[ \beta u_{i,j} \right]_{ab} = C^a_x \left[ \begin{array}{c} w_{1,j-1}^0 \\ w_{1,j}^0 \\ \vdots \\ w_{1,j+1}^0 \end{array} \right]^T \left[ \begin{array}{c} u_{i-1,j} \\ u_{i,j} \\ \vdots \\ u_{i+1,j} \end{array} \right] - C^b_x \left[ \begin{array}{c} w_{1,j-1}^0 \\ w_{1,j}^0 \\ \vdots \\ w_{1,j+1}^0 \end{array} \right]^T \left[ \begin{array}{c} u_{i,j} - \left[ \beta u_{i,j} \right]_{ab} \\ u_{i+1,j} \end{array} \right].
\]

(61)

We can easily solve \( f_{i,j} \) from the equation.

\[
f_{i,j} = \left\{ C \right\}_{3,6} \cdot \left( u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, [u_{i,j}^o, [\beta u_{i,j}^o]^T. \right.
\]

(62)

So one can discretize \((\beta u_{i,j})_x\) and \((\beta u_{i,j})_y\) at \( \Omega^a \) point \((i,j)\) as

\[
(\beta u_{i,j})_x = \frac{1}{\Delta x^2} \left( \rho_i^a - \rho_{i+1,j}^a - \rho_{i-1,j}^a + \rho_{i,j}^a \right) \cdot \left( u_{i-1,j}, u_{i,j}, u_{i+1,j} \right)^T \quad \text{at}(i,j).
\]

(63)

\[
(\beta u_{i,j})_y = \frac{1}{\Delta y^2} \left( \rho_i^a - \rho_{i,j+1}^a - \rho_{i,j-1}^a + \rho_{i,j}^a \right) \cdot \left( f_{i,j}, f_{i,j+1}, f_{i,j+1} \right)^T \quad \text{at}(i,j).
\]

### 3.3. Three-material interface scheme 3

When two interfaces intersect with mesheselines within a grid spacing, there will be a smallest acute angle limitation in the 2D domain for three-material situation as depicted in Fig. 10. When the angle is smaller than 2 arctan \((1/2)\), it is always difficult for us to find enough auxiliary points even when we implement the disassociation strategy. However, if the angle is bigger than 2 arctan \((1/2)\), we can find out that at least one of points \((i-1,j+1), (i,j+1), (i+1,j+1), (i+2,j+1), (i-1,j-1), (i-1,j-1), (i+2,j-1) \) and \((i+2,j-1)\) must locate in \( \Omega^a \). For example, here point \((i,j-1)\) locates in \( \Omega^a \). Since point \((i,j)\) is an irregular point in \( \Omega^a \), we can obtain fictitious value \( f_{i,j}^a \). Then we implement the disassociation strategy and create a new scheme as follows.

For situation in Fig. 10, we can obtain fictitious value \( f_{i,j}^a \) by using the information of a pair of irregular points \((i,j-1)\) and \((i,j)\). In order to use the jump condition at \( o \) and \( o' \) to implement the center difference scheme, we need to calculate two fictitious values \( f_{i,j}^{o,\alpha} \) and \( f_{i,j}^{o',\alpha} \). To make it clear, \( f_{i,j}^{o,\alpha} \) means the fictitious value at point \((i+1,j)\) and it is an extension of the function value from domain \( \Omega^a \). Two more fictitious values from \( \Omega^b \) are needed and they are combined with fictitious value \( f_{i,j}^{o,\alpha} \) to interpolate \( u_{i,j}^{o,\alpha} \). These two fictitious values are located adjacent to \( f_{i,j}^{o,\alpha} \) and they can be either \( f_{i,j}^{o,\alpha} \) and \( f_{i,j}^{o',\alpha} \) or \( f_{i,j}^{o,\alpha} \) and \( f_{i,j}^{o',\alpha} \). Let us just choose \( f_{i,j}^{o,\alpha} \) and \( f_{i,j}^{o',\alpha} \) in the computation below.

For point \( o \), we can discretize \( u_{i,j}^{o,\alpha}, u_{i,j}^{o',\alpha} \) and \( u_{i,j}^{o,\alpha} \) by using function values \( u_{i-1,j} \) and \( u_{i,j} \), and fictitious values \( f_{i,j}^{o,\alpha}, f_{i,j}^{o',\alpha} \) and \( f_{i,j}^{o,\alpha} \).
By substituting Eqs. (64) and (65) into Eq. (7), we attain two different linear equations as

\[ u_{ao} = \frac{w_{ao}}{C_0} u_{i,j}^{wao} + \frac{w_{bo}}{C_1} u_{i,j+1}^{wbo} + \frac{w_{co}}{C_0} u_{i,j}^{wco} + \frac{w_{co}}{C_1} u_{i,j+1}^{wco} \]  

Depending on the geometry, we can choose suitable auxiliary points to discretize \( u_{ao} \) or \( u_{bo} \). Here we need six more auxiliary values \( u_{i-3,j-1}, u_{i-2,j-1}, u_{i-1,j-1}, u_{i,j-1}, u_{i+1,j} \) and \( u_{i,j+1} \). Value \( u_{ao} \) can be represented as follows

\[ u_{ao} = \begin{bmatrix} w_{ao,0} & w_{ao,1} & \cdots & w_{ao,1} & w_{ao,2} \\ w_{ao,3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & w_{ao,1} & 0 \\ 0 & 0 & \cdots & 0 & w_{ao,1} \\ \end{bmatrix} \begin{bmatrix} u_{i,j-1} \\ u_{i,j} \\ u_{i,j+1} \\ u_{i+1,j} \\ u_{i+2,j} \\ \end{bmatrix} + \begin{bmatrix} f_{c}^{wao} \\ f_{c}^{wao} \\ f_{c}^{wao} \\ f_{c}^{wao} \\ f_{c}^{wao} \\ \end{bmatrix} \]  

(64)

By substituting Eqs. (64) and (65) into Eq. (7), we attain two different linear equations as

\[ [u]_o = \begin{bmatrix} w_{ao,0} & w_{ao,1} & \cdots & w_{ao,1} & w_{ao,2} \\ w_{ao,3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & w_{ao,1} & 0 \\ 0 & 0 & \cdots & 0 & w_{ao,1} \\ \end{bmatrix} \begin{bmatrix} u_{i,j-1} \\ u_{i,j} \\ u_{i,j+1} \\ u_{i+1,j} \\ u_{i+2,j} \\ \end{bmatrix} + \begin{bmatrix} f_{c}^{wao} \\ f_{c}^{wao} \\ f_{c}^{wao} \\ f_{c}^{wao} \\ f_{c}^{wao} \\ \end{bmatrix} \]  

(65)

Therefore, we obtain two linear equations by using the information of interface intersecting point o. For point o', we can make use of a similar idea and discretize \( u_{bo} \), \( u_{co} \), \( \beta \) and \( \theta \) explicitly as

\[ u_{bo}^o = \begin{bmatrix} w_{bo,0} & w_{bo,1} & \cdots & w_{bo,1} & w_{bo,2} \\ w_{bo,3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & w_{bo,1} & 0 \\ 0 & 0 & \cdots & 0 & w_{bo,1} \\ \end{bmatrix} \begin{bmatrix} u_{i,j-1} \\ u_{i,j} \\ u_{i,j+1} \\ u_{i+1,j} \\ u_{i+2,j} \\ \end{bmatrix} + \begin{bmatrix} f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ \end{bmatrix} \]  

(66)

According to Fig. 10, \( u_{bo}^o \) is to be discretized. We need other six auxiliary points \( u_{i,j-1}, u_{i,j-1}+1, u_{i,j}, u_{i,j+1}, u_{i,j+1}+1, u_{i,j+2} \) and \( u_{i,j+3} \). Then \( u_{ao}^o \) can be represented as

\[ u_{ao}^o = \begin{bmatrix} w_{ao,0} & w_{ao,1} & \cdots & w_{ao,1} & w_{ao,2} \\ w_{ao,3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & w_{ao,1} & 0 \\ 0 & 0 & \cdots & 0 & w_{ao,1} \\ \end{bmatrix} \begin{bmatrix} u_{i,j-1} \\ u_{i,j} \\ u_{i,j+1} \\ u_{i+1,j} \\ u_{i+2,j} \\ \end{bmatrix} + \begin{bmatrix} f_{c}^{wao} \\ f_{c}^{wao} \\ f_{c}^{wao} \\ f_{c}^{wao} \\ f_{c}^{wao} \\ \end{bmatrix} \]  

(67)

Substituting Eqs. (68) and (69) into Eq. (7), we obtain two other linear equations as

\[ [u]_o = \begin{bmatrix} w_{bo,0} & w_{bo,1} & \cdots & w_{bo,1} & w_{bo,2} \\ w_{bo,3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & w_{bo,1} & 0 \\ 0 & 0 & \cdots & 0 & w_{bo,1} \\ \end{bmatrix} \begin{bmatrix} u_{i,j-1} \\ u_{i,j} \\ u_{i,j+1} \\ u_{i+1,j} \\ u_{i+2,j} \\ \end{bmatrix} + \begin{bmatrix} f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ \end{bmatrix} \]  

(68)

\[ [\beta u]_o - \beta \tan \theta [u]_o = \begin{bmatrix} f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ \end{bmatrix} \begin{bmatrix} u_{i,j-1} \\ u_{i,j} \\ u_{i,j+1} \\ u_{i+1,j} \\ u_{i+2,j} \\ \end{bmatrix} \]  

(69)

Substituting Eqs. (68) and (69) into Eq. (7), we obtain two other linear equations as

\[ [u]_o = \begin{bmatrix} w_{bo,0} & w_{bo,1} & \cdots & w_{bo,1} & w_{bo,2} \\ w_{bo,3} & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & w_{bo,1} & 0 \\ 0 & 0 & \cdots & 0 & w_{bo,1} \\ \end{bmatrix} \begin{bmatrix} u_{i,j-1} \\ u_{i,j} \\ u_{i,j+1} \\ u_{i+1,j} \\ u_{i+2,j} \\ \end{bmatrix} + \begin{bmatrix} f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ \end{bmatrix} \]  

(70)

\[ [\beta u]_o - \beta \tan \theta [u]_o = \begin{bmatrix} f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ f_{c}^{wbo} \\ \end{bmatrix} \begin{bmatrix} u_{i,j-1} \\ u_{i,j} \\ u_{i,j+1} \\ u_{i+1,j} \\ u_{i+2,j} \\ \end{bmatrix} \]  

(71)
As such, we have attained four equations Eqs. (66), (67), (70) and (71). Here \( f_{ij}^0 \) has already been expressed in terms of function values and jump conditions. So we just need to calculate four other fictitious values and their expressions can be written as

\[
\begin{bmatrix}
    f_{i+1,j}^0 \\
    f_{ij}^0 \\
    f_{ij}^{op} \\
    f_{i+1,j}^{op}
\end{bmatrix} = (C)_{4\times45} \cdot \{U\}_{45\times1}.
\]

Here vector \( \{U\}_{45\times1} \) consists of 17 function values and 6 jump conditions from \( f_{ij}^0 \), and other 16 function values and 6 jump conditions as

\[
\{U\}_{22\times1} = (u_{i,j}, u_{i,j}, u_{i,j+1}, u_{i,j+2}, u_{i+1,j}, u_{i+2,j}, u_{i,j-1}, u_{i,j}, u_{i+1,j}, u_{i+1,j+1}, u_{i+2,j+1}, u_{i,j+1}, u_{i,j+2}, u_{i+1,j}, u_{i+1,j+1}, u_{i+2,j+1}, u_{i,j+1}, u_{i,j+2}, u_{i+1,j}, u_{i+1,j+1}, u_{i+2,j+1}, u_{i,j+1}, u_{i,j+2}, u_{i+1,j}, u_{i+1,j+1}, u_{i+2,j+1}, u_{i,j+1}, u_{i,j+2}, u_{i+1,j}, u_{i+1,j+1}, u_{i+2,j+1}, u_{i,j+1}, u_{i,j+2}, u_{i+1,j}, u_{i+1,j+1}, u_{i+2,j+1}, u_{i,j+1}, u_{i,j+2}, u_{i+1,j}, u_{i+1,j+1}, u_{i+2,j+1}, \}
\]

If we denote \( f_{ij}^0 = (C)_{4\times45} \cdot \{U\}_{45\times1} \) then \( \{U\}_{45\times1} = (U)_{22\times1} \cup (U)_{22\times1} \). Here \((C)_{4\times45}\) is the coefficient matrix and its components are the combination of weights and trigonometric functions of the unit normal vectors. Finally, one can discretize \((\beta_{ij})_x, (\beta_{ij})_y, (\beta_{ij})_z, (\beta_{ij})_y, (\beta_{ij})_z\), and \((\beta_{ij})_y, (\beta_{ij})_z\) at irregular points, \((i,j)\) and \((i+1,j)\) as

\[
(\beta_{ij})_x = \frac{1}{\Delta x^2} \left[ \beta_{i+1,j}^y - \beta_{ij}^y - \beta_{ij}^y - \beta_{i+1,j}^y \right] \cdot (u_{i-1,j} - u_{i+1,j}) \quad \text{at } (i,j),
\]

\[
(\beta_{ij})_y = \frac{1}{\Delta y^2} \left[ \beta_{i,j+1}^y - \beta_{ij}^y - \beta_{ij}^y - \beta_{ij}^y \right] \cdot (u_{i,j+1} - u_{ij}) \quad \text{at } (i+1,j).
\]

### 4. Numerical studies

In this section, we examine the performance of the proposed MIB schemes for solving the Poisson equation with geometric singularities of two-material interfaces and three-material interfaces. Section 4.1 presents two case studies about two-material interfaces. The geometry of Case 1 has many sharp tips and it is designed to test the proposed two-material interface scheme. In Case 2, the vertex of the triangular interface intersects with a grid node. This case is used to examine the on-interface schemes proposed in this work. Five numerical tests of three-material interfaces are presented in Section 4.2. We construct a number of different geometric shapes, solution functions and coefficient contrasts to test the robustness and demonstrate the efficiency of our three-material MIB schemes.

The standard \( L_\infty \) and \( L_2 \) error measurements are employed in this section. For all test cases, the computational domains are set to be the square of \( \Omega = [-1,1] \times [-1,1] \). Basic geometries are depicted on \( 20 \times 20 \) meshes and our numerical results are presented on \( 80 \times 80 \) meshes. The Dirichlet boundary condition is assumed in all the cases, although it is also easy to implement other boundary conditions. In all the test studies, the designed second order accuracy is confirmed.

### 4.1. Two-material interface problems

New two-material interface schemes are validated in this subsection. We designed two test examples to demonstrate the performance of our MIB method. 

**Case 1.** In this case, the 2D Poisson Eq. (1) is solved. To specify the interface, we design the level set function \( \phi(x,y) \)

\[
\phi(x,y) = -(y - 20x)(y + 20x)(x - 20y)(x + 20y),
\]

so that \( \Gamma = \{(x,y)|\phi = 0; x,y \in \Omega\}, \Omega^e = \{(x,y)|\phi > 0; x,y \in \Omega\}, \Omega^b = \{(x,y)|\phi < 0; x,y \in \Omega\}. \) The discontinuous coefficients are given by

\[
\beta^e(x,y) = 1, \quad \beta^b(x,y) = 2.
\]

The solution at two different domains is designed as

\[
u^e(x,y) = 2 + \sin(4\pi x) \sin(4\pi y), \quad u^b(x,y) = \sin(2\pi x) \sin(2\pi y).
\]

In this case, we test our two-material interface schemes. Fig. 11 presents the geometric shape and computed results for this problem. It is seen that on a \( 20 \times 20 \) grid, except for the origin, which is an on-interface node and treated as an point in \( \Omega^b \), all the points on the xy coordinates are irregular points. No matter how one decreases the mesh size, grid points near the origin are still irregular and need special treatments. Table 1 lists \( L_\infty \) and \( L_2 \) errors and their numerical orders. It is seen that the second order accuracy is confirmed.

**Case 2.** In this case, the 2D Poisson Eq. (1) is solved and \( \Omega^b \) consists of two parts which can be described by two level set functions
\[ \phi_1(x, y) = - \left( y - \frac{1}{\sqrt{2}}x \right) \left( y + \frac{1}{\sqrt{2}}x \right) \left( y - \frac{1}{\sqrt{3}} \right); \]
\[ \phi_2(x, y) = - \left( y + 3x^2 + \frac{1}{\sqrt{2}} \right). \]

\[ \Omega^a = \{(x, y) | \phi_1(x, y) \geq 0; x, y \in \Omega \} \cup \{(x, y) | \phi_2(x, y) \geq 0; x, y \in \Omega \}, \quad \Omega^b = \{(x, y) | \phi_1(x, y) \leq 0; x, y \in \Omega \} \cap \{(x, y) | \phi_2(x, y) \leq 0; x, y \in \Omega \}. \]

The basic geometry is illustrated in Fig. 12. The discontinuous coefficients are given by
\[ \beta^a(x, y) = 1, \quad \beta^b(x, y) = 2. \]

The solution at two different domains is designed as
\[ u^a(x, y) = 2 + \sin(2\pi x) \sin(2\pi y), \quad u^b(x, y) = \sin(2\pi x) \sin(2\pi y). \]

When the grid is 20 × 20, On-interface scheme 4 is needed because the interface intersects with a node point of the mesh. As for the unit normal vector of the interface intersecting node, we use \( \mathbf{n} = \left( 1/\sqrt{50}, -7/\sqrt{50} \right) \) in our computation. However, for the singular point, the way of choosing the unit normal vector does not matter too
much as long as the corresponding jump condition at normal vector is given accordingly. In a $20 \times 20$ grid if we change the unit normal vector to $\vec{n} = (0, -1)$, $L_\infty$ error changes from $0.1973$ to $0.1956$. Therefore, we conclude that On-interface scheme 4 works well for the singular point. Table 2 lists computed $L_\infty$ and $L_2$ errors. The second order convergence is observed.

4.2. Three-material interface problems

This subsection is devoted to validate the proposed MIB schemes for three-material interfaces. We consider a variety of geometric situations, topological variations, coefficient contrasts and solution types to illustrate the utility, test robustness, and demonstrate the accuracy of our method in five cases.

Case 3. In this case, the 2D Poisson Eq. (34) is solved. Now domain $\Omega$ is divided into three subdomains $\Omega^a$, $\Omega^b$ and $\Omega^c$ by two interfaces $\Gamma_1$ and $\Gamma_2$, which can be described by two level set functions $\phi_1(x,y)$ and $\phi_2(x,y)$ (See Fig. 13)

\[
\phi_1(x,y) = \frac{8}{17} - \sqrt{\left(x + \frac{8}{17}\right)^2 + y^2}, \quad (82)
\]
\[
\phi_2(x,y) = \frac{8}{17} - \sqrt{\left(x - \frac{8}{17}\right)^2 + y^2}, \quad (83)
\]

such that $\Omega^a = \{(x,y) | \phi_1(x,y) \geq 0; x,y \in \Omega\}$, $\Omega^b = \{(x,y) | \phi_2(x,y) \geq 0; x,y \in \Omega\}$, and $\Omega^c = \{(x,y) | \phi_1(x,y) \leq 0, \phi_2(x,y) \leq 0; x,y \in \Omega\}$. We test the robustness of our scheme by choosing different combinations of $\beta$ values and solution types. Here six different combinations of solutions and $\beta$ values are studied.

- Case 3(a):

  $\beta^a = 1$, $\beta^b = 2$, $\beta^c = 3$, \quad (84)

  $u^a(x,y) = 8 + \sin(4\pi x) \sin(4\pi y)$, \quad (85)

  $u^b(x,y) = 4 + \sin(2\pi x) \sin(2\pi y)$, \quad (86)

  $u^c(x,y) = x^2 + y^2 + \sin(4\pi x) \sin(4\pi y)$; \quad (87)

Fig. 13. Basic geometry on $20 \times 20$ mesh for Case 3.
- **Case 3(b):**

  \[
  \beta^a = 1, \beta^b = 2 + \sin(x + y), \beta^c = 3 + x^2 + y^2; \\
  u^a(x, y) = 8 + \sin(4\pi x) \sin(4\pi y), \\
  u^b(x, y) = 4 + \sin(2\pi x) \sin(2\pi y), \\
  u^c(x, y) = x^2 + y^2 + \sin(4\pi x) \sin(4\pi y);
  \]

  \((88)\)  

  \((89)\)  

  \((90)\)  

  \((91)\)  

  ![Fig. 14. Illustration of numerical solutions on the 80 × 80 mesh for Case 3.](image-url)
• Case 3(c):
\[
\begin{align*}
\beta^a &= 0.0001, \beta^b = 10000, \beta^c = 30000, \\
b^a(x,y) &= 4 + x^2 + y^2, \\
b^b(x,y) &= 6 + \exp(x + y), \\
b^c(x,y) &= x + y + xy;
\end{align*}
\]  
(92)

• Case 3(d):
\[
\begin{align*}
\beta^a &= 100, \beta^b = 100, \beta^c = 3000, \\
b^a(x,y) &= 8 + \sin(x + y), \\
b^b(x,y) &= 4 + \sin(x + y), \\
b^c(x,y) &= x^2 + y^2 + \sin(x + y);
\end{align*}
\]  
(96)

• Case 3(e):
\[
\begin{align*}
\beta^a &= 1, \beta^b = 2, \beta^c = 3000, \\
b^a(x,y) &= 8 + \sin(4\pi x) \sin(4\pi y), \\
b^b(x,y) &= 4 + \sin(2\pi x) \sin(2\pi y), \\
b^c(x,y) &= x^2 + y^2 + \sin(4\pi x) \sin(4\pi y);
\end{align*}
\]  
(100)

• Case 3(f):
\[
\begin{align*}
\beta^a &= 1000, \beta^b = 2, \beta^c = 3000, \\
b^a(x,y) &= 4 + \sin(x) \sin(y), \\
b^b(x,y) &= 8 + \pi x \sin(\pi y), \\
b^c(x,y) &= x^2 + y^2 + \sin(\pi x) \sin(\pi y);
\end{align*}
\]  
(104)

The geometry and solutions are illustrated in Fig. 14. Normally, when the singular point is located on a mesh node, the definition of its unit normal vector is necessary for our computation. So for the origin, the unit normal vector should be specified for three sets of jump conditions (between \( \Omega^a \) and \( \Omega^b \), between \( \Omega^a \) and \( \Omega^c \), and between \( \Omega^b \) and \( \Omega^c \)) as indicated in Eqs. (39)–(44). However, Three-material interface scheme 2 only makes use of the normal jump condition between \( \Omega^a \) and \( \Omega^b \). Based on the geometry, we choose the unit normal vector to be \( \hat{n} = (1, 0) \). We have

\[
\begin{align*}
\text{Table 3} & \quad \text{Errors for three-material interface schemes (Case 3(a), Case 3(b), Case 3(c) and Case 3(d)).}
\end{align*}
\]

| \( n_x \times n_y \) | Case 3(a) | | Case 3(b) |
|-----------------|-----------------|-----------------|
| \( L_x \) | Order | \( L_x \) | Order |
| 20 \times 20 | 5.172e-1 | 1.097e-1 | 5.246e-1 | 1.049e-1 |
| 40 \times 40 | 9.489e-2 | 2.45 | 1.996e-2 | 2.46 |
| 80 \times 80 | 2.084e-2 | 2.19 | 4.740e-3 | 2.07 |
| 160 \times 160 | 4.135e-3 | 2.13 | 1.074e-3 | 2.14 |

| \( n_x \times n_y \) | Case 3(c) | | Case 3(d) |
|-----------------|-----------------|-----------------|
| \( L_x \) | Order | \( L_x \) | Order |
| 20 \times 20 | 7.304e-4 | 1.179e-4 | 9.510e-4 | 2.127e-4 |
| 40 \times 40 | 9.681e-5 | 2.92 | 1.279e-5 | 3.20 |
| 80 \times 80 | 1.198e-5 | 3.01 | 3.360e-6 | 1.93 |
| 160 \times 160 | 3.598e-6 | 1.74 | 1.130e-6 | 1.57 |

\[
\begin{align*}
\text{Table 4} & \quad \text{Errors for three-material interface schemes (Case 3(e) and Case 3(f)).}
\end{align*}
\]

| \( n_x \times n_y \) | Case 3(e) | | Case 3(f) |
|-----------------|-----------------|-----------------|
| \( L_x \) | Order | \( L_x \) | Order |
| 20 \times 20 | 6.811e-1 | 1.665e-1 | 9.207e-3 | 3.445e-3 |
| 40 \times 40 | 1.702e-1 | 2.09 | 4.013e-2 | 2.05 |
| 80 \times 80 | 3.707e-2 | 2.20 | 7.240e-3 | 2.47 |
| 160 \times 160 | 6.408e-3 | 2.53 | 1.675e-3 | 2.11 |

| \( n_x \times n_y \) | Case 3(f) | | Case 3(f) |
|-----------------|-----------------|-----------------|
| \( L_x \) | Order | \( L_x \) | Order |
| 20 \times 20 | 6.811e-1 | 1.665e-1 | 9.207e-3 | 3.445e-3 |
| 40 \times 40 | 1.702e-1 | 2.09 | 4.013e-2 | 2.05 |
| 80 \times 80 | 3.707e-2 | 2.20 | 7.240e-3 | 2.47 |
| 160 \times 160 | 6.408e-3 | 2.53 | 1.675e-3 | 2.11 |
test different choices of the unit normal vector. For example, if we change it to $\vec{n} = \left(1/\sqrt{2}, 1/\sqrt{2}\right)$ for Case 3(a), the $L_\infty$ error will change from 0.5172 to 0.5177. Therefore, we conclude that our scheme works well for the singular point.

Case 3(a) is used to demonstrate that the proposed three-material interface schemes are stable for highly oscillatory solutions. In Case 3(b), we study the situation when the beta value in different subdomains is position dependent, and it may be discontinuous at the interfaces. From Case 3(c) to Case 3(f), we test the potential of our schemes for highly oscillatory solution and high coefficient contrast ratios. In Case 3(c), it is seen that for slowly oscillatory solutions, even when the coefficient discontinuous at the interfaces. From Case 3(c) to Case 3(f), we test the potential of our schemes for highly oscillatory solutions combined with high contrast coefficients. This is mainly due to the reason that in There-material scheme 2, the singular point on a grid node is treated as a point in $\Omega^2$, which decreases the accuracy when the solution and coefficient in $\Omega^2$ are challenging. For each case, computed $L_\infty$ and $L_2$ errors are given. The designed second order accuracy is observed for all the situations.

**Case 4.** In this case, one solves the 2D Poisson Eq. (34). Interfaces and subdomains are prescribed by two piecewise smooth level set functions to outline

$$\phi_1(x, y) = \begin{cases} (y - 0.4x - 0.2)(y - 0.4x + 0.2)(y + 0.4x - 0.2)(y + 0.4x + 0.2), & |x| \leq 0.5, |y| \leq 0.2; \\ -1, & \text{else.} \end{cases} \quad (a)$$

and

$$\phi_2(x, y) = \begin{cases} (y - 0.4x - 0.2)(y + 0.4x - 0.2), & y \geq 0.2; \\ (y + 0.4x + 0.2)(y - 0.4x + 0.2), & y \leq -0.2; \\ -(y + 0.4x + 0.2)(y - 0.4x - 0.2), & x \leq -0.5, |y| < 0.2; \\ -(y - 0.4x + 0.2)(y + 0.4x - 0.2), & x \geq 0.5, |y| < 0.2. \end{cases} \quad (b)$$

We employ the following discontinuous coefficients

$$\beta^0(x, y) = 1, \quad \beta^0(x, y) = 2, \quad \beta^0(x, y) = 3. \quad (109)$$

The solution at three different domains is designed as

$$u^0(x, y) = 6 + \sin(4\pi x) \sin(4\pi y), \quad u^1(x, y) = 6 + \sin(4\pi x) \sin(4\pi y), \quad u^2(x, y) = x^2 + y^2 + \sin(x + y). \quad (111)$$

Case 4 is designed to test the disassociation strategy of resolving fictitious values. The geometry of the problem is illustrated in Fig. 15. Two vertexes of $\Omega^2$ are treated as points in $\Omega^2$ and suitable disassociation schemes are used. Presented in Table 5, computed $L_\infty$ and $L_2$ errors indicate the second order accuracy.

**Case 5.** In this case, the 2D Poisson Eq. (34) is solved. We design two piecewise smooth level set functions as

$$\phi_1(x, y) = \begin{cases} -(y - \frac{1}{7}x + \frac{1}{6})(y + \frac{1}{7}x - \frac{2}{6}), & x \leq \frac{1}{7}; \\ -1, & \text{else}. \end{cases} \quad (a)$$

and

$$\phi_2(x, y) = \begin{cases} \frac{1}{2} \sin \left(\frac{\pi}{2}y\right), \quad \text{else.} \end{cases} \quad (b)$$

Fig. 15. Basic geometry on a $20 \times 20$ grid (Left) and the computed solution on a $80 \times 80$ grid (Right) for Case 4.
and
\[
\phi(x, y) = \begin{cases} 
-(y - \frac{3}{2}x + \frac{1}{12})(x + \frac{3}{2}y - \frac{1}{12}), & x \geq \frac{1}{2}; \\
-1, & \text{else}.
\end{cases} \tag{113}
\]

for interfaces and subdomains. Let us choose the discontinuous coefficients as
\[
\beta^a(x, y) = 1, \quad \beta^b(x, y) = 2, \quad \beta^c(x, y) = 3. \tag{114}
\]

The solution at three different domains is given by
\[
u^a(x, y) = 6 + \sin(2\pi x) \sin(2\pi y), \quad u^b(x, y) = 8 + \sin(x + y), \quad u^c(x, y) = x^2 + y^2. \tag{115}
\]

In this case, the performance of Three-material interface scheme 3 is tested. It is seen that, on a 20 \times 20 grid, the meshline between points (1, 1) and (2, 1) goes through three subdomains, as depicted in Fig. 16. Therefore, we need to employ the disassociation strategy and make use of four fictitious values together. Table 6 confirms the designed accuracy.

**Case 6.** In this case, we solve the 2D Poisson Eq. (34) with two piecewise smooth level set functions
\[
\phi_1(x, y) = \begin{cases} 
-(y - 4x^2 + \frac{1}{2})(y + 4x^2 - \frac{1}{2}), & |x| \leq \sqrt{\frac{1}{12}}; \\
-1, & \text{else}.
\end{cases} \tag{116}
\]

and
\[
\phi_2(x, y) = \begin{cases} 
-(y - 4x^2 + \frac{1}{2})(y + 4x^2 - \frac{1}{2}), & |x| \geq \sqrt{\frac{1}{12}}; \\
-1, & \text{else}.
\end{cases} \tag{117}
\]
it is difficult to find suitable auxiliary points to approximate 

As shown in Fig. 18, this case is used to test the critical angle of $2 \tan^{-1}(1/2)$ and the situations when beta value in different domains is position dependent and may be discontinuous at the interfaces. Table 8 lists computed $L_{\infty}$ and $L_2$ errors. Our results demonstrate the second order accuracy of the MIB method.

Table 7

<table>
<thead>
<tr>
<th>$n_x \times n_y$</th>
<th>$L_{\infty}$</th>
<th>Order</th>
<th>$L_2$</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 \times 20</td>
<td>3.786e-2</td>
<td></td>
<td>1.086e-2</td>
<td></td>
</tr>
<tr>
<td>40 \times 40</td>
<td>8.523e-3</td>
<td>2.15</td>
<td>2.824e-3</td>
<td>1.94</td>
</tr>
<tr>
<td>80 \times 80</td>
<td>2.083e-3</td>
<td>2.03</td>
<td>7.504e-4</td>
<td>1.91</td>
</tr>
<tr>
<td>160 \times 160</td>
<td>5.492e-4</td>
<td>1.92</td>
<td>2.014e-4</td>
<td>1.90</td>
</tr>
</tbody>
</table>

The domain geometry and interfaces are depicted Fig. 17. The discontinuous coefficients are prescribed by

$$
\beta^a(x, y) = 1, \quad \beta^b(x, y) = 2, \quad \beta^c(x, y) = 3. \tag{118}
$$

The solution at three different domains is designed as

$$
u^a(x, y) = 6 + \sin(2\pi x) \sin(2\pi y), \quad u^b(x, y) = 8 + \sin(2\pi x) \sin(2\pi y), \quad u^c(x, y) = x^2 + y^2. \tag{119}
$$

In this case, two singular points locate on the same meshline as shown in Fig. 17. Therefore, no special scheme is needed. But it is difficult to find suitable auxiliary points to approximate $u_y$. This case is also used to show that the challenging situation when two interfaces intersect with same meshline within a grid spacing can be avoid by choosing a suitable grid. However, more general situations are discussed in earlier test cases. Table 7 lists computed $L_{\infty}$ and $L_2$ errors. Obviously, the second order accuracy is achieved.

**Case 7.** In this case, we solve the 2D Poisson Eq. (34). Let us choose piecewise smooth level set functions as

$$
\phi_1(x, y) = \begin{cases} 
(y - 2x + \frac{1}{3}), & x < \frac{1}{3}; \\
0, & x = \frac{1}{3}; \\
-1, & y \geq \frac{1}{3}; \\
1, & \text{else.} 
\end{cases} \tag{120}
$$

and

$$
\phi_2(x, y) = \begin{cases} 
(y + x - \frac{2}{3}), & x > \frac{1}{3}; \\
0, & x = \frac{1}{3}; \\
-1, & y \geq \frac{1}{3}; \\
1, & \text{else.} 
\end{cases} \tag{121}
$$

The discontinuous coefficients are allowed to vary over different subdomains

$$
\beta^a(x, y) = 1 + \exp(x + y), \quad \beta^b(x, y) = 2 + \sin(x + y), \quad \beta^c(x, y) = 3 + x^2 + y^2. \tag{122}
$$

The solution at three different domains is designed as

$$
u^a(x, y) = 8 + \sin(\pi x) \sin(\pi y), \quad u^b(x, y) = 4 + \sin(2\pi x) \sin(2\pi y), \quad u^c(x, y) = x^2 + y^2 + \sin(\pi x) \sin(\pi y). \tag{123}
$$

In this case, two interfaces intersect with same meshline in a grid spacing. We have to use Three-material interface scheme 3. As shown in Fig. 18, this case is used to test the critical angle of $2 \tan^{-1}(1/2)$ and the situations when beta value in different domains is position dependent and may be discontinuous at the interfaces. Table 8 lists computed $L_{\infty}$ and $L_2$ errors. Our results demonstrate the second order accuracy of the MIB method.
Case 8. According to Grisvard [21], when the interface has geometric singularities, such as corners, tips, and wedges, the solution of the Poisson equation is usually singular too

\[ u(r, \theta) = r^{k \pi/\alpha} \sin(k \pi \theta/\alpha), \]  

where \( k \) is a wavenumber and \( \alpha \) is the angle parameter of the singular geometry. This solution diverges for certain \( k \) values as \( r \to 0 \).

To test the performance of the present MIB method for this class of problems, we use a pair of level set functions to define three-material domains as shown in Fig. 19

\[
\phi_1(x, y) = \begin{cases} 
-\left( y - \frac{x}{\sqrt{3}} \right) \left( y + \frac{x}{\sqrt{3}} \right) & x \leq 0; \quad (a) \\
-1 & \text{else.} \quad (b)
\end{cases}
\]

and

\[
\phi_2(x, y) = \begin{cases} 
-\left( y - \frac{x}{\sqrt{3}} \right) \left( y + \frac{x}{\sqrt{3}} \right) & x \geq 0; \quad (a) \\
-1 & \text{else.} \quad (b)
\end{cases}
\]
The discontinuous coefficients in different subdomains are
\[ \beta^a(x, y) = 1, \quad \beta^b(x, y) = 2, \quad \beta^c(x, y) = 3. \] (127)

We consider a solution that takes the form of Eq. (124)
\[ u^a(x, y) = 4 + \sin(2\pi x) \sin(2\pi y), \]
\[ u^b(r, \theta) = r^{\kappa/\omega} \sin(k \pi \theta / \omega), \] (128)
\[ u^c(x, y) = x^3 + y^3 + \sin(\pi x) \sin(\pi y), \]
where we choose \( k = \frac{2}{3} \) and \( \omega = \pi/3 \). Obviously, this solution has an unbound first order derivative. For this kind of problems, Galerkin formulations can handle it directly but collocation formulations cannot. A standard technique is to multiply the solution with an appropriate polynomial factor [66], which can be viewed as a transformation of the solution. This approach is also employed in the present work to deal with the singular solution at the geometric singularity.

We define a complex functions \( w \) in subdomain \( \Omega^i \) with real part,
\[ w_1(x, y) = r^2 \cos \left( \frac{8}{3} \theta \right), \] (129)
and imaginary part,
\[ w_2(x, y) = r^2 \sin \left( \frac{8}{3} \theta \right). \] (130)

Then the \( u^b \) can be calculated through the expression:
\[ u^b = \text{Im} \left( w^b \right). \] (131)

Functions \( w_1 \) and \( w_2 \) are harmonic functions. After taking care of the geometry singularity by the proposed MIB method, the solution \( u \) and \( w \) can be easily obtained. We list the computed \( L_\infty \) and \( L_2 \) errors of \( w_2 \) in Table 9. Clearly, the second order accuracy is achieved.

5. Concluding remarks

The present paper presents the first known second order method for solving two-dimensional (2D) elliptic partial differential equations with discontinuous coefficients resulted from modeling three-material interfaces. The matched interface and boundary (MIB) method is developed for this class of problems. We start from a completion of our earlier 2D MIB schemes for all possible geometric situations and topological variations. This completion lays some required technical foundations for us to construct new MIB schemes for three-material interface problems. Three new three-material interface schemes are proposed. We consider a number of topological variations in three-material settings. The validity of our new algorithms are extensively tested over a large number of test cases with various geometric singularities, material coefficients and solution types. The designed second order accuracy in both \( L_\infty \) and \( L_2 \) norms are confirmed in our numerical experiments.

Due to the enormous practical importance of multi-material interface problems, it is expected that there will be more attention to this class of problems. First, second order methods for solving elliptic PDEs with four- and five-material interfaces can be constructed at some special interface geometries. However, for general geometric singularities, it will be very difficult to design second order schemes. We expect that four- and five-material interfaces do not occur as frequently as three-material interfaces in practical applications. Additionally, for real-world applications, it is important to develop elliptic interface schemes for multi-material interfaces in 3D domains. This problem is under our consideration.

The remaining major challenges in the field are as follows. First, it is still enormously challenging to construct higher-order elliptic interface schemes in 1D, 2D and particularly in 3D domains. It is well known that stable and robust high order schemes are especially important to numerical efficiency. In 1D domains, schemes of orders up to sixteen were demonstrated in the MIB method [68,72]. No higher order schemes were reported to our knowledge. It appears that interface schemes of orders higher than 16 may not be numerically stable. This aspect needs to be investigated further. In 2D domains, schemes of orders up six have been reported for curved interfaces [72]. The MIB method may be able to provide even

<table>
<thead>
<tr>
<th>( n_x \times n_y )</th>
<th>( L_\infty )</th>
<th>Order</th>
<th>( L_2 )</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 \times 20</td>
<td>1.646e-1</td>
<td></td>
<td></td>
<td>2.536e-2</td>
</tr>
<tr>
<td>40 \times 40</td>
<td>4.602e-2</td>
<td>2.02</td>
<td>6.217e-3</td>
<td>2.03</td>
</tr>
<tr>
<td>80 \times 80</td>
<td>1.057e-2</td>
<td>1.94</td>
<td>1.858e-3</td>
<td>1.74</td>
</tr>
<tr>
<td>160 \times 160</td>
<td>2.781e-3</td>
<td>1.93</td>
<td>5.038e-4</td>
<td>1.88</td>
</tr>
</tbody>
</table>
higher-order 2D elliptic schemes for some interfaces with relatively small curvatures, however, no such result has been constructed yet. We expect that it will be extremely difficult, if not impossible, to construct 6th order schemes for 2D elliptic equations with arbitrarily curved interfaces. In 3D domains, the highest order elliptic interface schemes constructed so far is the 6th order MIB method for some special geometries, i.e., spherical and ellipsoid interfaces [65], to our knowledge. The construction of elliptic interface methods of orders higher than six for 3D arbitrarily curved smooth interfaces is still a challenging open problem. We are not sure whether such a scheme is feasible numerically. If it is indeed feasible, it is interesting to know whether it is stable. As most real-world applications are in 3D domains, it is essentially important to make progress in this open problem.

Acknowledgments

This work was supported in part by NSF grant CCF-0936830, NIH grant R01GM-090208 and MSU Competitive Discretionary Funding Program grant 91–4600. The authors thank Duan Chen for useful discussions.

References
