Published in final edited form as:
J Comput Phys. 2011 June 1; 230(12): 4588–4615. doi:10.1016/j.jcp.2011.02.037.

MIB method for elliptic equations with multi-material interfaces

Kelin Xia[1,2], **Meng Zhan**[1], and **Guo-Wei Wei**[2,3,*]

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

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

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

Abstract

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.

Keywords

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

1 Introduction

Mathematical modeling of material interfaces often leads to elliptic partial differential equations (PDEs) with discontinuous coefficients and singular sources. Solutions to such

*Corresponding author. Tel: (517)353 4689, Fax: (517)432 1562, wei@math.msu.edu.

PDEs have attracted much attention [2, 4, 5, 11, 14, 19, 27, 25, 28, 29, 31, 32, 34, 37, 39, 40, 42, 45, 56, 57, 58, 59, 50, 62, 6, 3] since Peskin pioneered the immersed boundary method (IBM) in 1977 [20, 35, 53, 54, 52]. Interface techniques have demonstrated an increasing importance in a number of areas, including fluid dynamics [12, 15, 16, 30, 49, 37, 39, 36], electromagnetic wave propagation [22, 23, 33, 34, 68, 67], materials science, [24, 27] and biological systems [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 [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 and coworkers [13, 44], finite-volume-based methods [51], integral equation methods [46, 47, 48], and the piecewise-polynomial discretization (PPD) by Chen and Strain [9].

We have recently proposed a highly accurate algorithm, the matched interface and boundary (MIB) method [66, 65, 68, 72, 71] for solving elliptic equations with discontinuous coefficients and singular sources. The essential ideas of the current MIB method were introduced in an earlier interface scheme for solving Maxwell's equation [68]. The MIB method is of arbitrarily high-order accuracy in principle. The 16th order accurate MIB scheme has been demonstrated with a simple interface geometry [68, 72] and sixth-order accurate MIB schemes have been constructed for curved interfaces in two-dimensional (2D) and three-dimensional (3D) domains [65, 72].

The main features of the MIB method are as follows. First, whenever possible, it employs a dimensional splitting strategy inherited from our earlier discrete singular convolution algorithm [60, 61] to reduce 2D and 3D interface problems into 1D ones. This strategy significantly reduces the topological complexity in handling 3D interface geometries. Additionally, the MIB method enforces only lowest order interface jump conditions (i.e., interface continuity conditions). In order to achieve higher order accuracy, which requires many more jump conditions, the set of lowest order interface jump conditions is iteratively utilized in the MIB method [68]. This approach avoids the numerical instability and accuracy reduction in implementing higher order jump conditions. Moreover, in our MIB method, as well as our earlier discrete singular convolution algorithm [60, 61], fictitious points are created so that a uniform discretization scheme can be directly employed throughout the domain, including in the vicinity of the interface and singular source. This approach avoids the confusion in determining the discretization scheme in the vicinity of the interface when the geometry is very complex, as the case in many biomolecular applications [70, 64, 17, 7]. Furthermore, because of the use of fictitious points, enforcing the jump conditions is separated 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].

Independently, Zhao has developed impressive MIB schemes for the Helmholtz equation [67]. Note that interface conditions in this problem are more complex. Zhao's MIB schemes achieve fourth order accuracy for these equations with arbitrarily curved interfaces [67]. High order schemes are particularly valuable for electromagnetic wave scattering and propagation due to their nature of involving high frequencies.

In the past, much attention was paid to the construction of elliptic interface methods to ensure the symmetric and/or diagonally dominant matrices due to the concerns of the condition number and the speed of convergence of iterative matrix solvers [65]. These concerns have become essentially unnecessary, thank to the development of Krylov-accelerated multigrid methods [9] and Krylov subspace techniques [7]. The non-symmetric

and non-diagonally dominant matrices resulting from finite difference and finite volume types of elliptic interface discretizations can be easily solved with the state of the art Krylov subspace techniques. 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 time-dependent problems, matrix properties are still important [68, 69].

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, 66, 65, 64]. Nonsmooth interfaces are also called geometric singularities. This class of problems are 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 nonsmooth 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, 7, 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, 7, 18], is able to derive second order accuracy in solving the Poisson-Boltzmann equation with primitive molecular surfaces 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 briey reported elsewhere [63]. This class of problems

is referred to as multi-material interface 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 $\Omega \in \mathbb{R}^2$. A given interface Γ divides Ω into two subdomains, Ω^a and Ω^b, hence $\Omega = \Omega^a \cup \Omega^b$ and $\Gamma = \Omega^a \cap \Omega^b$. We assumed that the boundary $\partial\Omega$ and interfaces Γ are Lipschitz continuous and there is a piecewise smooth level-set function Φ on $\bar{\Omega}$, 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

$$(\beta(x,y)u_x(x,y))_x+(\beta(x,y)u_y(x,y))_y=g(x,y), \tag{1}$$

with variable coefficients $\beta(x, y)$, which may have jumps at the interface Γ. To make the problem well defined, there are two given jump conditions associated with the interface, i.e.,

$$[u]=u^a-u^b=\Psi_1, \tag{2}$$

$$[\beta u_n]=\beta^a u_n^a-\beta^b u_n^b=\Psi_2, \tag{3}$$

where superscripts a and b denote the limiting value of a function from Ω^a and Ω^b sides of the interface Γ 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, $[u_\tau]=u_\tau^a-u_\tau^b$. Hence for a point on the interface, if we define unit normal vector $\vec{n}=(\cos\theta, \sin\theta)$, the unit tangential vector is $\vec{\tau}=(-\sin\theta, \cos\theta)$. Thus, we have three jump conditions

$$[u]=u^a-u^b, \tag{4}$$

$$[u_\tau]=(-u_x^a\sin\theta+u_y^a\cos\theta)-(-u_x^b\sin\theta+u_y^b\cos\theta), \tag{5}$$

$$[\beta u_n]=\beta^a(u_x^a\cos\theta+u_y^a\sin\theta)-\beta^b(u_x^b\cos\theta+u_y^b\sin\theta). \tag{6}$$

In the MIB approach, the implementation of jump conditions is disassociated from the discretization of the elliptic equation. Here $(\beta u_x)_x$ and $(\beta u_y)_y$ are discretized separately with the second order central finite difference scheme [71]. Therefore, a direct calculation of $(\beta u_x)_x$ 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 jth mesh line at a point between (i, j) and $(i + 1, j)$, the $u_{i,j}$ and $u_{i+1,j}$ 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 $u_{i,j}$ side. To estimate the fictitious value, we enforce the jump conditions by discretizing some of u^a, u^b, u_x^a, u_x^b, u_y^a, and u_y^b based on the geometry. We can then construct two linear equations involving two fictitious values.

It is always possible in practice to find out two fictitious values in a 2D domain. If u_y^b is difficult to evaluate, we can eliminate it from Eq. (5) and Eq. (6) to attain

$$[u]=u^a-u^b, \quad \text{and} \quad [\beta u_n]-\beta^b\tan\theta[u_\tau]=C_x^a u_x^a-C_x^b u_x^b+C_y^a u_y^a, \tag{7}$$

where $C_x^a=\beta^a \cos\theta+\beta^b \tan\theta \sin\theta$, $C_x^b=\beta^b \cos\theta+\beta^b \tan\theta \sin\theta$ and $C_y^a=\beta^a \sin\theta-\beta^b \sin\theta$. Based on the geometry, we can choose to eliminate one of u_y^a, u_x^b and u_x^a and attain two linear equations. If one eliminates u_y^a,

$$[u]=u^a-u^b, \quad \text{and} \quad [\beta u_n]-\beta^a \tan\theta[u_\tau]=C_x^a u_x^a-C_x^b u_x^b-C_y^b u_y^b, \tag{8}$$

where $C_x^a=\beta^a \cos\theta+\beta^a \tan\theta \sin\theta$, $C_x^b=\beta^b \cos\theta+\beta^a \tan\theta \sin\theta$ and $C_y^b=\beta^b \sin\theta-\beta^a \sin\theta$. If one eliminates u_x^b,

$$[u]=u^a-u^b, \quad \text{and} \quad [\beta u_n]+\beta^b \cot\theta[u_\tau]=C_x^a u_x^a+C_y^a u_y^a-C_y^b u_y^b, \tag{9}$$

where $C_x^a=(\beta^a-\beta^b)\cos\theta$, $C_y^a=\beta^b \cos\theta \cot\theta+\beta^a \sin\theta$ and $C_y^b=\beta^b(\cos\theta \cot\theta+\sin\theta)$. If one eliminates u_x^a,

$$[u]=u^a-u^b, \quad \text{and} \quad [\beta u_n]+\beta^a \cot\theta[u_\tau]=C_x^b u_x^b+C_y^a u_y^a-C_y^b u_y^b, \tag{10}$$

where $C_x^b=(\beta^a-\beta^b)\cos\theta$, $C_y^b=\beta^a \cos\theta \cot\theta+\beta^b \sin\theta$ and $C_y^a=\beta^a(\cos\theta \cot\theta+\sin\theta)$.

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 $u_{i,j}$ and $u_{i+1,j}$ are in different regions, two fictitious values $f_{i,j}$ and $f_{i+1,j}$ and two auxiliary values $u_{i-1,j}$ and $u_{i+2,j}$ are needed to construct the central difference scheme at point (i, j), which involve values $u_{i-1,j}$, $u_{i,j}$ and $f_{i+1,j}$. Similarly, at point $(i + 1, j)$, $f_{i,j}$, $u_{i+1,j}$ and $u_{i+2,j}$ 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 $u_{i+1,j}$ and $u_{i,j}$ are in the same domain. In order to overcome this geometric restriction, we replace $u_{i+1,j}$ 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 Ω^a, and Ω^b. In this paper, Ω^a is marked with yellow color, while Ω^b is marked with green color. To make it clear, we denote u^{ao} the limiting value of function u at point o from the Ω^a, and u^{bo} the limiting value at point o from Ω^b. The derivative of u with respect to x at point o from Ω^a is represented as u_x^{ao}, and from domain b as u_x^{bo}. We use a set of similar notations for point o'. For example, $u^{ao'}$ stands for the limiting value of function u from Ω^a at point o'.

Therefore, for point o, using $u_{i-2,j}$, $u_{i-1,j}$, $u_{i,j}$, $u_{i+1,j}$ and the fictitious values $f_{i-1,j}$ $f_{i,j}$ and $f_{i+1,j}$, we can discretize u^{ao}, u^{bo}, u_x^{ao}, u_x^{bo} explicitly as follows

$$\begin{aligned} u^{ao} &= (w^{ao}_{0,i-1}, w^{ao}_{0,i}, w^{ao}_{0,i+1}) \cdot (f_{i-1,j}, u_{i,j}, f_{i+1,j})^T, \\ u^{bo} &= (w^{bo}_{0,i-2}, w^{bo}_{0,i-1}, w^{bo}_{0,i}) \cdot (u_{i-2,j}, u_{i-1,j}, f_{i,j})^T, \\ u_x^{ao} &= (w^{ao}_{1,i-1}, w^{ao}_{1,i}, w^{ao}_{1,i+1}) \cdot (f_{i-1,j}, u_{i,j}, f_{i+1,j})^T, \\ u_x^{bo} &= (w^{bo}_{1,i-2}, w^{bo}_{1,i-1}, w^{bo}_{1,i}) \cdot (u_{i-2,j}, u_{i-1,j}, f_{i,j})^T, \end{aligned} \tag{11}$$

where $w^l_{n,m}$ 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, ao stands for the limiting value at point o from Ω^a.

According to Fig. 1, it is difficult for us to find suitable auxiliary points to discretize u_y^{ao} as there are not enough points in Ω^a around point o. We choose to discretize u_y^{bo}. In order to do so, we need three u values along the y-direction in Ω^b. Here we denote the coordinate of point o as (i_o, j). We can choose these three values at point (i_o, j), $(i_o, j + 1)$ and $(i_o, j + 2)$. Having already obtained u^{bo}, the other two values can be approximated by auxiliary values $u_{i-3,j+1}$, $u_{i-2,j+1}$, $u_{i-1,j+1}$, $u_{i-2,j+2}$, $u_{i-1,j+2}$ and $u_{i,j+2}$. Here, u_y^{bo} can be represented as

$$\begin{aligned} u_y^{bo} = &\begin{bmatrix} w^{bo}_{1,j} \\ w^{bo}_{1,j+1} \\ w^{bo}_{1,j+2} \end{bmatrix}^T \cdot \begin{bmatrix} w^{bo}_{0,i-2} & w^{bo}_{0,i-1} & w^{bo}_{0,i} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w1^{bo}_{0,i-3} & w1^{bo}_{0,i-2} & w1^{bo}_{0,i-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w2^{bo}_{0,i-2} & w2^{bo}_{0,i-1} & w2^{bo}_{0,i} \end{bmatrix} \\ &\times [u_{i-2,j}, u_{i-1,j}, f_{i,j}, u_{i-3,j+1}, u_{i-2,j+1}, u_{i-1,j+1}, u_{i-2,j+2}, u_{i-1,j+2}, u_{i,j+2}]^T, \end{aligned} \tag{12}$$

where, $w1$ and $w2$ are finite difference coefficients for the points at two auxiliary lines. Based on the local geometry, we eliminate u_y^{ao}, and two equations can be attained by substituting Eqs. (11) and (12) into Eq. (8)

$$[u]_o=(w_{0,i-1}^{ao}, w_{0,i}^{ao}, w_{0,i+1}^{ao})\cdot(f_{i-1,j}, u_{i,j}, f_{i+1,j})^T-(w_{0,i-2}^{bo}, w_{0,i-1}^{bo}, w_{0,i}^{bo})\cdot(u_{i-2,j}, u_{i-1,j}, f_{i,j})^T, \tag{13}$$

$$[\beta u_n]_o-\beta^a\tan\theta[u_\tau]_o=C_x^a\begin{bmatrix} w_{1,i-1}^{ao} \\ w_{1,i}^{ao} \\ w_{1,i+1}^{ao}\end{bmatrix}^T\cdot\begin{bmatrix} f_{i-1,j} \\ u_{i,j} \\ f_{i+1,j}\end{bmatrix}-C_x^b\begin{bmatrix} w_{1,i-2}^{bo} \\ w_{1,i-1}^{bo} \\ w_{1,i}^{bo}\end{bmatrix}^T\cdot\begin{bmatrix} u_{i-2,j} \\ u_{i-1,j} \\ f_{i,j}\end{bmatrix}-C_y^b\begin{bmatrix} w_{1,j} \\ w_{1,j+1} \\ w_{1,j+2}\end{bmatrix}^T \tag{14}$$

$$\cdot\begin{bmatrix} w_{0,i-2}^{bo} & w_{0,i-1}^{bo} & w_{0,i}^{bo} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w1_{0,i-3}^{bo} & w1_{0,i-2}^{bo} & w1_{0,i-1}^{bo} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w2_{0,i-2}^{bo} & w2_{0,i-1}^{bo} & w2_{0,i}^{bo}\end{bmatrix}\cdot[u_{i-2,j}, u_{i-1,j}, f_{i,j}, u_{i-3,j+1}, u_{i-2,j+1}, u_{i-1,j+1}, u_{i-2,j+2}, u_{i-1,j+2}$$

(15)

Here the unit normal vector is $\vec{n}$ = (cos θ, sin θ), $C_x^a=\beta^a\cos\theta+\beta^a\tan\theta\sin\theta$, $C_x^b=\beta^b\cos\theta+\beta^a\tan\theta\sin\theta$ and $C_y^b=\beta^b\sin\theta-\beta^a\sin\theta$. 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 $u^{ao'}$ and $u^{bo'}$, and enforce Eq. (4) as follows

$$\begin{aligned} u^{ao'}&=(w_{0,i-1}^{ao'}, w_{0,i}^{ao'}, w_{0,i+1}^{ao'})\cdot(f_{i-1,j}, u_{i,j}, f_{i+1,j})^T, \\ u^{bo'}&=(w_{0,i-1}^{bo'}, w_{0,i}^{bo'}, w_{0,i+1}^{bo'})\cdot(u_{i-1,j}, f_{i,j}, u_{i+1,j})^T. \end{aligned} \tag{16}$$

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

$$[u]_{o'}=(w_{0,i-1}^{ao'}, w_{0,i}^{ao'}, w_{0,i+1}^{ao'})\cdot(f_{i-1,j}, u_{i,j}, f_{i+1,j})^T-(w_{0,i-1}^{bo'}, w_{0,i}^{bo'}, w_{0,i+1}^{bo'})\cdot(u_{i-1,j}, f_{i,j}, u_{i+1,j})^T. \tag{17}$$

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

$$\begin{bmatrix} f_{i-1,j} \\ f_{i,j} \\ f_{i+1,j}\end{bmatrix}=\{C\}_{3\times14}\cdot\{U\}_{14\times1}, \tag{18}$$

where vector $\{U\}_{14\times1}$ consists of 10 function values and 4 jump conditions as follows

$$\{U\}_{14\times1}=(u_{i-2,j}, u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i-3,j+1}, u_{i-2,j+1}, u_{i-1,j+1}, u_{i-2,j+2}, u_{i-1,j+2}, u_{i,j+2}, [u]_o, [u]_{o'}, [\beta u_n]_o, [u_\tau]_o)^T. \quad (19)$$

Here, $\{C\}_{3\times14}$ 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 $(\beta u_x)_x$ at irregular points $(i-1, j)$, (i, j) and $(i+1, j)$ as follows

$$\begin{aligned}
(\beta u_x)_x &= \tfrac{1}{\Delta x^2}(\beta^b_{i-\frac{3}{2},j}, -\beta^b_{i-\frac{3}{2},j} - \beta^b_{i+\frac{1}{2},j}, \beta^b_{i+\frac{1}{2},j}) \cdot (u_{i-2,j}, u_{i-1,j}, f_{i,j})^T \quad \text{at } (i-1, j),\\
(\beta u_x)_x &= \tfrac{1}{\Delta x^2}(\beta^a_{i-\frac{1}{2},j}, -\beta^a_{i-\frac{1}{2},j} - \beta^a_{i+\frac{1}{2},j}, \beta^a_{i+\frac{1}{2},j}) \cdot (f_{i-1,j}, u_{i,j}, f_{i+1,j})^T \quad \text{at } (i, j),\\
(\beta u_x)_x &= \tfrac{1}{\Delta x^2}(\beta^b_{i+\frac{1}{2},j}, -\beta^b_{i+\frac{1}{2},j} - \beta^b_{i+\frac{3}{2},j}, \beta^b_{i+\frac{3}{2},j}) \cdot (f_{i,j}, u_{i+1,j}, u_{i+2,j})^T \quad \text{at } (i+1, j).
\end{aligned} \quad (20)$$

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 $u^{ao'}$ and $u^{bo'}$. We can however discretize $u_x^{ao'}$ and $u_x^{bo'}$, and find suitable auxiliary points to approximate $u_y^{ao'}$ or $u_y^{bo'}$ 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 discretize the derivatives, and to construct equations to solve fictitious values.

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

$$[u_\tau]=(-\sin\theta, \cos\theta, \sin\theta, -\cos\theta) \cdot (u_x^a, u_y^a, u_x^b, u_y^b)^T, \quad (21)$$

$$[\beta u_n]=(\beta^a\cos\theta,\beta^a\sin\theta,-\beta^b\cos\theta,-\beta^b\sin\theta)\cdot(u_x^a,u_y^a,u_x^b,u_y^b)^T. \tag{22}$$

For node point on the interface, if we regard it as in Ω^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_x^a, u_x^b, u_y^a and u_y^b 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 Ω^b, and all the red points belong to Ω^a. Here for the situation in Fig. 3 (On-interface scheme 4), the on-interface grid node is treated as a Ω^b point. With the relation $u_{i,j}^a=u_{i,j}^b+[u]$, we can discretize three derivatives as

$$u_x^b=(w_{1,i-1},w_{1,i},w_{1,i+1})\cdot(u_{i-1,j},u_{i,j},u_{i+1,j})^T, \tag{23}$$

$$u_y^a=(w_{1,j-2},w_{1,j},w_{1,j+1})\cdot(u_{i,j-2},u_{i,j}+[u],u_{i,j+1})^T, \tag{24}$$

$$u_y^b=(w'_{1,j-1},w'_{1,j},w'_{1,j+1})\cdot(u_{i,j-1},u_{i,j},f_{i,j+1})^T. \tag{25}$$

Substituting above relations into Eq. (10), we can easily calculate $f_{i,j+1}$.

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 Ω^b, and note that $u_{i,j}^a=u_{i,j}^b+[u]$. The two fictitious values are $f_{i-1,j}$ and $f_{i,j-1}$. Four derivatives can be discretized as

$$u_x^a=(w_{1,i-1},w_{1,i},w_{1,i+2})\cdot(u_{i-1,j},u_{i,j}+[u],u_{i+2,j})^T, \tag{26}$$

$$u_x^b=(w'_{1,i-1},w'_{1,i},w'_{1,i+2})\cdot(f_{i-1,j},u_{i,j},u_{i+2,j})^T, \tag{27}$$

$$u_y^a=(w_{1,j-2},w_{1,j-1},w_{1,j})\cdot(u_{i,j-2},u_{i,j-1},u_{i,j}+[u])^T, \tag{28}$$

$$u_y^b=(w'_{1,j-1}, w'_{1,j}, w'_{1,j+1})\cdot(f_{i,j-1}, u_{i,j}, u_{i,j+1})^T. \tag{29}$$

By substituting these equations into Eq. (21) and Eq. (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 Ω^a, and use the relation $u^b_{i,j}=u^a_{i,j}-[u]$. Two fictitious values are $f_{i-1,j}$ and $f_{i,j+1}$. We obtain the explicit discretization equations of four derivatives as

$$u_x^a=(w_{1,i-1}, w_{1,i}, w_{1,i+1})\cdot(f_{i-1,j}, u_{i,j}, u_{i+1,j})^T, \tag{30}$$

$$u_x^b=(w'_{1,i-1}, w'_{1,i}, w'_{1,i+2})\cdot(u_{i-1,j}, u_{i,j}-[u], u_{i+2,j})^T, \tag{31}$$

$$u_y^a=(w_{1,j-1}, w_{1,j}, w_{1,j+1})\cdot(u_{i,j-1}, u_{i,j}, u_{i,j+1})^T, \tag{32}$$

$$u_y^b=(w'_{1,j-2}, w'_{1,j}, w'_{1,j+1})\cdot(u_{i,j-2}, u_{i,j}-[u], f_{i,j+1})^T. \tag{33}$$

We compute $f_{i-1,j}$ and $f_{i,j+1}$ by substituting above equations into Eq. (21) and Eq. (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 Ω^a as inside the interface and in Ω^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 $\frac{1}{2}\pi, \pi$ or $\frac{3}{2}\pi$. 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 gird point pattern can be solved by using one of

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 gird point pattern cannot be solved by using one of 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 Figs. 4(A). Then, depending on the interface morphology, there can be three different topological variations as shown in Fig. 4(A1) – 4(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 Figs. 4(B1) – 4(B3). Among these three variations, the cases of Figs. 4(B1) and Fig. 4(B3) can be 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 Figs. 4(C1) – 4(C3). Among them, the case of Fig. 4(C1) can be solved by using one of the aforementioned six on-interface schemes. Moreover, the case of 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 Figs. 4(D1) – 4(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 Figs. 5(E1) – 5(E6). The pattern in Fig. 5(E1) can be solved without the use of an interface scheme. Cases in Figs. 5(E2), 5(E4) and 5(E6) can be easily solved by using the aforementioned six on-interface schemes. The other two cases, Figs. 5(E3) and 5(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 Figs. 6. Interface schemes are required to resolve the first three distinguished situations. Cases in Figs. 6(F1) and 6(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 Γ^1 and Γ^2 to divide the open bounded domain $\Omega \in \mathbb{R}^2$ into three subdomains Ω^a, Ω^b and Ω^c, such that $\Omega = \Omega^a \cup \Omega^b \cup \Omega^c$. We assumed that the boundary $\partial\Omega$ and interfaces Γ^1 and Γ^2 are Lipschitz continuous and there are two piecewise smooth level-set functions Φ_1 and Φ_2 on Ω, such that $\Gamma^1 = \{x, y | \Phi_1 = 0; x, y \in \Omega\}$, $\Gamma^2 = \{x, y | \Phi_2 = 0; x, y \in \Omega\}$, $\Omega^a = \{x, y | \Phi_1 \geq 0; x, y \in \Omega\}$, $\Omega^b = \{x, y | \Phi_2 \geq 0; x, y \in \Omega\}$ and $\Omega^c = \{x, y | \Phi_1 \leq 0; \Phi_2 \leq 0; x, y \in \Omega\}$. We seek the solution of the 2D elliptical equation with variable coefficient $\beta(x, y)$

$$(\beta(x,y)u_x(x,y))_x+(\beta(x,y)u_y(x,y))_y=g(x,y), \tag{34}$$

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

Category 1. The points in this category belong to those parts of Γ^1 that do not intersect or touch Γ^2. These points are denoted as o_1. Then $\exists\ \varepsilon_1 > 0$, $\forall\ \varepsilon < \varepsilon_1$, and $\varepsilon > 0$, there exists an open disk $O(o_1, \varepsilon)$, such that $O(o_1, \varepsilon) \in \Omega^a \cup \Omega^c$. Jump conditions are given as

$$[u]_{o_1}=u^a - u^c, \tag{35}$$

$$[\beta u_n]_{o_1}=\beta^a u_n^a - \beta^c u_n^c. \tag{36}$$

Here, superscripts, a and c, denote the limiting values of a function at point o_1 from Ω^a and Ω^c, respectively.

Category 2. The points in this category locate in the parts of Γ^1 that overlap with Γ^2 and do not include the critical initial points where Γ^1 just intersects Γ^2. These points are denoted as o_2. Then $\exists\ \varepsilon_1 > 0$, $\forall\ \varepsilon < \varepsilon_1$, and $\varepsilon > 0$, there is a disk $O(o_2, \varepsilon)$, such that $O(o_2, \varepsilon) \in \Omega^a \cup \Omega^b$. Jump conditions are given as

$$[u]_{o_2}=u^a - u^b, \tag{37}$$

$$[\beta u_n]_{o_2}=\beta^a u_n^a - \beta^b u_n^b. \tag{38}$$

Category 3. The points in this category are the initial contacting points when Γ^1 and Γ^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 $\forall\ \varepsilon > 0$, $\exists\ o_i \in \Omega^i$, $(i = a, b, c)$, such that $o_i \in O(o_3, \varepsilon)$. For points in Category 3, we may have six jump conditions, they define the interface relations between Ω^a and Ω^b, between Ω^a and Ω^c, and between Ω^b and Ω^c. To make it clear, we use notation $[u]_{o_3}^{ij}, [\beta u_n]_{o_3}^{ij} (i \neq j; i, j = a, b, c)$. Jump conditions are expressed as

$$[u]_{o_3}^{ab}=u^a - u^b, \tag{39}$$

$$[u]_{o_3}^{ac}=u^a-u^c, \tag{40}$$

$$[u]_{o_3}^{bc}=u^b-u^c, \tag{41}$$

$$[\beta u_{n_1}]_{o_3}^{ab}=\beta^a u_{n_1}^a-\beta^b u_{n_1}^b, \tag{42}$$

$$[\beta u_{n_2}]_{o_3}^{ac}=\beta^a u_{n_2}^a-\beta^c u_{n_2}^c, \tag{43}$$

$$[\beta u_{n_3}]_{o_3}^{bc}=\beta^b u_{n_3}^b-\beta^c u_{n_3}^c. \tag{44}$$

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 $[u]^{ij} = u^i - u^j$ 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 $\tau = (-\sin\theta, \cos\theta)$. We can have the third set of jump conditions

$$[u_\tau]^{ij}=(-u_x^i\sin\theta+u_y^i\cos\theta)-(-u_x^j\sin\theta+u_y^j\cos\theta). \tag{45}$$

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

In this work, Ω^a, Ω^b, Ω^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 Ω^c and the adjacent points $(i-1, j)$ and $(i+1, j)$ locate in another two different domains, Ω^a and Ω^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 $f_{i-1,j}^{co}$ $f_{i,j}^{ao}$ $f_{i,j}^{bo'}$ and $f_{i+1,j}^{co'}$. The superscripts here denote the

directions of the extension. For example, $f^{co}_{i-1,j}$ denotes the extension of the function value from Ω^c through point o to point $(i-1, j)$ and $f^{bo'}_{i,j}$ denotes extension of the function value from Ω^b through point o' to point (i, j). Here $f^{ao}_{i,j}$ and $f^{bo'}_{i,j}$ 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 $u_{i-2,j}$, $u_{i-1,j}$, $u_{i,j}$ and fictitious values $f^{co}_{i-1,j}$, $f^{ao}_{i,j}$ and $f^{co'}_{i+1,j}$, we can discretize the u^{ao}, u^{co}, u^{ao}_x and u^{co}_x explicitly as

$$\begin{aligned} u^{ao} &= (w^{ao}_{0,i-2}, w^{ao}_{0,i-1}, w^{ao}_{0,i}) \cdot (u_{i-2,j}, u_{i-1,j}, f^{ao}_{i,j})^T, \\ u^{co} &= (w^{co}_{0,i-1}, w^{co}_{0,i}, w^{co}_{0,i+1}) \cdot (f^{co}_{i-1,j}, u_{i,j}, f^{co'}_{i+1,j})^T, \\ u^{ao}_x &= (w^{ao}_{1,i-2}, w^{ao}_{1,i-1}, w^{ao}_{1,i}) \cdot (u_{i-2,j}, u_{i-1,j}, f^{ao}_{i,j})^T, \\ u^{co}_x &= (w^{co}_{1,i-1}, w^{co}_{1,i}, w^{co}_{1,i+1}) \cdot (f^{co}_{i-1,j}, u_{i,j}, f^{co'}_{i+1,j})^T. \end{aligned} \tag{46}$$

According to Fig. 7, u^{ao}_y is to be discretized. We need six auxiliary points $u_{i-3,j+1}$, $u_{i-2,j+1}$, $u_{i-1,j+1}$, $u_{i-3,j+2}$, $u_{i-2,j+2}$ and $u_{i-1,j+2}$. The value of u^{ao}_y can be represented as

$$\begin{aligned} u^{ao}_y = &\begin{bmatrix} w^{ao}_{1,j} \\ w^{ao}_{1,j+1} \\ w^{ao}_{1,j+2} \end{bmatrix}^T \cdot \begin{bmatrix} w^{ao}_{0,i-2} & w^{ao}_{0,i-1} & w^{ao}_{0,i} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w1^{ao}_{0,i-3} & w1^{ao}_{0,i-2} & w1^{ao}_{0,i-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w2^{ao}_{0,i-3} & w2^{ao}_{0,i-2} & w2^{ao}_{0,i-1} \end{bmatrix} \\ &\times [u_{i-2,j}, u_{i-1,j}, f^{ao}_{i,j}, u_{i-3,j+1}, u_{i-2,j+1}, u_{i-1,j+1}, u_{i-3,j+2}, u_{i-2,j+2}, u_{i-1,j+2}]^T. \end{aligned} \tag{47}$$

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

$$\begin{aligned} [u]_o = &(w^{ao}_{0,i-2}, \\ &w^{ao}_{0,i-1}, w^{ao}_{0,i}) \cdot (u_{i-2,j}, u_{i-1,j}, f^{ao}_{i,j})^T - (w^{co}_{0,i-1}, \\ &w^{co}_{0,i}, w^{co}_{0,i+1}) \cdot (f^{co}_{i-1,j}, u_{i,j}, f^{co'}_{i+1,j})^T, \ [\beta u_n]_o \\ &- \beta^a \tan\theta_o [u_\tau]_o \\ = &C^{ao}_x \begin{bmatrix} w^{ao}_{1,i-2} \\ w^{ao}_{1,i-1} \\ w^{ao}_{1,i} \end{bmatrix}^T \cdot \begin{bmatrix} u_{i-2,j} \\ u_{i-1,j} \\ f^{ao}_{i,j} \end{bmatrix} \\ &- C^{co}_x \begin{bmatrix} w^{co}_{1,i-1} \\ w^{co}_{1,i} \\ w^{co}_{1,i+1} \end{bmatrix}^T \cdot \begin{bmatrix} f^{co}_{i-1,j} \\ u_{i,j} \\ f^{co'}_{i+1,j} \end{bmatrix} \\ &+ C^{ao}_y \begin{bmatrix} w_{1,j} \\ w_{1,j+1} \\ w_{1,j+2} \end{bmatrix}^T \end{aligned} \tag{48}$$

$$\cdot\begin{bmatrix} w^{ao}_{0,i-2} & w^{ao}_{0,i-1} & w^{ao}_{0,i} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w1^{ao}_{0,i-3} & w1^{ao}_{0,i-2} & w1^{ao}_{0,i-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w2^{ao}_{0,i-3} & w2^{ao}_{0,i-2} & w2^{ao}_{0,i-1} \end{bmatrix}\cdot[u_{i-2,j}, u_{i-1,j}, f^{ao}_{i,j}, u_{i-3,j+1}, u_{i-2,j+1}, u_{i-1,j+1}, u_{i-3,j+2}, u_{i-2} \tag{49}$$

Here the unit normal vector at point o is $\vec{n} = (\cos\theta_o, \sin\theta_o)$, $C^{ao}_x = \beta^a \cos\theta_o + \beta^c \tan\theta_o \sin\theta_o$, $C^{co}_x = \beta^c \cos\theta_o + \beta^c \tan\theta_o \sin\theta_o$ and $C^{ao}_y = \beta^a \sin\theta_o - \beta^c \sin\theta_o$. 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 $u^{bo'}$, $u^{co'}$, $u^{bo'}_x$ and $u^{co'}_x$ explicitly as

$$\begin{aligned} u^{bo'} &= (w^{bo'}_{0,i}, w^{bo'}_{0,i+1}, w^{bo'}_{0,i+2}) \cdot (f^{bo'}_{i,j}, u_{i+1,j}, u_{i+2,j})^T, \\ u^{co'} &= (w^{co'}_{0,i-1}, w^{co'}_{0,i}, w^{co'}_{0,i+1}) \cdot (f^{co}_{i-1,j}, u_{i,j}, f^{co'}_{i+1,j})^T, \\ u^{bo'}_x &= (w^{bo'}_{1,i}, w^{bo'}_{1,i+1}, w^{bo'}_{1,i+2}) \cdot (f^{bo'}_{i,j}, u_{i+1,j}, u_{i+2,j})^T, \\ u^{co'}_x &= (w^{co'}_{1,i-1}, w^{co'}_{1,i}, w^{co'}_{1,i+1}) \cdot (f^{co}_{i-1,j}, u_{i,j}, f^{co'}_{i+1,j})^T. \end{aligned} \tag{50}$$

According to Fig. 7, $u^{bo'}_y$ 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 $u^{bo'}_y$ can be represented as

$$\begin{aligned} u^{bo'}_y = &\begin{bmatrix} w^{bo'}_{1,j} \\ w^{bo'}_{1,j+1} \\ w^{bo'}_{1,j+2} \end{bmatrix}^T \cdot \begin{bmatrix} w^{bo'}_{0,i} & w^{bo'}_{0,i+1} & w^{bo'}_{0,i+2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w1^{bo'}_{0,i+1} & w1^{bo'}_{0,i+2} & w1^{bo'}_{0,i+3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w2^{bo'}_{0,i+1} & w2^{bo'}_{0,i+2} & w2^{bo'}_{0,i+3} \end{bmatrix} \\ &\times \left[f^{bo'}_{i,j}, u_{i+1,j}, u_{i+2,j}, u_{i+1,j+1}, u_{i+2,j+1}, u_{i+3,j+1}, u_{i+1,j+2}, u_{i+2,j+2}, u_{i+3,j+2}\right]^T. \end{aligned} \tag{51}$$

Substituting Eq. (50) and Eq. (51) into Eq. (7), we attain other different two linear equations as

$$[u]_{o'} = (w^{bo'}_{0,i}, w^{bo'}_{0,i+1}, w^{bo'}_{0,i+2}) \cdot (f^{bo'}_{i,j}, u_{i+1,j}, u_{i+2,j})^T - (w^{co'}_{0,i-1}, w^{co'}_{0,i}, w^{co'}_{0,i+1}) \cdot (f^{co}_{i-1,j}, u_{i,j}, f^{co'}_{i+1,j})^T \tag{52}$$

$$
\begin{aligned}
&[\beta u_n]_{o'} - \beta^c \tan\theta_{o'}\,[u_\tau]_{o'} \\
&= C_x^{bo'} \begin{bmatrix} w_{1,i}^{bo'} \\ w_{1,i+1}^{bo'} \\ w_{1,i+2}^{bo'} \end{bmatrix}^T \cdot \begin{bmatrix} f_{i,j}^{bo'} \\ u_{i+1,j} \\ u_{i+2,j} \end{bmatrix} \\
&- C_x^{co'} \begin{bmatrix} w_{1,i-1}^{co} \\ w_{1,i}^{co} \\ w_{1,i+1}^{co} \end{bmatrix}^T \cdot \begin{bmatrix} f_{i-1,j}^{co} \\ u_{i,j} \\ f_{i+1,j}^{co'} \end{bmatrix} \\
&+ C_y^{bo'} \begin{bmatrix} w_{1,j} \\ w_{1,j+1} \\ w_{1,j+2} \end{bmatrix}^T \cdot \begin{bmatrix} w_{0,i}^{bo'} & w_{0,i+1}^{bo'} & w_{0,i+2}^{bo'} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w1_{0,i+1}^{bo'} & w1_{0,i+2}^{bo'} & w1_{0,i+3}^{bo'} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w2_{0,i+1}^{bo'} & w2_{0,i+2}^{bo'} & w2_{0,i+3}^{bo'} \end{bmatrix} \cdot \left[f_{i,j}^{bo'}, u_{i+1,j}, u_{i+2,j}, u_{i+1,} \right.
\end{aligned}
\tag{53}
$$

Here the unit normal vector at point o' is $\vec{n} = (\cos\theta_o', \sin\theta_o')$. Interpretations for $C_x^{bo'}$, $w_{1,i}^{bo'}$, $w_{1,i-1}^{co}$, 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

$$
\begin{bmatrix} f_{i-1,j}^{co} \\ f_{i,j}^{ao} \\ f_{i,j}^{bo'} \\ f_{i+1,j}^{co'} \end{bmatrix} = \{C\}_{4\times 23} \cdot \{U\}_{23\times 1},
\tag{54}
$$

where vector $\{U\}_{23\times 1}$ consists of 17 function values and 6 jump conditions

$$
\{U\}_{23\times 1} = (u_{i-2,j}, u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, u_{i-3,j+1}, u_{i-2,j+1}, u_{i-1,j+1}, u_{i+1,j+1}, u_{i+2,j+1}, u_{i+3,j+1}, u_{i-3,j+2}, u_{i-2,j+2}, u_{i-1,j+2}, u_{i+1,j+2}, u_{i+2,j+2}, u_{i+3,j+}
\tag{55}
$$

Here, $\{C\}_{3\times 23}$ is the coefficient matrix and its components are the combination of weights and trigonometric functions of unit normal vectors at points o and o'. Finally, one can discretize $(\beta u_x)_x$ at irregular points $(i-1, j)$, (i, j), and $(i+1, j)$ as

$$
\begin{aligned}
(\beta u_x)_x &= \frac{1}{\Delta x^2}(\beta^a_{i-\frac{3}{2},j}, -\beta^a_{i-\frac{3}{2},j} - \beta^a_{i-\frac{1}{2},j}, \beta^a_{i-\frac{1}{2},j}) \cdot (u_{i-2,j}, u_{i-1,j}, f^{ao}_{i,j})^T \quad \text{at } (i-1, j),\\
(\beta u_x)_x &= \frac{1}{\Delta x^2}(\beta^c_{i-\frac{1}{2},j}, -\beta^c_{i-\frac{1}{2},j} - \beta^c_{i+\frac{1}{2},j}, \beta^c_{i+\frac{1}{2},j}) \cdot (f^{co}_{i-1,j}, u_{i,j}, f^{co'}_{i+1,j})^T \quad \text{at } (i, j),\\
(\beta u_x)_x &= \frac{1}{\Delta x^2}(\beta^b_{i+\frac{1}{2},j}, -\beta^b_{i+\frac{1}{2},j} - \beta^b_{i+\frac{3}{2},j}, \beta^b_{i+\frac{3}{2},j}) \cdot (f^{bo'}_{i,j}, u_{i+1,j}, u_{i+2,j})^T \quad \text{at } (i+1, j).
\end{aligned} \tag{56}
$$

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 $(\beta u_x)_x$.

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 Fig. 8(b), we just need to treat the interface intersecting node as a node in Ω^c and make use of the disassociation strategy at points $(i - 1, j)$ and $(i + 1, j)$. However, for situations in Fig. 9(a) and Fig. 9(b), the simple disassociation approach dose 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 Ω^a, that is $u_{i,j}=u^{ao}_{i,j}$, then using the jump condition between Ω^a and Ω^b, we can obtain $u^{bo}_{i,j}=u^{ao}_{i,j} - [u]^{ab}_o$. The $[u]^{ab}_o$ here means the jump value between Ω^a and Ω^b at point o. When we introduce a fictitious value $f_{i+1,j}$, which is the extension of function in Ω^a to point $(i + 1, j)$, we can discretize u^{ao}_x and u^{bo}_x explicitly as

$$
\begin{aligned}
u^{ao}_x &= (w^{ao}_{1,i-1}, w^{ao}_{1,i}, w^{ao}_{1,i+1}) \cdot (u_{i-1,j}, u_{i,j}, f_{i+1,j})^T,\\
u^{bo}_x &= (w^{bo}_{1,i}, w^{bo}_{1,i+1}, w^{bo}_{1,i+2}) \cdot (u_{i,j} - [u]^{ab}_o, u_{i+1,j}, u_{i+2,j})^T.
\end{aligned} \tag{57}
$$

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 Ω^a. As a result, it is easy for us to obtain an explicit representation of u^{ao}_y in terms of $f_{i,j-1}$, $f_{i,j+1}$ and $u_{i,j}$ as

$$
u^{ao}_y = (w^{ao}_{1,j-1}, w^{ao}_{1,i}, w^{ao}_{1,j+1}) \cdot (f_{i,j-1}, u_{i,j}, f_{i,j+1})^T. \tag{58}
$$

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

$$[\beta u_n]_o^{ab}-\beta^b \tan\theta_o [u_\tau]_o^{ab}=C_x^{ao}\begin{bmatrix} w_{1,i-1}^{ao} \\ w_{1,i}^{ao} \\ w_{1,i+1}^{ao} \end{bmatrix}^T \cdot \begin{bmatrix} u_{i-1,j} \\ u_{i,j} \\ f_{i+1,j} \end{bmatrix} - C_x^{bo}\begin{bmatrix} w_{1,i}^{bo} \\ w_{1,i+1}^{bo} \\ w_{1,i+2}^{bo} \end{bmatrix}^T \cdot \begin{bmatrix} u_{i,j}-[u]_o^{ab} \\ u_{i+1,j} \\ u_{i+2,j} \end{bmatrix} + C_y^{ao}\begin{bmatrix} w_{1,j-1}^{ao} \\ w_{1,j}^{ao} \\ w_{1,j+1}^{ao} \end{bmatrix}^T \cdot \begin{bmatrix} f_{i,j-1} \\ u_{i,j} \\ f_{i,j+1} \end{bmatrix}, \tag{59}$$

here $C_x^{ao}=\beta^a \cos\theta+\beta^b \tan\theta \sin\theta$, $C_x^{bo}=\beta^b \cos\theta+\beta^b \tan\theta \sin\theta$ and $C_y^{ao}=\beta^a \sin\theta-\beta^b \sin\theta$. We can calculate $f_{i+1,j}$ from Eq. (59) as

$$\begin{aligned} f_{i+1,j} &= \{C\}\cdot(u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, [u]_o^{ab}, [\beta u_n]_o^{ab}, [u_\tau]_o^{ab}, f_{i,j-1}, f_{i,j+1})^T \\ &= \{C_1\}\cdot(u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, [u]_o^{ab}, [\beta u_n]_o^{ab}, [u_\tau]_o^{ab})^T + \{C_2\}\cdot f_{i,j-1} + \{C_3\}\cdot f_{i,j+1} \\ &= \{C_1\}\cdot(u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, [u]_o^{ab}, [\beta u_n]_o^{ab}, [u_\tau]_o^{ab})^T + \{C_2\}\cdot\{C\}_{1\times 23}^{i,j-1}\cdot\{U\}_{23\times 1}^{i,j-1} + \{C_3\}\cdot\{C\}_{1\times 23}^{i,j+1}\cdot\{U\}_{23\times 1}^{i,j+1} \\ &= \{C'\}_{1\times 54}\cdot\{U'\}_{54\times 1}, \end{aligned} \tag{60}$$

where $\{C\} = \{\{C_1\}, \{C_2\}, \{C_3\}\}$ and $f_{i,j-1}=\{C\}_{1\times 23}^{i,j-1}\cdot\{U\}_{23\times 1}^{i,j-1}$ $f_{i,j+1}=\{C\}_{1\times 23}^{i,j+1}\cdot\{U\}_{23\times 1}^{i,j+1}$. Although we write $\{U'\}$ as a vector $\{U'\}_{54\times 1}$ including 45 function values and 9 jump conditions, some of the function values may have been repeated counted, as we have different ways of choosing auxiliary points for fictitious value $f_{i,j-1}$ and $f_{i,j+1}$. Therefore, the number of the independent components is usually less than 54. Moreover, if we pay more attention to Fig. 9(b), we can find out that if the unit normal vector at the point o from Ω^a is defined as $\vec{n}$ = (cos θ, sin θ) = (1, 0), Eq. (59) can be simplified as:

$$[\beta u_n]_o^{ab}=C_x^{ao}\begin{bmatrix} w_{1,i-1}^{ao} \\ w_{1,i}^{ao} \\ w_{1,i+1}^{ao} \end{bmatrix}^T \cdot \begin{bmatrix} u_{i-1,j} \\ u_{i,j} \\ f_{i+1,j} \end{bmatrix} - C_x^{bo}\begin{bmatrix} w_{1,i}^{bo} \\ w_{1,i+1}^{bo} \\ w_{1,i+2}^{bo} \end{bmatrix}^T \cdot \begin{bmatrix} u_{i,j}-[u]_o^{ab} \\ u_{i+1,j} \\ u_{i+2,j} \end{bmatrix}. \tag{61}$$

We can easily solve $f_{i+1,j}$ from the equation,

$$f_{i+1,j}=\{C\}_{1\times 6}\cdot(u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, [u]_o^{ab}, [\beta u_n]_o^{ab})^T. \tag{62}$$

So one can discretize $(\beta u_x)_x$ and $(\beta u_y)_y$ at Ω^a point (i, j) as

$$\begin{aligned} (\beta u_x)_x &= \frac{1}{\Delta x^2}(\beta^a_{i-\frac{1}{2},j}, -\beta^a_{i-\frac{1}{2},j}-\beta^a_{i+\frac{1}{2},j}, \beta^a_{i+\frac{1}{2},j})\cdot(u_{i-1,j}, u_{i,j}, f_{i+1,j})^T \quad \text{at } (i, j), \\ (\beta u_y)_y &= \frac{1}{\Delta y^2}(\beta^a_{i,j-\frac{1}{2}}, -\beta^a_{i,j-\frac{1}{2}}-\beta^a_{i,j+\frac{1}{2}}, \beta^a_{i,j+\frac{1}{2}})\cdot(f_{i,j-1}, u_{i,j}, f_{i,j+1})^T \quad \text{at } (i, j). \end{aligned} \tag{63}$$

3.3 Three-material interface scheme 3

When two interfaces intersect with meshlines 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 $(i-1, j+1)$, $(i, j+1)$, $(i+1, j+1)$, $(i+2, j+1)$, $(i-1, j-1)$, $(i, j-1)$, $(i+1, j-1)$ and $(i+2, j-1)$ must locate in Ω^c. For example, here point $(i, j-1)$ locates in Ω^c. Since point (i, j) is an irregular point in Ω^a, we can obtain fictitious value $f_{i,j}^c$. Then we implement the disassociation strategy and create a new scheme as follows.

For situation in Fig. 10, we can obtain fictitious value $f^c_{i,j}$ 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 $f^{ao}_{i+1,j}$ and $f^{bo'}_{i,j}$. To make it clear, $f^{ao}_{i+1,j}$ means the fictitious value at point $(i+1, j)$ and it is an extension of the function value from domain Ω^a. Two more fictitious values from Ω^c are needed and they are combined with fictitious value $f^c_{i,j}$ to interpolate u^{co}, $u^{co'}$, u^{co}_x, and $u^{co'}_x$. These two fictitious values are located adjacent to $f^c_{i,j}$ and they can be either $f^c_{i-1,j}$ and $f^c_{i+1,j}$ or $f^c_{i+1,j}$ and $f^c_{i+2,j}$. Let us just choose $f^c_{i-1,j}$ and $f^c_{i+1,j}$ in the computation below.

For point o, we can discretize u^{ao}, u^{co}, u^{ao}_x, and u^{co}_x by using function values $u_{i-1,j}$ and $u_{i,j}$, and fictitious values $f^c_{i-1,j}$, $f^c_{i,j}$, $f^c_{i+1,j}$ and $f^{ao}_{i+1,j}$.

$$\begin{aligned} u^{ao} &= (w^{ao}_{0,i-1}, w^{ao}_{0,i}, w^{ao}_{0,i+1}) \cdot (u_{i-1,j}, u_{i,j}, f^{ao}_{i+1,j})^T, \\ u^{co} &= (w^{co}_{0,i-1}, w^{co}_{0,i}, w^{co}_{0,i+1}) \cdot (f^c_{i-1,j}, f^c_{i,j}, f^c_{i+1,j})^T, \\ u^{ao}_x &= (w^{ao}_{1,i-1}, w^{ao}_{1,i}, w^{ao}_{1,i+1}) \cdot (u_{i-1,j}, u_{i,j}, f^{ao}_{i+1,j})^T, \\ u^{co}_x &= (w^{co}_{1,i-1}, w^{co}_{1,i}, w^{co}_{1,i+1}) \cdot (f^c_{i-1,j}, f^c_{i,j}, f^c_{i+1,j})^T. \end{aligned} \tag{64}$$

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

$$\begin{aligned} u^{ao}_y = [w^{ao}_{1,j}, w^{ao}_{1,j-1}, w^{ao}_{1,j+1}] \cdot \begin{bmatrix} w^{ao}_{0,i-1} & w^{ao}_{0,i} & w^{ao}_{0,i+1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w1^{ao}_{0,i-3} & w1^{ao}_{0,i-2} & w1^{ao}_{0,i-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w2^{ao}_{0,i-2} & w2^{ao}_{0,i-1} & w2^{ao}_{0,i} \end{bmatrix} \\ \times [u_{i-1,j}, u_{i,j}, f^{ao}_{i+1,j}, u_{i-3,j-1}, u_{i-2,j-1}, u_{i-1,j-1}, u_{i-2,j+1}, u_{i-1,j+1}, u_{i,j+1}]^T. \end{aligned} \tag{65}$$

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

$$[u]_o = (w^{ao}_{0,i-1}, w^{ao}_{0,i}, w^{ao}_{0,i+1}) \cdot (u_{i-1,j}, u_{i,j}, f^{ao}_{i+1,j})^T - (w^{co}_{0,i-1}, w^{co}_{0,i}, w^{co}_{0,i+1}) \cdot (f^c_{i-1,j}, f^c_{i,j}, f^c_{i+1,j})^T. \tag{66}$$

$$
\begin{aligned}
&[\beta u_n]_o - \beta^c \tan\theta_o [u_\tau]_o \\
&= C_x^{ao} \begin{bmatrix} w_{1,i-1}^{ao} \\ w_{1,i}^{ao} \\ w_{1,i+1}^{ao} \end{bmatrix}^T \cdot \begin{bmatrix} u_{i-1,j} \\ u_{i,j} \\ f_{i+1,j}^{ao} \end{bmatrix} \\
&\quad - C_x^{co} \begin{bmatrix} w_{1,i-1}^{co} \\ w_{1,i}^{co} \\ w_{1,i+1}^{co} \end{bmatrix}^T \cdot \begin{bmatrix} f_{i-1,j}^{c} \\ f_{i,j}^{c} \\ f_{i+1,j}^{c} \end{bmatrix} \\
&\quad + C_y^{ao} \begin{bmatrix} w_{1,j}^{ao} \\ w_{1,j-1}^{ao} \\ w_{1,j+1}^{ao} \end{bmatrix}^T \cdot \begin{bmatrix} w_{0,i-1}^{ao} & w_{0,i}^{ao} & w_{0,i+1}^{ao} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w1_{0,i-3}^{ao} & w1_{0,i-2}^{ao} & w1_{0,i-1}^{ao} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w2_{0,i-2}^{ao} & w2_{0,i-1}^{ao} & w2_{0,i}^{ao} \end{bmatrix} \cdot [u_{i-1,j}, u_{i,j}, f_{i+1,j}^{ao}, u_{i-3,j-1},
\end{aligned}
\tag{67}
$$

Therefore, we obtain two linear equations by using the information of interface intersecting point o. For point o', we can make use a similar idea and discretize $u^{bo'}$, $u^{co'}$, $u_x^{bo'}$ and $u_x^{co'}$ explicitly as

$$
\begin{aligned}
u^{bo'} &= (w_{0,i}^{bo'}, w_{0,i+1}^{bo'}, w_{0,i+2}^{bo'}) \cdot (f_{i,j}^{bo'}, u_{i+1,j}, u_{i+2,j})^T, \\
u^{co'} &= (w_{0,i-1}^{co'}, w_{0,i}^{co'}, w_{0,i+1}^{co'}) \cdot (f_{i-1,j}^{c}, f_{i,j}^{c}, f_{i+1,j}^{c})^T, \\
u_x^{bo'} &= (w_{1,i}^{bo'}, w_{1,i+1}^{bo'}, w_{1,i+2}^{bo'}) \cdot (f_{i,j}^{bo'}, u_{i+1,j}, u_{i+2,j})^T, \\
u_x^{co'} &= (w_{1,i-1}^{co'}, w_{1,i}^{co'}, w_{1,i+1}^{co'}) \cdot (f_{i-1,j}^{c}, f_{i,j}^{c}, f_{i+1,j}^{c})^T.
\end{aligned}
\tag{68}
$$

According to Fig. 10, $u_y^{bo'}$ is to be discretized. We need other six auxiliary points $u_{i+2,j-1}$, $u_{i+3,j-1}$, $u_{i+4,j-1}$, $u_{i+1,j+1}$ $u_{i+2,j+1}$ and $u_{i+3,j+1}$. Then $u_y^{bo'}$ can be represented as

$$
u_y^{bo'} = \begin{bmatrix} w_{1,j}^{bo'} \\ w_{1,j-1}^{bo'} \\ w_{1,j+1}^{bo'} \end{bmatrix}^T \cdot \begin{bmatrix} w_{0,i}^{bo'} & w_{0,i+1}^{bo'} & w_{0,i+2}^{bo'} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w1_{0,i+2}^{bo'} & w1_{0,i+3}^{bo'} & w1_{0,i+4}^{bo'} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w2_{0,i+1}^{bo'} & w2_{0,i+2}^{bo'} & w2_{0,i+3}^{bo'} \end{bmatrix}
\tag{69}
$$

$$
\times \left[f_{i,j}^{bo'}, u_{i+1,j}, u_{i+2,j}, u_{i+2,j-1}, u_{i+3,j-1}, u_{i+4,j-1}, u_{i+1,j+1}, u_{i+2,j+1}, u_{i+3,j+1} \right]^T.
\tag{70}
$$

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

$$
[u]_{o'} = (w_{0,i}^{bo'}, w_{0,i+1}^{bo'}, w_{0,i+2}^{bo'}) \cdot (f_{i,j}^{bo'}, u_{i+1,j}, u_{i+2,j})^T - (w_{0,i-1}^{co'}, w_{0,i}^{co'}, w_{0,i+1}^{co'}) \cdot (f_{i-1,j}^{c}, f_{i,j}^{c}, f_{i+1,j}^{c})^T
\tag{71}
$$

$$
\begin{aligned}
&[\beta u_n]_{o'} - \beta^c \tan\theta_{o'} [u_\tau]_{o'} \\
&= C_x^{bo'} \begin{bmatrix} w_{1,i}^{bo'} \\ w_{1,i+1}^{bo'} \\ w_{1,i+2}^{bo'} \end{bmatrix}^T \cdot \begin{bmatrix} f_{i,j}^{bo'} \\ u_{i+1,j} \\ u_{i+2,j} \end{bmatrix} \\
&\quad - C_x^{co'} \begin{bmatrix} w_{1,i-1}^{co} \\ w_{1,i}^{co} \\ w_{1,i+1}^{co} \end{bmatrix}^T \cdot \begin{bmatrix} f_{i-1,j}^{c} \\ f_{i,j}^{c} \\ f_{i+1,j}^{c} \end{bmatrix} \\
&\quad + C_y^{bo'} \begin{bmatrix} w_{1,j}^{bo'} \\ w_{1,j-1}^{bo'} \\ w_{1,j+1}^{bo'} \end{bmatrix}^T \cdot \begin{bmatrix} w_{0,i}^{bo'} & w_{0,i+1}^{bo'} & w_{0,i+2}^{bo'} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w1_{0,i+2}^{bo'} & w1_{0,i+3}^{bo'} & w1_{0,i+4}^{bo'} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w2_{0,i+1}^{bo'} & w2_{0,i+2}^{bo'} & w2_{0,i+3}^{bo'} \end{bmatrix} \cdot \left[f_{i,j}^{bo'}, u_{i+1,j}, u_{i+2,j}, u_{i+2,} \right.
\end{aligned}
\tag{72}
$$

As such, we have attained four equations Eq. (66), Eq. (67), Eq. (71) and Eq. (72). Here $f_{i,j}^c$ 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

$$
\begin{bmatrix} f_{i-1,j}^{co} \\ f_{i,j}^{ao} \\ f_{i,j}^{bo'} \\ f_{i,j}^{bo'} \\ f_{i+1,j}^{co'} \end{bmatrix} = \{C\}_{4\times 45} \cdot \{U\}_{45\times 1}.
\tag{73}
$$

Here vector $\{U\}_{45\times1}$ consists of 17 function values and 6 jump conditions from $f_{i,j}^c$, and other 16 function values and 6 jump conditions as

$$
\begin{aligned}
&\{U'\}_{22\times1} \\
&= (u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i+2,j}, u_{i-3,j-1}, u_{i-2,j-1}, u_{i-1,j-1}, u_{i+2,j-1}, u_{i+3,j-1}, u_{i+4,j-1}, u_{i-2,j+1}, u_{i-1,j+1}, u_{i,j+1}, u_{i+1,j+1}, u_{i+2,j+1}, u_{i+3,j+1}
\end{aligned}
\tag{74}
$$

If we denote $f_{i,j}^c$ as $f_{i,j}^c = \{C_f\}_{1\times 23}^{i,j} \cdot \{U_f\}_{23\times 1}^{i,j}$ then $\{U\}_{45\times1} = \{U'\}_{22\times1} \cup \{U_f\}_{23\times1}^{i,j}$. 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 $(\beta u_x)_x$ at irregular points $(i-1, j)$, (i, j) and $(i+1, j)$ as

$$(\beta u_x)_x=\frac{1}{\Delta x^2}(\beta^a_{i-\frac{1}{2},j}, -\beta^a_{i-\frac{1}{2},j}-\beta^a_{i+\frac{1}{2},j}, \beta^a_{i+\frac{1}{2},j})\cdot(u_{i-1,j}, u_{i,j}, f^{ao}_{i+1,j})^T \quad \text{at } (i, j),$$
$$(\beta u_x)_x=\frac{1}{\Delta x^2}(\beta^b_{i+\frac{1}{2},j}, -\beta^b_{i+\frac{1}{2},j}-\beta^b_{i+\frac{3}{2},j}, \beta^b_{i+\frac{3}{2},j})\cdot(f^{bo'}_{i,j}, u_{i+1,j}, u_{i+2,j})^T \quad \text{at } (i+1, j). \tag{75}$$

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_∞ 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 20×20 meshes and our numerical results are presented on 80×80 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 equation (1) is solved. To specify the interface, we design the level set function $\phi(x, y)$

$$\phi(x, y)= -(y-20x)(y+20x)(x-20y)(x+20y), \tag{76}$$

so that $\Gamma = \{(x, y)|\Phi = 0; x, y, \in \Omega\}$, $\Omega^a = \{(x, y)|\Phi \geq 0; x, y, \in \Omega\}$, $\Omega^b = \{(x, y)|\Phi \leq 0; x, y, \in \Omega\}$. The discontinuous coefficients are given by

$$\beta^a(x, y)=1, \beta^b(x, y)=2. \tag{77}$$

The solution at two different domains is designed as

$$u^a(x, y)=2+\sin(4\pi x)\sin(4\pi y),\ u^b(x, y)=\sin(2\pi x)\sin(2\pi y). \tag{78}$$

In this case, we test our two-material interface schemes. Figure 11 presents the geometric shape and computed results for this problem. It is seen that on a 20×20 grid, except for the origin, which is an on-interface node and treated as an point in Ω^a, 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_∞ 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 equation (1) is solved and Ω^a consists of two parts which can be described by two level set functions

$$\phi_1(x,y) = -\left(y-\frac{1}{7}x\right)\left(y+\frac{1}{7}x\right)\left(y-\frac{1}{9}\right); \tag{79}$$

$$\phi_2(x,y) = -\left(y+3x^2+\frac{1}{9}\right), \tag{80}$$

$\Omega^a = \{(x,y)|\phi_1(x,y) \geq 0;\ x,y, \in \Omega\} \cup \{(x,y)|\phi_2(x,y) \geq 0;\ x,y, \in \Omega\}$, and $\Omega^b = \{(x,y)|\phi_1(x,y) \leq 0;\ x,y, \in \Omega\} \cap \{(x,y)|\phi_2(x,y) \leq 0;\ x,y, \in \Omega\}$. The basic geometry is illustrated in Fig. 12. The discontinuous coefficients are given by

$$\beta^a(x,y)=1, \beta^b(x,y)=2. \tag{81}$$

The solution at two different domains is designed as

$$u^a(x,y)=2+\sin(2\pi x)\sin(2\pi y), u^b(x,y)=\sin(2\pi x)\sin(2\pi y). \tag{82}$$

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 $\vec{n}=(1/\sqrt{50}, -7/\sqrt{50})$ 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 × 20 grid if we change the unit normal vector to $\vec{n}$ = (0, −1), L_∞ 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_∞ 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 equation (34) is solved. Now domain Ω is divided into three subdomains Ω^a, Ω^b and Ω^c by two interfaces Γ_1 and Γ_2, which can be described by two level set functions $\phi_1(x,y)$ and $\phi_2(x,y)$

$$\phi_1(x,y)=\frac{8}{17}-\sqrt{\left(x+\frac{8}{17}\right)^2+y^2}; \tag{83}$$

$$\phi_2(x,y)=\frac{8}{17}-\sqrt{\left(x-\frac{8}{17}\right)^2+y^2}, \tag{84}$$

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 β values and solution types. Here six different combinations of solutions and β values are studied

- **Case 3(a):**

$$\beta^a=1, \beta^b=2, \beta^c=3, \tag{85}$$

$$u^a(x, y)=8+\sin(4\pi x)\ \sin(4\pi y), \tag{86}$$

$$u^b(x, y)=4+\sin(2\pi x)\ \sin(2\pi y), \tag{87}$$

$$u^c(x, y)=x^2+y^2+\sin(4\pi x)\ \sin(4\pi y); \tag{88}$$

- **Case 3(b):**

$$\beta^a=1, \beta^b=2+\sin(x+y), \beta^c=3+x^2+y^2, \tag{89}$$

$$u^a(x, y)=8+\sin(4\pi x)\ \sin(4\pi y), \tag{90}$$

$$u^b(x, y)=4+\sin(2\pi x)\ \sin(2\pi y), \tag{91}$$

$$u^c(x, y)=x^2+y^2+\sin(4\pi x)\ \sin(4\pi y); \tag{92}$$

- **Case 3(c):**

$$\beta^a=0.0001, \beta^b=10000, \beta^c=30000, \tag{93}$$

$$u^a(x, y)=4+x^2+y^2, \tag{94}$$

$$u^b(x, y)=6+\exp(x+y), \tag{95}$$

$$u^c(x, y)=x+y+xy; \tag{96}$$

- **Case 3(d):**

$$\beta^a=100,\beta^b=100,\beta^c=3000, \tag{97}$$

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

$$u^b(x,y)=4+\sin(x+y), \tag{99}$$

$$u^c(x,y)=x^2+y^2+\sin(x+y); \tag{100}$$

- **Case 3(e):**

$$\beta^a=1,\beta^b=2,\beta^c=3000, \tag{101}$$

$$u^a(x,y)=8+\sin(4\pi x)\sin(4\pi y), \tag{102}$$

$$u^b(x,y)=4+\sin(2\pi x)\sin(2\pi y), \tag{103}$$

$$u^c(x,y)=x^2+y^2+\sin(4\pi x)\sin(4\pi y); \tag{104}$$

- **Case 3(f):**

$$\beta^a=1000,\beta^b=2,\beta^c=3000, \tag{105}$$

$$u^a(x,y)=4+\sin(x)\sin(y), \tag{106}$$

$$u^b(x,y)=8+\sin(\pi x)\sin(\pi y), \tag{107}$$

$$u^c(x,y)=x^2+y^2+\sin(\pi x)\sin(\pi y); \tag{108}$$

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 Ω^a and Ω^b, between Ω^a and Ω^c, and between Ω^b and Ω^c) as indicated in Eqs. (39)–(44). However, Three-material interface scheme 2 only makes use of

the normal jump condition between Ω^a and Ω^b. Based on the geometry, we choose the unit normal vector to be $\vec{n} = (1, 0)$. We have test different choices of the unit normal vector. For example, if we change it to $\vec{n}=(1/\sqrt{2}, 1/\sqrt{2})$ for Case 3(a), the L_∞ 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 contrast ratios between different domains are very high, the proposed schemes can achieve second order accuracy as indicated by Tables 3 and 4.

Case 3(d) is used to test moderately oscillatory solution situations. In Case 3(e) and Case 3(f), we test sensitivity of the solutions in subdomains. It is seen that Ω^c can have more potential 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 Ω^a, which decreases the accuracy when the solution and coefficient in Ω^a are challenging. For each case, computed L_∞ 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 equation (34). Interfaces and subdomains are prescribed by two piecewise smooth level set functions to outline

$$\phi_1(x,y)=\begin{cases}(y-0.4x-0.2)(y-0.4x+0.2)(y+0.4x-0.2)(y+0.4x+0.2) & |x|\le 0.5, |y|\le 0.2;\\ -1 & \text{else.}\end{cases} \tag{109}$$

and

$$\phi_2(x,y)=\begin{cases}(y-0.4x-0.2)(y+0.4x-0.2) & y\ge 0.2;\\ (y+0.4x+0.2)(y-0.4x+0.2) & y\le -0.2;\\ -(y+0.4x+0.2)(y-0.4x-0.2) & x\le -0.5, |y|<0.2;\\ -(y-0.4x+0.2)(y+0.4x-0.2) & x\ge -0.5, |y|<0.2.\end{cases} \tag{110}$$

We employ the following discontinuous coefficients

$$\beta^a(x,y)=1, \beta^b(x,y)=2, \beta^c(x,y)=3. \tag{111}$$

The solution at three different domains is designed as

$$u^a(x,y)=6+\sin(4\pi x)\sin(4\pi y), u^b(x,y)=4+\sin(4\pi x)\sin(4\pi y), u^c(x,y)=x^2+y^2+\sin(x+y). \tag{112}$$

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 Ω^a are treated as points in Ω^c and suitable disassociation schemes are used. Presented in Table 5, computed L_∞ and L_2 errors indicate the second order accuracy.

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

$$\phi_1(x,y)=\begin{cases}\left(y-\frac{3}{2}x+\frac{1}{18}\right)\left(y+\frac{3}{2}x-\frac{5}{18}\right) & x\geq\frac{1}{9};\\ -1 & \text{else.}\end{cases} \tag{113}$$

and

$$\phi_2(x,y)=\begin{cases}\left(y-\frac{3}{2}x+\frac{1}{18}\right)\left(y+\frac{3}{2}x-\frac{5}{18}\right) & x\leq\frac{1}{9};\\ -1 & \text{else,}\end{cases} \tag{114}$$

for interfaces and subdomains. Let us choose the discontinuous coefficients as

$$\beta^a(x,y)=1,\beta^b(x,y)=2,\beta^c(x,y)=3. \tag{115}$$

The solution at three different domains is given by

$$u^a(x,y)=6+\sin(2\pi x)\sin(2\pi y),\ u^b(x,y)=8+\sin(x+y),\ u^c(x,y)=x^2+y^2. \tag{116}$$

In this case, the performance of Three-material interface scheme 3 is tested. It is seen that, on a 20 × 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 equation (34) with two piecewise smooth level set functions

$$\phi_1(x,y)=\begin{cases}-\left(y-4x^2+\frac{1}{3}\right)\left(y+4x^2-\frac{1}{3}\right) & |x|\leq\sqrt{\frac{1}{12}};\\ -1 & \text{else.}\end{cases} \tag{117}$$

and

$$\phi_2(x,y)=\begin{cases}-\left(y-4x^2+\frac{1}{3}\right)\left(y+4x^2-\frac{1}{3}\right) & |x|\geq\sqrt{\frac{1}{12}};\\ -1 & \text{else.}\end{cases} \tag{118}$$

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

$$\beta^a(x,y)=1,\ \beta^b(x,y)=2,\ \beta^c(x,y)=3. \tag{119}$$

The solution at three different domains is designed as

$$u^a(x,y)=6+\sin(2\pi x)\sin(2\pi y),\ u^b(x,y)=8+\sin(2\pi x)\sin(2\pi y),\ u^c(x,y)=x^2+y^2. \tag{120}$$

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_∞ and L_2 errors. Obviously, the second order accuracy is achieved.

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

$$\phi_1(x,y)=\begin{cases}\left(y-2x+\frac{1}{9}\right) & x<\frac{1}{9};\\ 0 & x=\frac{1}{9}, y\geq\frac{1}{9};\\ -1 & \text{else.}\end{cases} \tag{121}$$

and

$$\phi_2(x,y)=\begin{cases}\left(y+x-\frac{2}{9}\right) & x>\frac{1}{9};\\ 0 & x=\frac{1}{9}, y\geq\frac{1}{9};\\ -1 & \text{else.}\end{cases} \tag{122}$$

The discontinuous coefficients are allowed to vary over different subdomains

$$\beta^a(x,y)=1+\exp(x+y),\ \beta^b(x,y)=2+\sin(x+y),\ \beta^c(x,y)=3+x^2+y^2. \tag{123}$$

The solution at three different domains is designed as

$$u^a(x,y)=8+\sin(\pi x)\sin(\pi y),\ u^b(x,y)=4+\sin(2\pi x)\sin(2\pi y),\ u^c(x,y)=x^2+y^2+\sin(\pi x)\sin(\pi y). \tag{124}$$

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_∞ 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

$$u(r,\theta)=r^{k\pi/\omega}\sin(k\pi\theta/\omega), \tag{125}$$

where k is a wavenumber and ω is the angle parameter of the singular geometry. This solution diverges for certain k values as $r\rightarrow 0$.

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

$$\phi_1(x,y)=\begin{cases} -\left(y-\frac{x}{\sqrt{3}}\right)\left(y+\frac{x}{\sqrt{3}}\right) & x\le 0; \\ -1 & \text{else.} \end{cases} \tag{126}$$

and

$$\phi_2(x,y)=\begin{cases} -\left(y-\frac{x}{\sqrt{3}}\right)\left(y+\frac{x}{\sqrt{3}}\right) & x\ge 0; \\ -1 & \text{else.} \end{cases} \tag{127}$$

The discontinuous coefficients in different subdomains are

$$\beta^a(x,y)=1,\ \beta^b(x,y)=2,\ \beta^c(x,y)=3. \tag{128}$$

We consider a solution that takes the form of Eq. (125)

$$\begin{aligned} u^a(x,y) &= 4+\sin(2\pi x)\sin(2\pi y) \\ u^b(r,\theta) &= r^{k\pi/\omega}\sin(k\pi\theta/\omega) \\ u^c(x,y) &= x^2+y^2+\sin(\pi x)\sin(\pi y), \end{aligned} \tag{129}$$

where we choose $k=\frac{2}{9}$ and ω = π/3. Obviously, this solution has an unbound first order derivative. For this kind of problems, Galerkin formulations can handle it directly but collocation formulations can not. 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 Ω^b with real part,

$$w_1(x,y)=r^{\frac{8}{3}}\cos\frac{8}{3}\theta, \tag{130}$$

and imaginary part,

$$w_2(x,y)=r^{\frac{8}{3}}\sin\frac{8}{3}\theta. \tag{131}$$

Then the u^b can be calculated through the expression:

$$u^b=\mathrm{Im}(w^{\frac{1}{4}}). \tag{132}$$

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_∞ 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_∞ 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 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.

Literature cited

1. Adams L, Li ZL. The immersed interface/multigrid methods for interface problems. SIAM J. Sci. Comput. 2002; 24:463–479.

2. Ameur HB, Burger M, Hackl B. Level set methods for geometric inverse problems in linear elasticity. Inverse Problems. 2004; 20(3):673–696.
3. Beale JT, Layton AT. On the accuracy of finite difference methods for elliptic problems with interfaces. Comm. Appl. Math. Comp. Sci. 2006; 1:91–119.
4. Berthelsen PA. A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions. J. Comput. Phys. 2004; 197(1):364–386.
5. Biros G, Ying L, Zorin D. A fast solver for the Stokes equations with distributed forces in complex geometries. J. Comput. Phys. 2004; 193(1):317–348.
6. Cai W, Deng SZ. An upwinding embedded boundary method for Maxwell's equations in media with material interfaces: 2d case. J. Comput. Phys. 2003; 190:159–183.
7. Chen D, Chen Z, Chen C, Geng WH, Wei GW. MIBPB: A software package for electrostatic analysis. J. Comput. Chem. 2010 in press.
8. Chen D, Wei GW, Cong X, Wang G. Computational methods for optical molecular imaging. Communications in Numerical Methods in Engineering. 2009; 25:1137–1161. [PubMed: 20485461]
9. Chen T, Strain J. Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems. J. Comput. Phys. 2008; 16:7503–7542.
10. Deng SZ, Ito K, Li ZL. Three-dimensional elliptic solvers for interface problems and applications. J. Comput. Phys. 2003; 184:215–243.
11. Dumett MA, Keener JP. An immersed interface method for solving anisotropic elliptic boundary value problems in three dimensions. SIAM Journal on Scientific Computing. 2003; 25(1):348–367.
12. Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. J. Comput. Phys. 2000; 161(1):35–60.
13. Fedkiw RP, Aslam T, Merriman B, Osher S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). J. Comput. Phys. 1999; 152:457–492.
14. Fogelson AL, Keener JP. Immersed interface methods for neumann and related problems in two and three dimensions. SIAM Journal on Scientific Computing. 2001; 22(5):1630–1654.
15. Francois M, Shyy W. Computations of drop dynamics with the immersed boundary method, part 2: Drop impact and heat transfer. Numer. Heat Trans. Part B-Fund. 2003; 44
16. Francois M, Uzgoren E, Jackson J, Shyy W. Multigrid computations with the immersed boundary technique for multiphase flows. Int. J. Numer. Meth. Heat Fluid Flow. 2004; 14:98–115.
17. Geng W, Yu S, Wei GW. Treatment of charge singularities in implicit solvent models. Journal of Chemical Physics. 2007; 127:114106. [PubMed: 17887827]
18. Geng WH, Wei GW. Multiscale molecular dynamics via the matched interface and boundary (mib) method. J. Comput. Phys. 2010 in press.
19. Gibou F, Fedkiw R. A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem. J. Comput. Phys. 2005; 202(2):577–601.
20. Griffith BE, Peskin CS. On the order of accuracy of the immersed boundary method: Higher order convergence rates for sufficiently smooth problems. J. Comput. Phys. 2005; 208:75–105.
21. Grisvard, P. Elliptic Problems in Nonsmooth Domains. Pitman Advanced Pub. Program; 1985.
22. Hadley GR. High-accuracy finite-difference equations for dielectric waveguide analysis i: uniform regions and dielectric interfaces. Journal of Lightwave Technology. 2002; 20:1210–1218.
23. Hesthaven JS. High-order accurate methods in time-domain computational electromagnetics. a review. Advances in Imaging and Electron Physics. 2003; 127:59–123.
24. Horikis TP, Kath WL. Modal analysis of circular bragg fibers with arbitrary index profiles. Optics Lett. 2006; 31:3417–3419.
25. Hou S, Liu X-D. A numerical method for solving variable coefficient elliptic equation with interfaces. J. Comput. Phys. 2005; 202(2):411–445.
26. Hou SM, Wang W, Wang LQ. Numerical method for solving matrix coefficient elliptic equation with sharp-edged interfaces. J. Comput. Phys. 2010; 229(7162–7179)
27. Hou TY, Li ZL, Osher S, Zhao HK. A hybrid method for moving interface problems with application to the heleshaw flow. J. Comput. Phys. 1997; 134(2):236–252.

28. Huang H, Li ZL. Convergence analysis of the immersed interface method. IMA Journal of Numerical Analysis. 1999; 19(4):583–608.
29. Hunter JK, Li ZL, Zhao H. Reactive autophobic spreading of drops. J. Comput. Phys. 2002; 183(2):335–366.
30. Iaccarino G, Verzicco R. Immersed boundary technique for turbulent flow simulations. Appl. Mech. Rev. 2003; 56:331–347.
31. Jin S, Wang X. Robust numerical simulation of porosity evolution in chemical vapor infiltration: II. two-dimensional anisotropic fronts. J. Comput. Phys. 2002; 179(2):557–577.
32. Johansen H, Colella P. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. J. Comput. Phys. 1998; 147(1):60–85.
33. Kafafy R, Lin T, Lin Y, Wang J. Three-dimensional immersed finite element methods for electric field simulation in composite materials. Int. J. Numer. Methods Engng. 2005; 64:940–972.
34. Kandilarov, JD. Immersed interface method for a reaction-diffusion equation with a moving own concentrated source; NMA '02: Revised Papers from the 5th International Conference on Numerical Methods and Applications; London, UK: Springer-Verlag; 2003. p. 506-513.
35. Lai MC, Peskin CS. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. J. Comput. Phys. 2000; 160:705–719.
36. Layton AT. Using integral equations and the immersed interface method to solve immersed boundary problems with stiff forces. Comput. Fluids. 2009; 38:266–272.
37. Lee L, LeVeque RJ. An immersed interface method for incompressible navier–stokes equations. SIAM Journal on Scientific Computing. 2003; 25(3):832–856.
38. LeVeque RJ, Li ZL. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J. Numer. Anal. 1994; 31:1019–1044.
39. Li Z, Lubkin SR. Numerical analysis of interfacial two-dimensional stokes flow with discontinuous viscosity and variable surface tension. International Journal for Numerical Methods in Fluids. 2001; 37(5):525–540.
40. Li Z, Wang W-C, Chern I-L, Lai M-C. New formulations for interface problems in polar coordinates. SIAM Journal on Scientific Computing. 2003; 25(1):224–245.
41. Li ZL, Ito K. Maximum principle preserving schemes for interface problems with discontinuous coefficients. SIAM J. Sci. Comput. 2001; 23:339–361.
42. Linnick MN, Fasel HF. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. J. Comput. Phys. 2005; 204(1):157–192.
43. Liu WK, Liu Y, Farrell D, Zhang L, Wang X, Fukui Y, Patankar N, Zhang Y, Bajaj C, Chen X, Hsu H. Immersed finite element method and its applications to biological systems. Computer Methods in Applied Mechanics and Engineering. 2006; 195:1722–1749. [PubMed: 20200602]
44. Liu XD, Fedkiw RP, Kang M. A boundary condition capturing method for Poisson's equation on irregular domains. J. Comput. Phys. 2000; 160:151–178.
45. Lombard B, Piraux J. How to incorporate the spring-mass conditions in finite-difference schemes. SIAM Journal on Scientific Computing. 2003; 24(4):1379–1407.
46. Mayo A. The fast solution of Poisson's and the biharmonic equations on irregular regions. SIAM J. Numer. Anal. 1984; 21:285–299.
47. Mayo M, Greenbaum A. Fourth order accurate evaluation of integrals in potential theory on exterior 3d regions. J. Comput. Phys. 2007; 220:900–914.
48. McKenney A, Greengard L, Mayo A. A fast Poisson solver for complex geometries. J. Comput. Phys. 1995; 118:348–355.
49. Mittal M, Iaccarino G. Immersed boundary methods. Annu. Rev. Fluid Mech. 2005; 37:236–261.
50. Morgenthal G, Walther JH. An immersed interface method for the vortex-in-cell algorithm. Computers & Structures. 2007; 85(11–14):712–726.
51. Oevermann M, Klein R. A cartesian grid finite volume method for elliptic equations with variable coefficients and embedded interfaces. J. Comput. Phys. 2006; 219:749–769.
52. Peskin CP, McQueen DM. A 3-dimensional computational method for blood-flow in the heart. 1. immersed elastic fibers in a viscous incompressible fluid. J. Comput. Phys. 1989; 81:372–405.
53. Peskin CS. Numerical analysis of blood flow in the heart. J. Comput. Phys. 1977; 25(3):220–252.

54. Peskin CS. Lectures on mathematical aspects of physiology. Lectures in Appl. Math. 1981; 19:69–107.
55. Sanner MF, Olson AJ, Spehner JC. Reduced surface: An efficient way to compute molecular surfaces. Biopolymers. 1996; 38:305–320. [PubMed: 8906967]
56. Schulz M, Steinebach G. Two-dimensional modelling of the river Rhine. Journal of Computational and Applied Mathematics. 2002; 145(1):11–20.
57. Sethian JA. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. J. Comput. Phys. 2001; 169(2):503–555.
58. Tornberg A-K, Engquist B. Numerical approximations of singular source terms in differential equations. J. Comput. Phys. 2004; 200(2):462–488.
59. Vande Voorde JJ, Vierendeels J, Dick E. Flow simulations in rotary volumetric pumps and compressors with the fictitious domain method. Journal of Computational and Applied Mathematics. 2004; 168(1–2):491–499.
60. Wei GW. Discrete singular convolution for the solution of the Fokker-Planck equations. J. Chem. Phys. 1999; 110(8930–8942)
61. Wei GW, B ZY, Xiang Y. Discrete singular convolution and its application to the analysis of plates with internal supports, I theory and algorithm. Int. J. Numer. Methods Engng. 2002; 55:913–946.
62. Wiegmann A, Bube KP. The explicit-jump immersed interface method: Finite difference methods for PDEs with piecewise smooth solutions. SIAM Journal on Numerical Analysis. 2000; 37(3): 827–862.
63. Xia, KL.; Zhan, M.; Wei, GW. Poster: Matched interface and boundary (MIB) method for multi-domain elliptic interface problems. Workshop on Fluid Motion Driven by Immersed Structures; August 9–13 2010; Toronto: Fields Institute;
64. Yu S, Geng W, Wei GW. Treatment of geometric singularities in implicit solvent models. Journal of Chemical Physics. 2007; 126:244108. [PubMed: 17614538]
65. Yu S, Wei GW. Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities. J. Comput. Phys. 2007; 227:602–632.
66. Yu S, Zhou Y, Wei GW. Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces. J. Comput. Phys. 2007; 224(2):729–756.
67. Zhao S. High order matched interface and boundary methods for the helmholtz equation in media with arbitrarily curved interfaces. J. Comput. Phys. 2010; 229:3155–3170.
68. Zhao S, Wei GW. High-order FDTD methods via derivative matching for Maxwell's equations with material interfaces. J. Comput. Phys. 2004; 200(1):60–103.
69. Zhao S, Wei GW. Matched interface and boundary (MIB) for the implementation of boundary conditions in high order central finite differences. International Journal for Numerical Methods in Engineering. 2009; 77:1690–1730. [PubMed: 20485574]
70. Zhou YC, Feig M, Wei GW. Highly accurate biomolecular electrostatics in continuum dielectric environments. Journal of Computational Chemistry. 2008; 29:87–97. [PubMed: 17508411]
71. Zhou YC, Wei GW. On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method. J. Comput. Phys. 2006; 219(1):228–246.
72. Zhou YC, Zhao S, Feig M, Wei GW. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. J. Comput. Phys. 2006; 213(1):1–30.

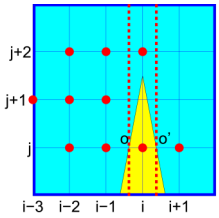

Figure 1.
Scheme for two-material interfaces. In this figure as well as all other figures Ω^a is marked with yellow color and Ω^b is denoted by green color.

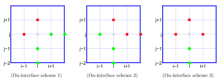

Figure 2.
On-interface schemes 1, 2 and 3 proposed in our earlier work [66]. The interface passes through point (*i*, *j*). All the green dots are on the same side of the interface and all the red dots are on the other side of the interface.

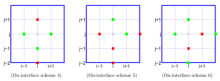

Figure 3.
On-interface schemes 4, 5 and 6. The interface passes through node point (i, j). All the green dots are on the same side of the interface and all the red dots are on the other side of the interface.

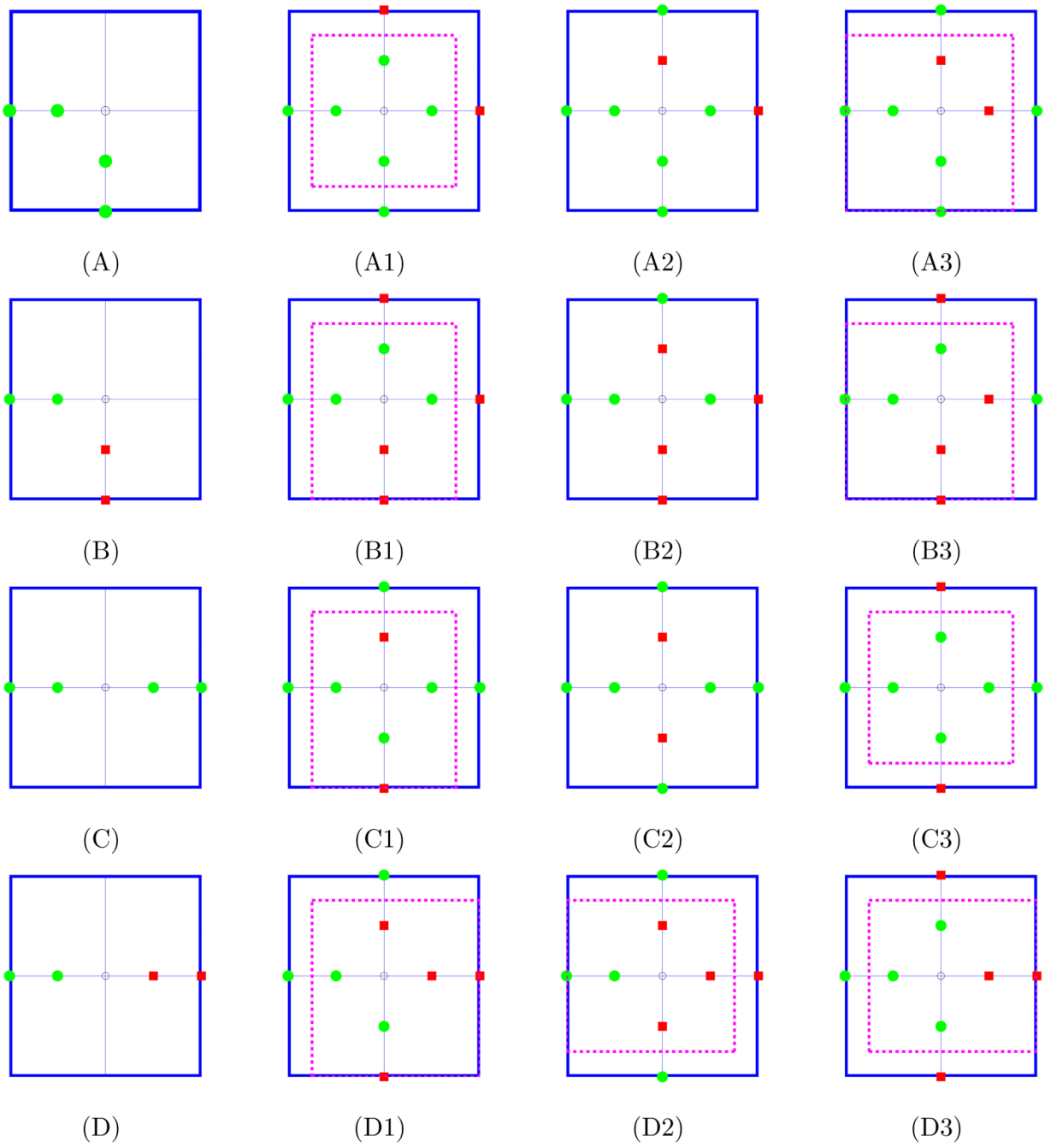

Figure 4.
The situations that there are two pairs of grid points surrounding the interface intersecting node of the grid. The cases encircled by manganviolett dash lines can be resolved by using one of six on-interface schemes.

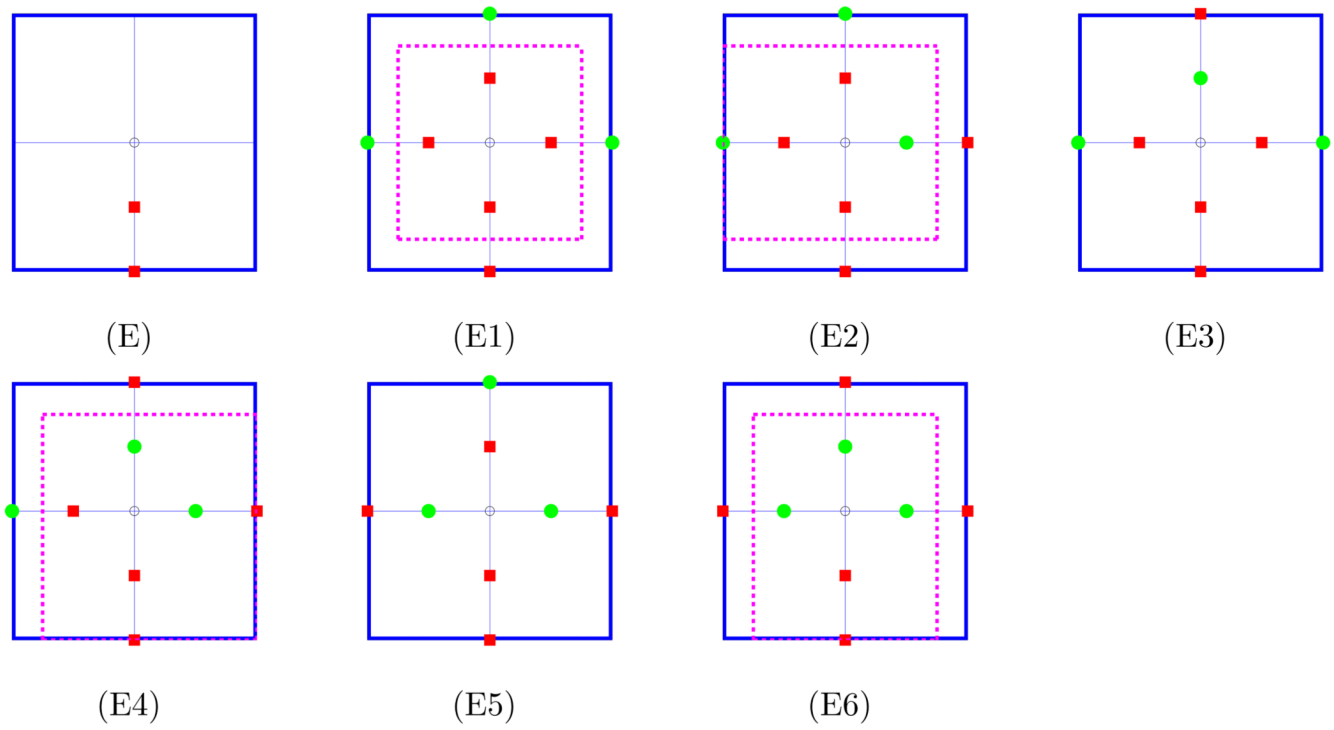

Figure 5.
The situations that there is one pair of grid points surrounding the interface intersecting node of the grid. The cases encircled by manganviolett dash lines can be resolved by using one of six on-interface schemes.

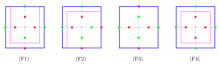

Figure 6.
The situations that there is no pair of grid points surrounding the interface intersecting node of the grid. The cases encircled by manganviolett dash lines can be resolved by using one of six on-interface schemes.

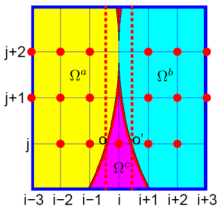

Figure 7.
Three-material interface scheme 1. The left yellow area is Ω^a, the right green area is Ω^b, the middle red area is Ω^c.

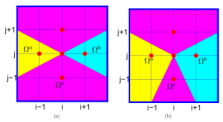

Figure 8.
Three-material interface scheme 2. The left yellow area is Ω^a, the right green area is Ω^b, the middle red area is Ω^c.

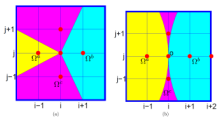

Figure 9.
Three-material interface scheme 2. The left yellow area is Ω^a, the right green area is Ω^b, the middle red area is Ω^c.

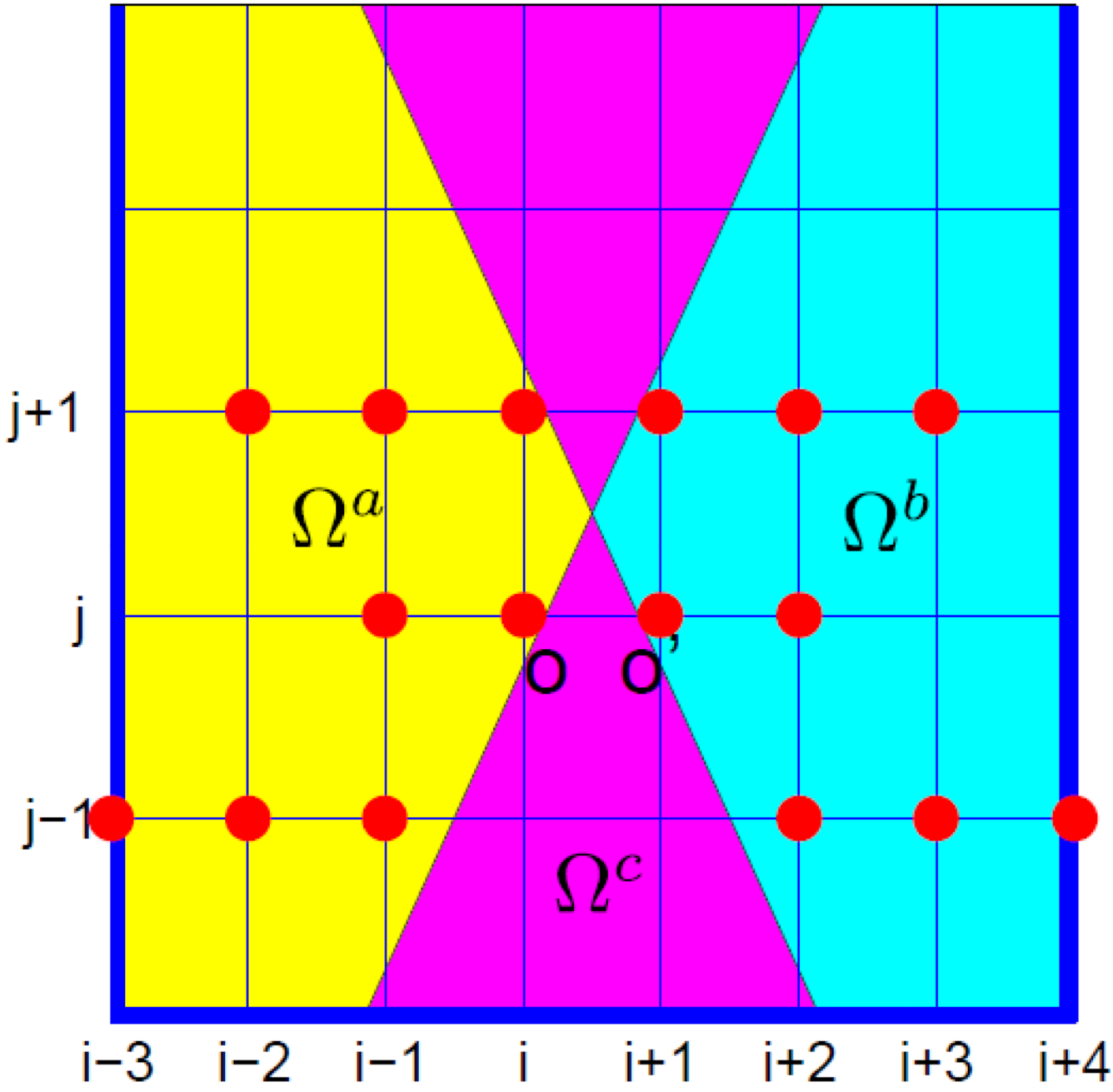

Figure 10.
Three-material interface scheme 3: The left yellow area is Ω^a, the right green area is Ω^b, the middle red area is Ω^c.

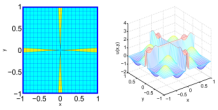

Figure 11.
Basic geometry on a 20 × 20 grid (Left) and the computed solution on 80 × 80 grid (Right) for Case 1.

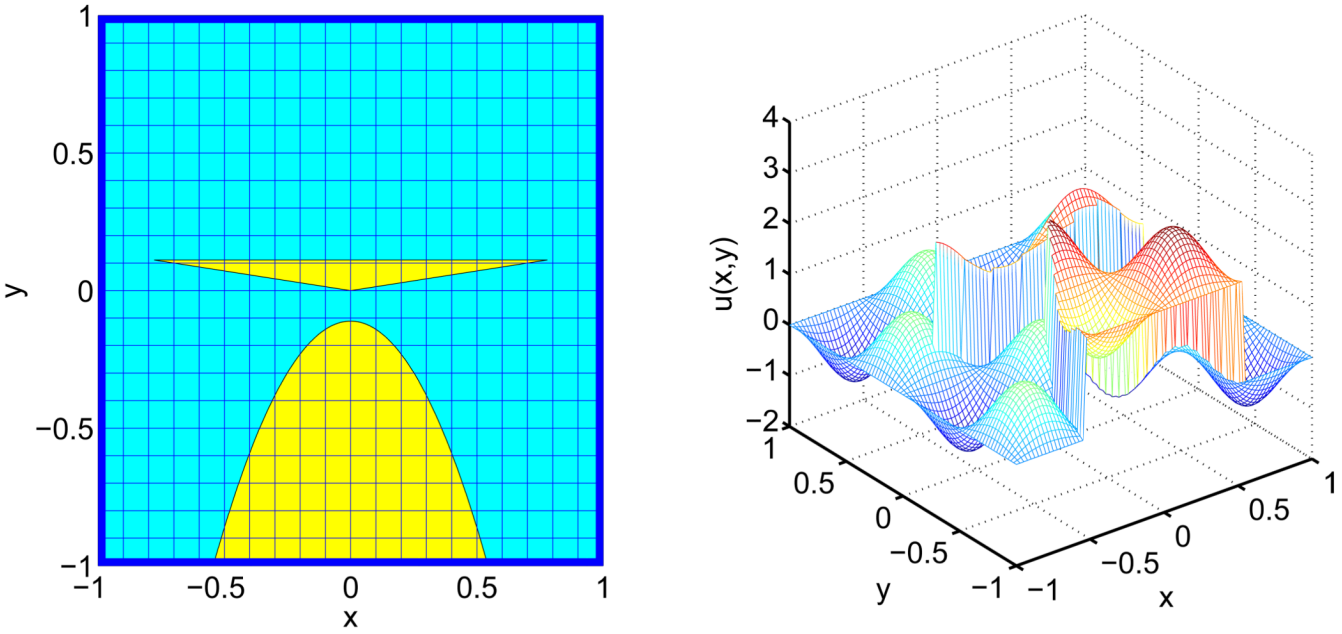

Figure 12.
Basic geometry on a 20 × 20 grid (Left) and the computed solution on a 80 × 80 grid (Right) for Case 2.

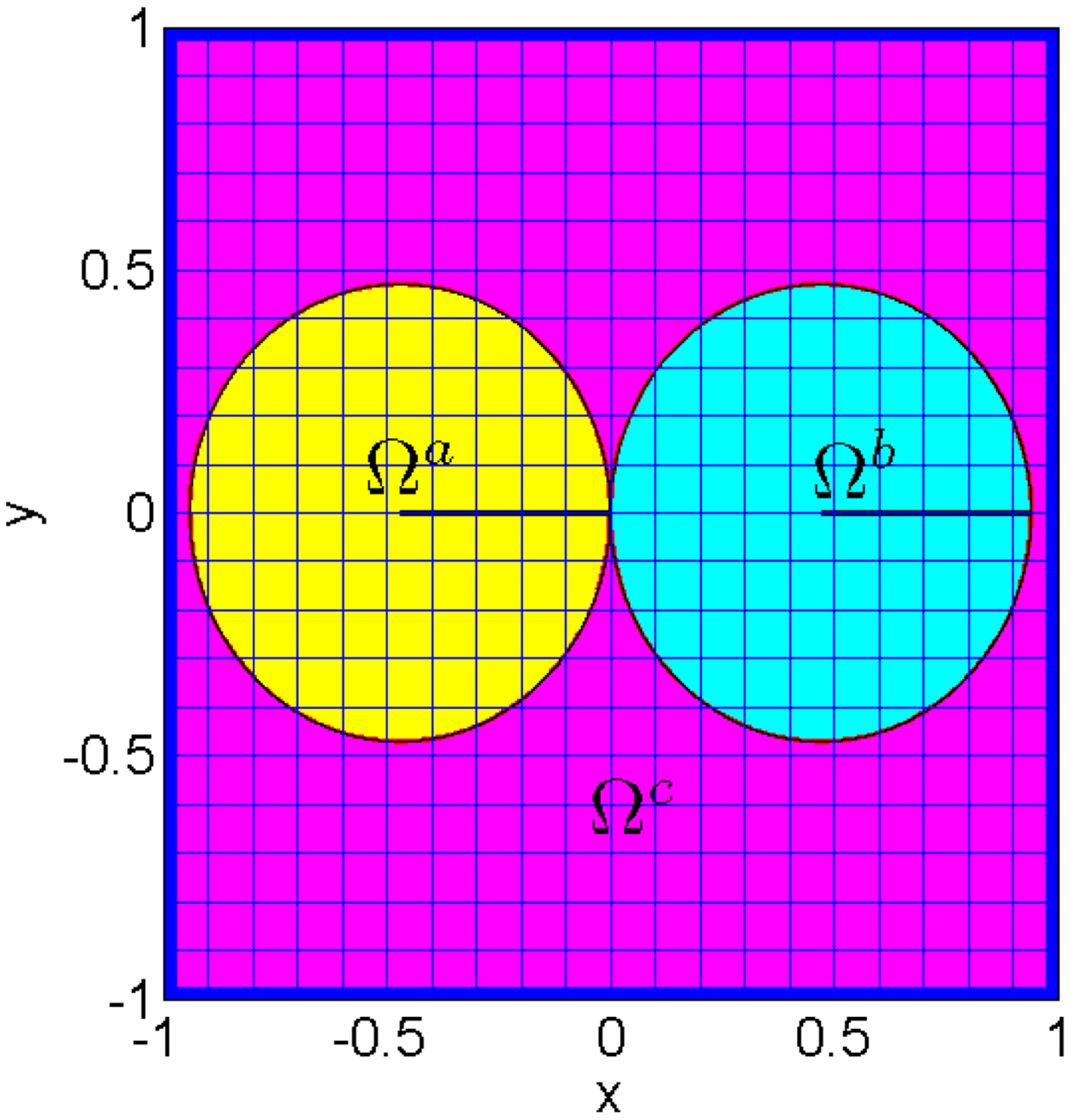

Figure 13.
Basic geometry on 20 × 20 mesh for Case 3.

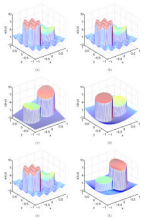

Figure 14.
Illustration of numerical solutions on the 80 × 80 mesh for Case 3.

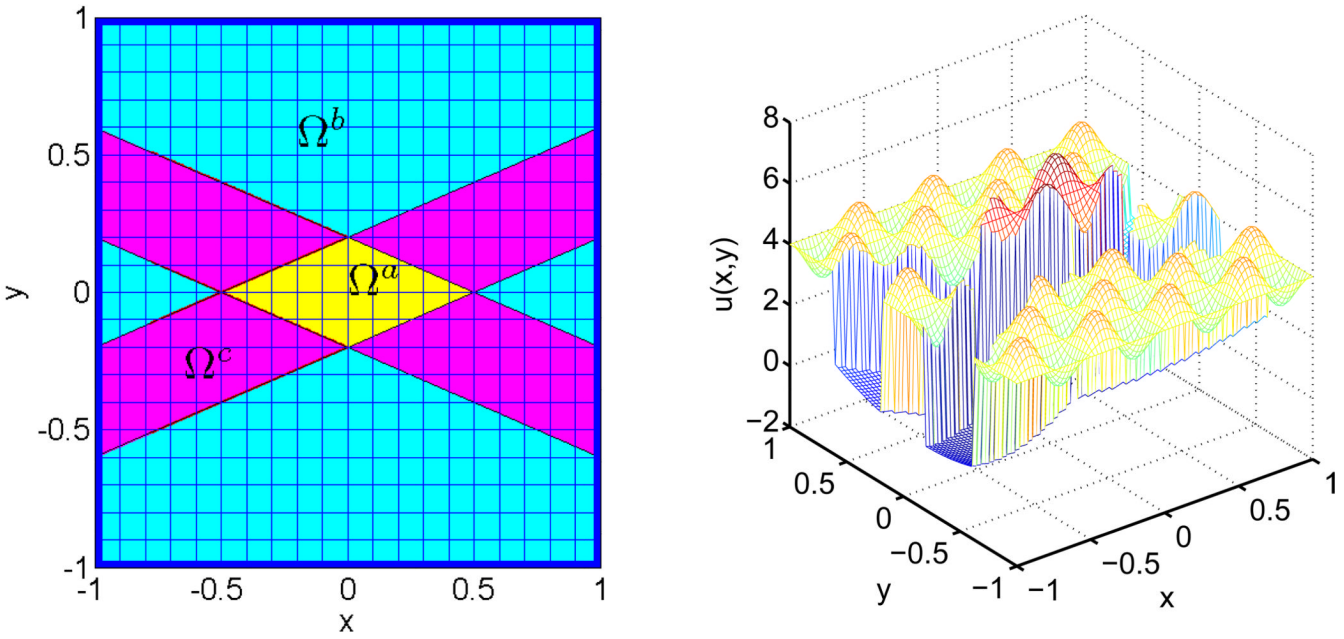

Figure 15.
Basic geometry on a 20 × 20 grid (Left) and the computed solution on a 80 × 80 grid (Right) for Case 4.

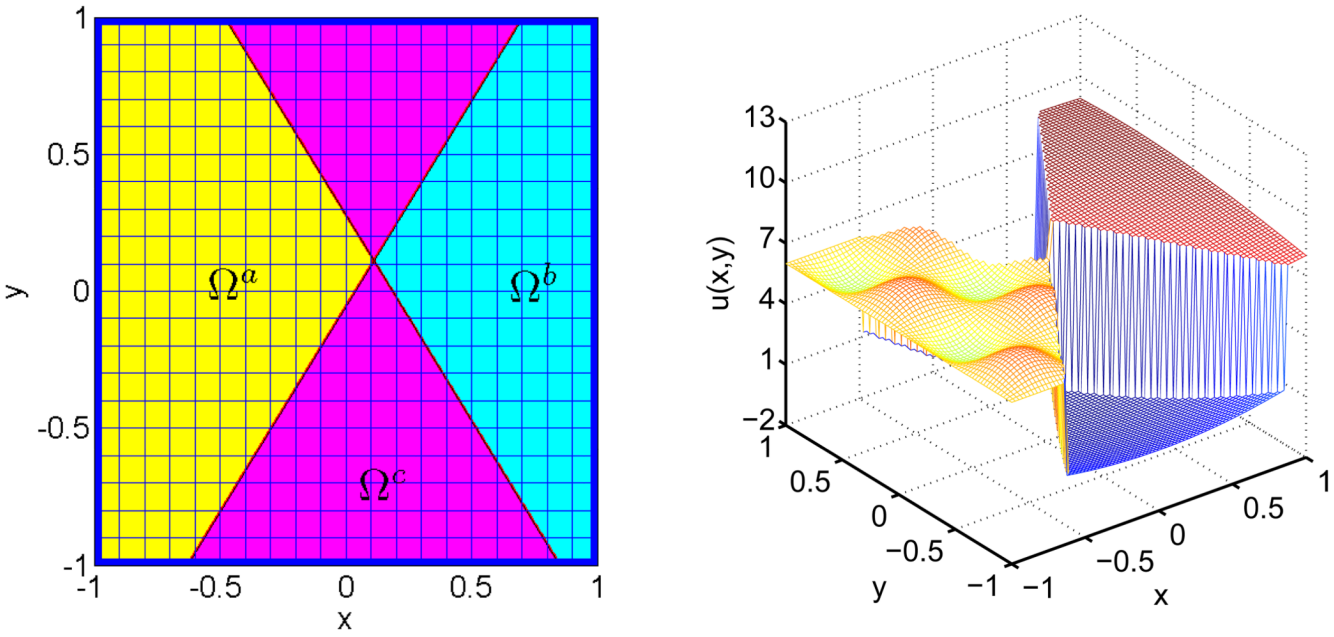

Figure 16.
Basic geometry on a 20 × 20 grid (left) and the computed solution on a 80 × 80 grid (right) for Case 5.

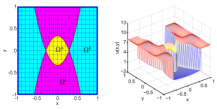

Figure 17.
Basic geometry on a 20 × 20 mesh (left) and the numerical solution on 80 × 80 mesh (right) for Case 6.

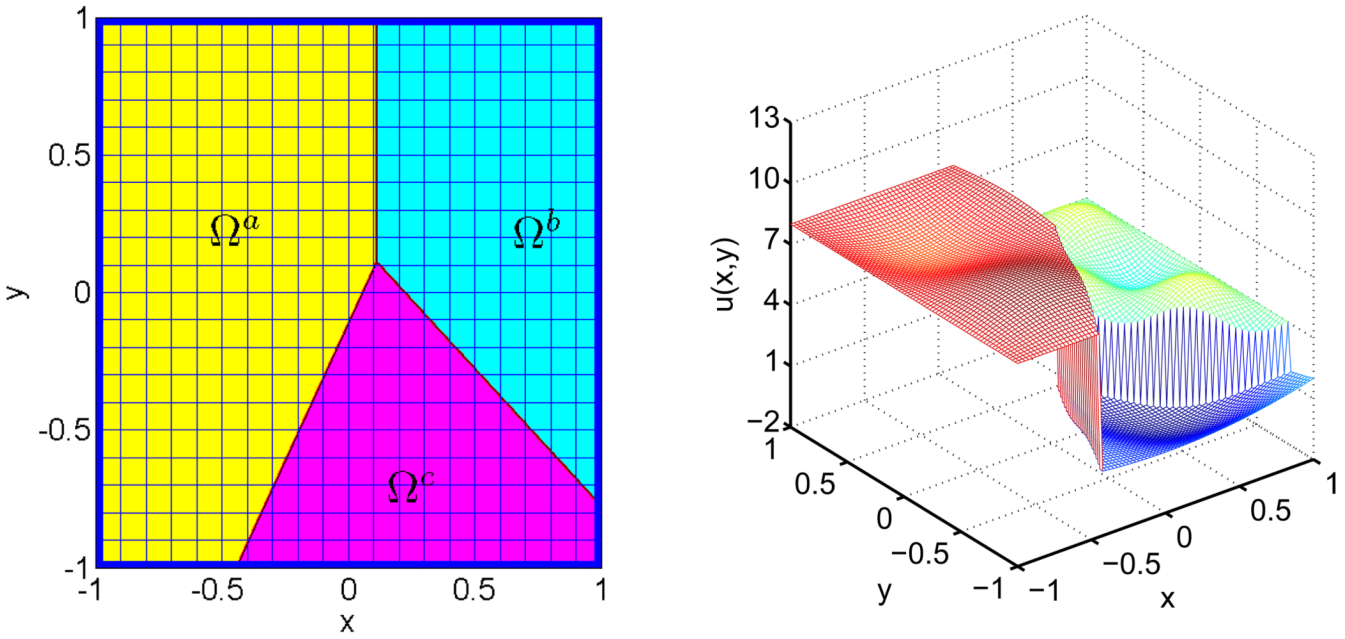

Figure 18.
Basic geometry on 20 × 20 mesh (left) and the computed solution on 80 × 80 mesh (right) for Case 7.

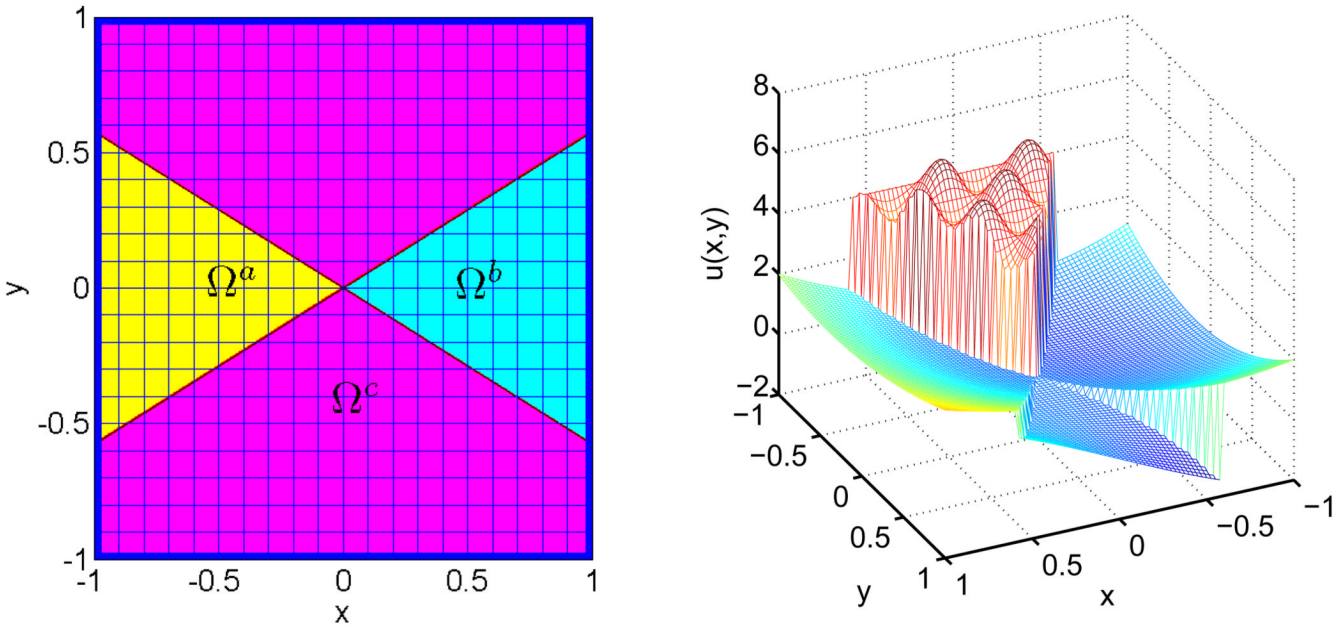

Figure 19.
Basic geometry on 20 × 20 mesh (left) and the computed solution on 80 × 80 mesh (right) for Case 8.

Table 1

Numerical test of new two-material interface schemes (Case 1).

$n_x \times n_y$	L_∞	Order	L_2	Order
20 × 20	3.762e-1		8.605e-2	
40 × 40	6.804e-2	2.47	1.778e-2	2.27
80 × 80	2.413e-2	1.50	6.302e-3	1.50
160 × 160	4.653e-3	2.37	9.777e-4	2.69

Table 2

Numerical test of new two-material interface schemes (Case 2).

$n_x \times n_y$	L_∞	Order	L_2	Order
20 × 20	1.973e-1		3.504e-2	
40 × 40	1.645e-2	3.58	4.666e-3	2.91
80 × 80	2.202e-3	2.90	9.389e-4	2.31
160 × 160	8.049e-4	1.45	3.093e-4	1.60

Table 3

Errors for three-material interface schemes (Case 3(c) and Case 3(d)).

	Case 3(c)				Case 3(d)			
$n_x \times n_y$	L_∞	Order	L_2	Order	L_∞	Order	L_2	Order
20 × 20	7.304e-4		1.179e-4		9.510e-4		2.127e-4	
40 × 40	9.681e-5	2.92	1.279e-5	3.20	2.491e-4	1.93	7.425e-5	1.52
80 × 80	1.198e-5	3.01	3.360e-6	1.93	3.974e-5	2.65	1.296e-5	2.52
160 × 160	3.598e-6	1.74	1.130e-6	1.57	1.021e-5	1.96	3.428e-6	1.92

Table 4

Errors for three-material interface schemes (Case 3(e) and Case 3(f)).

	Case 3(e)				Case 3(f)			
$n_x \times n_y$	L_∞	Order	L_2	Order	L_∞	Order	L_2	Order
20 × 20	6.811e-1		1.665e-1		9.207e-3		3.445e-3	
40 × 40	1.702e-1	2.00	4.013e-2	2.05	2.158e-3	2.09	8.538e-4	2.01
80 × 80	3.707e-2	2.20	7.240e-3	2.47	5.585e-4	1.95	2.290e-4	1.90
160 × 160	6.408e-3	2.53	1.675e-3	2.11	1.490e-4	1.90	6.134e-5	1.90

Table 5

Errors for three-material interface schemes (Case 4).

$n_x \times n_y$	L_∞	Order	L_2	Order
20 × 20	2.334e-1		7.432e-2	
40 × 40	4.102e-2	2.51	1.246e-2	2.58
80 × 80	9.859e-3	2.06	3.116e-3	2.00
160 × 160	2.429e-3	2.02	8.104e-4	1.94

Table 6

Errors for three-material interface schemes (Case 5).

$n_x \times n_y$	L_∞	Order	L_2	Order
20 × 20	2.000e-1		2.924e-2	
40 × 40	5.010e-2	2.00	6.851e-3	2.09
80 × 80	6.168e-3	3.02	1.328e-3	2.37
160 × 160	6.443e-4	3.26	1.621e-4	3.03

Table 7

Errors for three-material interface schemes (Case 6).

$n_x \times n_y$	L_∞	Order	L_2	Order
20 × 20	3.786e-2		1.086e-2	
40 × 40	8.523e-3	2.15	2.824e-3	1.94
80 × 80	2.083e-3	2.03	7.504e-4	1.91
160 × 160	5.492e-4	1.92	2.014e-4	1.90

Table 8

Errors for three-material interface schemes (Case 7).

$n_x \times n_y$	L_∞	Order	L_2	Order
20 × 20	7.466e-2		2.114e-2	
40 × 40	1.334e-2	2.48	3.095e-3	2.77
80 × 80	3.200e-3	2.06	9.461e-4	1.71
160 × 160	7.596e-4	2.07	2.548e-4	1.89

Table 9

Errors for three-material interface schemes (Case 8).

$n_x \times n_y$	L_∞	Order	L_2	Order
20 × 20	1.646e-1		2.536e-2	
40 × 40	4.062e-2	2.02	6.217e-3	2.03
80 × 80	1.057e-2	1.94	1.858e-3	1.74
160 × 160	2.781e-3	1.93	5.038e-4	1.88