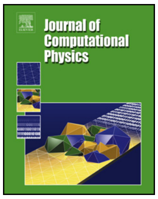
Journal of Computational Physics 477 (2023) 111924

Contents lists available at ScienceDirect

Journal of Computational Physics

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



A FFT accelerated fourth order finite difference method for solving three-dimensional elliptic interface problems

Yiming Ren, Shan Zhao [*]

Department of Mathematics, University of Alabama, Tuscaloosa, AL, 35487, USA

ARTICLE INFO

Article history:
Received 9 August 2022
Received in revised form 3 January 2023
Accepted 5 January 2023
Available online 11 January 2023

Keywords:
Elliptic interface problem
High order central difference schemes
Fast Fourier transform (FFT)
Matched interface and boundary method (MIB)
Mixed boundary conditions
Gradient recovery

ABSTRACT

In this paper, a fourth order augmented matched interface and boundary (AMIB) method is proposed for solving a three-dimensional elliptic interface problem which involves a smooth material interface inside a cuboid domain. On the boundary of the cuboid domain, the fourth order AMIB method can handle different types of boundary conditions, including Dirichlet, Neumann, Robin and their mix combinations in fictitious value generation. Moreover, zero-padding solutions are introduced so that the fast Fourier transform (FFT) algorithm is still valid near the boundary. In dealing with the interior interface, a fourth order ray-casting matched interface and boundary (MIB) scheme is proposed, which enforces the jump conditions along the normal direction for calculating fictitious values. Comparing with the existing MIB scheme, the ray-casting scheme naturally bypasses the corner issue and becomes more robust in handling complex geometry. Based on fictitious values generated near interface and boundary, the fourth order central difference can be corrected at various irregular points including corner points, by introducing Cartesian derivative jumps as auxiliary variables. This gives rise to an enlarged linear system, which can be efficiently solved by the Schur complement procedure together with the FFT inversion of the discrete Laplacian. Extensive numerical experiments have been carried to test the proposed ray-casting AMIB method for numerical accuracy, efficiency, and robustness in corner treatment. The numerical results demonstrate that the ray-casting AMIB scheme not only maintains a fourth order of accuracy in treating various interfaces and boundaries for both solutions and solution gradients, but also attains an overall efficiency on the order of $O(n^3 \log n)$ for a $n \times n \times n$ uniform grid.

1. Introduction

This paper focuses on solving three-dimensional (3D) elliptic interface problems with discontinuous coefficients. We consider an elliptic partial difference equation (PDE) in a domain Ω

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

subject to boundary conditions in a generic form on the boundary $\partial\Omega$,

$$\alpha_\Gamma u + \beta_\Gamma \frac{\partial u}{\partial n} = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega. \tag{2}$$

---

[*] Corresponding author.
E-mail address: szhao@ua.edu (S. Zhao).

https://doi.org/10.1016/j.jcp.2023.111924

Equation (2) represents three commonly used boundary conditions, i.e., Dirichlet ($\alpha_\Gamma \neq 0, \beta_\Gamma = 0$), Neumann ($\alpha_\Gamma = 0, \beta_\Gamma \neq 0$), and Robin ($\alpha_\Gamma \neq 0, \beta_\Gamma \neq 0$). For simplicity, the 3D domain Ω is assumed to be cuboid. A lower dimensional interface Γ defined by $\Gamma = \Omega^+ \cap \Omega^-$ divides the computational domain Ω into disjoint subdomains $\Omega = \Omega^+ \cup \Omega^-$. An illustration of subdomains is given in Fig. 1 (a). In Eq. (1), the coefficient κ of the reaction term is a constant. The coefficient β and source term $f(\mathbf{x})$ are continuous on each disjoint subdomain, but may be discontinuous across the interface Γ. Furthermore, β is assumed to be a piecewise constant, that is, $\beta = \beta^+$ in Ω^+ and $\beta = \beta^-$ in Ω^-, where β^+ and β^- are some positive constants. Similarly, the discontinuous source term $f(\mathbf{x})$ is denoted as $f^+(\mathbf{x})$ and $f^-(\mathbf{x})$, respectively, in Ω^+ and Ω^-. Across the interface Γ, two jump conditions are known for the function and its flux in the normal direction

$$[\![u]\!] := u^+ - u^- = \phi(\mathbf{x}), \tag{3}$$

$$[\![\beta u_n]\!] := \beta^+ \nabla u^+ \cdot \vec{n} - \beta^- \nabla u^- \cdot \vec{n} = \psi(\mathbf{x}), \tag{4}$$

where $\vec{n}$ is the outward normal direction of Γ pointing from Ω^- to Ω^+, and the superscript stands for the limiting value from each side of the interface. Equations (3) and (4) are called as the zeroth and first order jump conditions. Such an elliptic interface problem with discontinuous coefficients has wide application in a variety of fields such as fluid dynamics [34], material science [27], and biological science [23].

Numerous studies have been devoted to algorithm development and numerical analysis for elliptic interface problems in the past several decades. In order to solve the elliptic interface problems numerically, the computational grid can be either fitted or unfitted with the interface. Classical finite element method (FEM) [2,5,10,36] can achieve a satisfactory accuracy if the interfaces are well fitted by the underlying meshes. However, it may be a costly and nontrivial process to generate meshes for interfaces with complicated geometry. Therefore, there is always a practical need to develop systematic ways for constructing numerical methods on unfitted meshes. Within the FEM context, this motivates the development of immersed finite element (IFE) method [28]. The basic idea of the IFE is modifying the basis functions of the cut-through elements based on jump conditions. Similar technique has been applied to the immersed finite volume (IFV) method [42]. There are other FEMs developed for elliptic interface problems, such as multiscale finite element method [12], extended finite element method [22,29], and so on.

There has been a great deal of effort in developing finite difference methods on unfitted meshes [44,49,33,15,43,14]. The key idea is to modify finite difference weights for nodes near the interface, so that jump conditions are approximately satisfied. For instance, the immersed boundary method (IBM) introduced by Peskin is known to be a key trigger in this field for simulation of flow pattern of blood in the heart [44]. Albeit being typically first order accurate in higher dimensions, the IBM is quite flexible, robust and efficient. The immersed interface method (IIM) proposed by LeVeque and Li [33] is the first second-order finite difference method, where the Taylor series expansions are used to assign stencil weights. Another popular approach is the ghost fluid method (GFM) [15], originally designed to treat contact discontinuities in the inviscid Euler equations. In solving elliptic interface problems, the GFM is typically first-order [43] and has been extended to second order in [41]. The recovery of flux convergence of the GFM has been studied in [14]. A review of a second order compact finite difference method is offered in [9]. In [54,55], a second order accurate matched interface and boundary (MIB) scheme has been developed for treating complex interfaces with geometric singularities, in which jump conditions are enforced to generate necessary fictitious values for central differences. In [51], the MIB fictitious values have been applied to the Galerkin formulation to formulate a second-order accurate FEM for complex interfaces such as protein surfaces. A second-order generalized finite difference method has been developed in [52] based on Taylor series expansions, which can be regarded as a meshless method. Apart from finite element and finite difference methods, other effective algorithms for solving elliptic interface problems include virtual node method [3,26], finite volume method [4], and coupling interface method [11,47]. We note that the aforementioned methods usually deliver first or second order accuracy.

Despite of a great success achieved in the numerical solution of interface problems, there are still a few remaining challenges in the field. One of such challenges is the development of high order (at least third order) interface methods. Although high order discretizations are demanded for many real-world problems, e.g. those associated with high frequency waves, only a few studies have been conducted in the literature to advance along this direction. A fourth order method can be constructed by discretizing high order jump conditions and mixed derivatives in the IIM formulation [38]. By iteratively imposing zeroth and first order jump conditions, the MIB method [60,54,55] introduces fictitious points to support high order central differences across the interface. Being arbitrarily high-order accuracy in principle, up to 16th order MIB scheme has been proposed with straight interface geometry [60], and sixth-order MIB schemes have been demonstrated for smooth interfaces in two-dimensional (2D) and three-dimensional (3D) domains [60,54,55]. In [59], a fourth order IIM was developed by using zero and first order jump conditions too. By combining IBM with a discontinuous Galerkin spatial discretization, a high-order method has been constructed for elliptic problems with discontinuous coefficients and singular sources [7]. In [30], the idea of the IIM is combined with a continuous finite element discretization to derive a high order finite element method for a class of elliptic problems with jumps in the solution and its flux across smooth interfaces. Recently, a sixth order compact finite difference scheme has been constructed in [19] and has been generalized in [20] for solving elliptic interface problems with discontinuous and high-contrast coefficients. By embedding the governing PDE into local Taylor series expansions, this compact finite difference scheme achieves the fourth order accuracy in solving elliptic interface problems. Recently, a fourth-order reproducing kernel method for one-dimensional elliptic interface problems has been introduced in [53]. Besides these elegant interface schemes, there are many attractive high order numerical methods introduced in the literature for elliptic boundary value problems over irregular domains, see for example [40,24,21,46].

Besides the accuracy improvement, another challenge concerns the construction of fast algorithms for elliptic interface problems. It is worth noting that for elliptic PDE without interfaces, the acceleration of algebraic computations can be accomplished by using fast Poisson solvers, which include geometric multigrid with a complexity $O(N)$ and fast Fourier transform (FFT) with a complexity $O(N \log N)$, where N is the spatial degree of freedom. The multigrid acceleration has been successfully applied in several interface algorithms, such as ghost point method [13], piecewise-polynomial interface method [8], P_1 conforming finite element methods on a semi-uniform grid [32], IFE [31] and IIM [1]. The use of the FFT Poisson solver has also gained a great popularity in accelerating algebraic solution of elliptic interface problems [37,50,16,18].

A major advancement in this field is the augmented IIM (AIIM) [37,39], whose formulation consists of two main components. First, by introducing auxiliary variables, the Laplacian operator can be approximated by the standard finite difference stencil, which results in a symmetric and diagonally dominant matrix for fast Poisson solver. Second, the auxiliary variables are solved iteratively either through the Schur complement method [37,39,16,50] or from the previous iterative step [6]. For instance, AIIM was originally explored by Li in 1998 for solving elliptic interface problems in the case of a piecewise constant β [37], by using the FFT. Since then, the AIIM has found a great success in a wide range of applications [38] and has been extended to variable coefficients elliptic interface problems [39]. Recently, a new AIIM has been introduced for 3D elliptic interface problems with a piecewise constant [56]. In the AIIMs, solution jumps in the normal derivative, i.e., $[\![u_n]\!]$, are usually chosen as auxiliary variables. Alternatively, some other AIIMs utilized another type of auxiliary variables, i.e., Cartesian derivative jumps $[\![u_x]\!], [\![u_{xx}]\!], [\![u_y]\!]$ and $[\![u_{yy}]\!]$. For instance, with these auxiliary variables, jump corrected Taylor expansions can be obtained near the jumps, giving rise to the construction of an explicit jump immerse interface method (EJIIM) [50] and a decomposed immersed interface method (DIIM) [6] for solving interface problems with piecewise constant and variable coefficients.

Recently, an augmented matched interface and boundary (AMIB) method has been introduced for two-dimensional (2D) elliptic interface problems [18], which is the first known finite difference method that combines fourth order convergence with the $O(N \log N)$ efficiency. In the AMIB method, finite difference discretization is modified separately for interface and boundary. The classical MIB scheme [60] is employed to treat curved interfaces up to fourth order accuracy. On boundaries of a cuboid domain, the MIB scheme [57,17] can achieve arbitrarily high order in handling the Dirichlet, Neumann, and Robin boundary conditions and their mixed combination. By using the augmented formulation introduced in the first AMIB method [16], the Cartesian derivative jumps are employed as auxiliary variables and are reconstructed via fictitious values at both interfaces and boundaries. In the Schur complement solution of the augmented system, the FFT algorithm can be applied to efficiently invert the discrete Laplacian. Consequently, the AMIB scheme [18] can not only provide fourth order accurate approximation to solution and its gradient, but also produces an overall complexity of $O(N \log N)$.

The goal of this paper is to develop a fourth order AMIB method with the FFT acceleration for solving three-dimensional (3D) elliptic interface problems. The new AMIB method is not a simple generalization of the previous AMIB method [16,18] to 3D. Instead, two new developments will be carried out.

First, a novel ray-casting MIB scheme is proposed for solving interface problems in 2D and 3D. In the classical MIB scheme for 2D interface treatment, a tangential derivative jump condition will be derived first [60]. Then the tangential jump condition and the zeroth and first order jump conditions will be decomposed into Cartesian directions, e.g. x and y directions in 2D. These Cartesian conditions will be discretized repeatedly to generate necessary fictitious values across the interface. Such Cartesian MIB scheme has been constructed in 3D [55], and its second order version has found a great success in real applications [23]. Motivated by the ray-casting MIB for solving elliptic boundary value problems over irregular domains [58,46], in the proposed ray-casting MIB scheme, only zeroth and first order jump conditions will be employed in fictitious value generation in a one-dimensional (1D) manner along the normal direction. This is considerably simpler than the Cartesian MIB scheme [60,54,55], for which jump condition discretization involves all three Cartesian directions in 3D. Moreover, because the normal direction makes it possible to reach more grid points inside/outside the interface, compared to the Cartesian directions, the ray-casting MIB scheme becomes more robust in handling complex geometry.

Second, the augmented formulation of the AMIB scheme [16,18] will be further improved. In particular, in the approximation of auxiliary variables, i.e., Cartesian derivative jumps, a rigorous corner treatment will be conducted, so that the new AMIB scheme can accommodate more complicated geometries. A similarly corner treatment has been considered in [35] for solving parabolic PDEs over irregular domains. The present study will extend the corner treatment to interface problems, i.e., considering possible corners in both Ω^- and Ω^+.

Several nice features of the original AMIB scheme [16,18] will be maintained. For instance, at the boundary $\partial\Omega$, the same fourth order AMIB method [17] will be utilized for imposing Dirichlet, Neumann, Robin or any combination of boundary conditions. The augmented scheme [16] is able to incorporate both interior interface and exterior boundary treatments into one formulation. This gives rise to a fourth order finite difference method with the FFT efficiency for solving 3D elliptic interface problems with various boundary conditions. Moreover, like the 2D AMIB scheme [18], a fourth order of convergence can be fulfilled in the approximation of solution gradients or fluxes for 3D elliptic interface problems, by using the calculated numerical solutions and fictitious values. The numerical order of gradient approximation is usually one order lower than that of solution itself. This motivated many studies in the literature for gradient recovery, see for example [39,48,14,25]. The AMIB method can provide accurate gradient approximation without additional recovery.

The rest of the paper is organized as follows. In Section 2, a new ray-casting MIB scheme is proposed to solve interface problems in 2D and 3D. The boundary treatment will be described. Then a uniform augmented MIB system will be formulated with fast computation. Section 3 is dedicated to the numerical results to demonstrate the performance of the proposed algorithm. A conclusion and future plan will be given at the end of this paper.

2. Theory and algorithm

To introduce the proposed AMIB scheme, we first transform the original PDE (1) by dividing the coefficient β on both hand sides of the equation

$$\Delta u - \kappa u = -\frac{f(\mathbf{x})}{\beta}, \quad \mathbf{x} \in (\Omega^- \cup \Omega^+) \setminus \Gamma. \tag{5}$$

We will solve (5) subject to the boundary condition (2) and interface jump conditions (3) and (4), which actually gives the solution of the original elliptic interface problem.

We concern ourselves on a cuboid domain $\Omega = [a,b] \times [c,d] \times [e,f]$, which is separated as $\Omega = \Omega^+ \cup \Omega^-$ by a closed interface Γ, as shown in Fig. 1 (a). A uniform grid spacing h is employed to partition the domain Ω into n_x, n_y and n_z equally spaced intervals in the x-, y- and z-directions respectively such that the $n_x = (b-a)/h$, $n_y = (d-c)/h$ and $n_z = (f-e)/h$. The grid coordinates in 3D are therefore defined as

$$x_i = a + ih, \; y_j = c + jh, \; z_k = e + kh, \; i = 0, \cdots, n_x, \; j = 0, \cdots, n_y, \; k = 0, \cdots, n_z. \tag{6}$$

The details of the proposed AMIB method will be discussed in subsection 2.1 - 2.4. Similar to the 2D AMIB scheme [18], we will first introduce an immersed boundary problem so that high order approximation and FFT inversion can be carried out near boundaries. Then, a new ray-casting MIB scheme is proposed to generate fictitious values near the interface, while the same MIB boundary scheme [17] is applied at boundaries. The corrections to fourth order finite difference are considered next, and particular attention will be paid for corner treatments. Finally, the augmented formulation and FFT inversion [16] will be established.

2.1. Immersed boundary formulation

In the proposed AMIB method, the standard fourth order central difference will be utilized for approximating the second order partial derivatives with a truncation error $O(h^4)$, e.g.,

$$\begin{aligned} u_{xx}(x_i, y_j, z_k) \approx \frac{1}{h^2}[-&\frac{1}{12}u(x_{i-2}, y_j, z_k) + \frac{4}{3}u(x_{i-1}, y_j, z_k) - \frac{5}{2}u(x_i, y_j, z_k) \\ &+ \frac{4}{3}u(x_{i+1}, y_j, z_k) - \frac{1}{12}u(x_{i+2}, y_j, z_k)] \end{aligned} \tag{7}$$

Near the boundary $\partial\Omega$, boundary conditions need be imposed to modify finite difference weights. Moreover, in order apply the FFT Poisson solver throughout the domain for central difference, an immersed boundary formulation has been proposed in [17], by adding several layers of zero-padding solutions beyond $\partial\Omega$. Then, over the boundary of the enlarged domain, the anti-symmetric property is naturally satisfied so that the FFT fast inversion is feasible. Similarly, we will first convert the present 3D boundary $\partial\Omega$ into an immersed boundary.

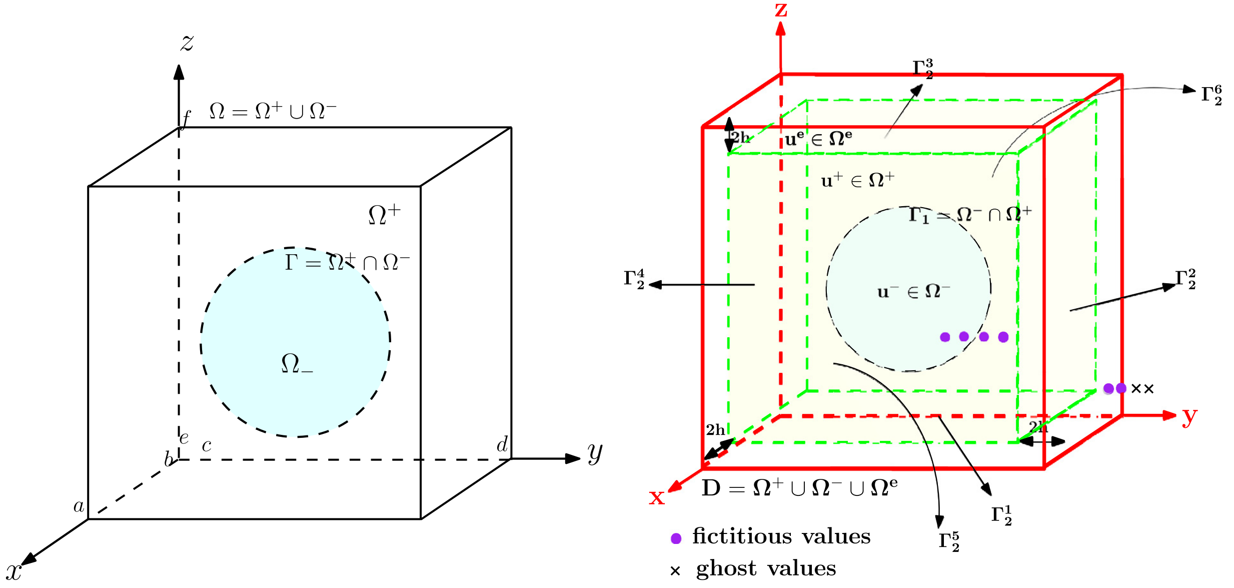
Now we take the fourth order central difference scheme in 3D for a demonstration. As shown in Fig. 1, the original problem is recasted into an immersed boundary problem. To be more clear, the original cuboid domain Ω is embedded in a larger cuboid region D such that the extended domain is equipped with an external subdomain Ω^e with a width being $2h$ for x-, y-, and z-direction to support the fourth order central difference. As a consequence, the grid coordinates are redefined as below in the domain D

$$\begin{aligned} &x_i = a + (i-2)h, \quad y_j = c + (j-2)h, \quad z_k = e + (k-2)h, \\ &\qquad i = 0, \cdots, n_x + 4, \; j = 0, \cdots, n_y + 4, \; k = 0, \cdots, n_z + 4 \end{aligned} \tag{8}$$

where n_x, n_y, n_z are identically defined as above.

From now on, the original interface Γ is redefined as Γ_1. We denote the extended subdomain as Ω^e, and separate the whole domain $D = [a-2h, b+2h] \times [c-2h, d+2h] \times [e-2h, f+2h]$ as $D = \Omega^- \cup \Omega^+ \cup \Omega^e$ by two closed interfaces $\Gamma_1 = \Omega^- \cap \Omega^+$ and $\Gamma_2 = \Omega^+ \cap \Omega^e$. Recall that the original solution u and the source term f can be piecewisely defined, i.e., we have u^- and f^- in Ω^- and u^+ and f^+ in Ω^+. In the extended subdomain Ω^e, we simply have $u^e = 0$ and $f^e = 0$. With these notations, the immersed boundary problem can be modeled as

$$\Delta u - \kappa u = -\frac{f(\mathbf{x})}{\beta}, \quad \mathbf{x} \in D \setminus (\Gamma_1 \cup \Gamma_2), \tag{9}$$



(a) Original interface problem (b) Immersed boundary problem

Fig. 1. A demonstration of the original and corresponding immersed boundary problems. In the enlarged 3D domain D, Γ_1 stands for the original interface Γ, and Γ_2 is composed of six sides of the original cuboid domain Ω, i.e., Γ_2^i for $i = 1, \cdots, 6$.

with source term being

$$f = \begin{cases} f^-, & \mathbf{x} \in \Omega^-, \\ f^+, & \mathbf{x} \in \Omega^+, \\ f^e, & \mathbf{x} \in \Omega^e, \end{cases}$$

and coefficient defined as

$$\beta = \begin{cases} \beta^-, & \mathbf{x} \in \Omega^-, \\ \beta^+, & \mathbf{x} \in \Omega^+, \\ 1, & \mathbf{x} \in \Omega^e. \end{cases}$$

The new immersed boundary problem is subject to the original interface conditions (3) - (4) on $\Gamma_1 = \Gamma$ and the boundary condition (2) on the immersed interface $\Gamma_2 = \partial\Omega$. In the case of Dirichlet boundary conditions, Eq. (2) can be converted to additional interface condition (10) on Γ_2:

$$[\![u]\!] := u^e - u^+ = g(\mathbf{x}), \tag{10}$$

because $u^e = 0$ in the zero-padding zone Ω^e. On the extended boundary ∂D, a trivial boundary condition can be assumed $u = u^e = 0$. Moreover, the ghost values outside ∂D can be obviously assumed to be zero. Consequently, the anti-symmetry property in u is naturally satisfied across ∂D, so that the FFT inversion can be applied over D [17].

In our computation, one constraint we require is that the shortest distance between Γ_1 and Γ_2 must be $4h$ in each Cartesian direction to allow for fourth order of convergence. Otherwise, the MIB treatments for Γ_1 and Γ_2 might affect each other.

2.2. Fictitious values generation

To compensate the solution discontinuity near Γ_1 and Γ_2, special numerical treatments are needed. In the MIB method, this is accomplished through introducing necessary irregular points and fictitious values [60]. Briefly speaking, while the standard fourth order central difference Eq. (7) could be employed for approximation at regular points, modification is required for irregular points where finite difference is not well-defined due to the discontinuous solution across Γ_1 and Γ_2. For this purpose, we first define irregular points near Γ_1 and Γ_2 in the fourth order case below.

Firstly, we define irregular point near Γ_1. Assume that the interface Γ_1 is governed by the zero level set $\Gamma_1 = \{(x, y, z), \varphi(x, y, z) = 0\}$, with $\varphi(x, y, z) < 0$ in Ω^- and $\varphi(x, y, z) > 0$ in $\Omega^+ \cup \Omega^e$. Define minimal and maximal level set values at a node (x_i, y_j, z_k) over a set of values

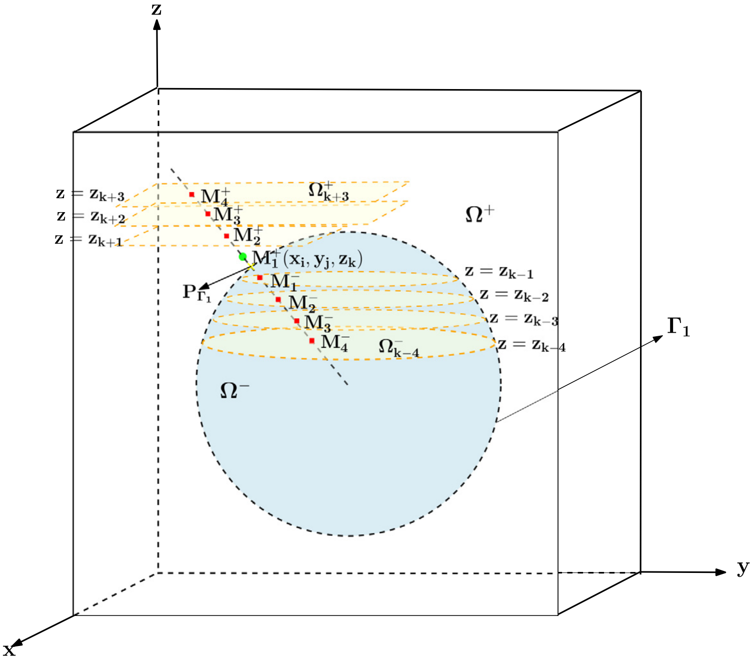


Fig. 2. In the ray-casting MIB scheme, the desired fictitious value at $M_1^+ = (x_i, y_j, z_k)$ (green filled circle) is generated along the normal line passing M_1^+ (dashed line), involving 8 auxiliary points M_l^- and M_l^+ for $l = 1, 2, 3, 4$ (red squares and green filled circle). Except for M_1^+, each auxiliary point is an intersection point of the normal line with a nearby xy plane. Moreover, this auxiliary point will be interpolated within certain region on the xy plane. Inside Γ_1, the interpolation is conducted within Ω^- on the xy plane $z = z_l$, giving rise to yellow disks Ω_l^- for $l = k-4, k-3, k-2, k-1$. Without restriction outside Γ_1, Ω_l^+ is taken as a rectangular (yellow) region on the xy plane $z = z_l$ for $l = k+1, k+2, k+3$. (For interpretation of the colors in the figures, the reader is referred to the web version of this article.)

$$\{\varphi_{i-2,j,k}, \varphi_{i-1,j,k}, \varphi_{i,j,k}, \varphi_{i+1,j,k}, \varphi_{i+2,j,k}, \varphi_{i,j-2,k}, \varphi_{i,j-1,k}, \varphi_{i,j+1,k}, \varphi_{i,j+2,k}, \varphi_{i,j,k-2}, \varphi_{i,j,k-1}, \varphi_{i,j,k+1}, \varphi_{i,j,k+2}\} \tag{11}$$

as φ_{ijk}^{min} and φ_{ijk}^{max}. When $\varphi_{ijk}^{min}\varphi_{ijk}^{max} < 0$, we call the grid node (x_i, y_j, z_k) an irregular point, otherwise regular point. In particular, consider an irregular point (x_i, y_j, z_k) near Γ_1 and within Ω^-. We have $\varphi_{i,j,k} < 0$ and at least one point in the set of Eq. (11) has positive value, say $\varphi_{I,J,K} > 0$. To approximate the Laplacian Δu at (x_i, y_j, z_k), the use of the function value $u(x_I, y_J, z_K)$ in the fourth order central difference (7) will suffer an accuracy reduction, because (x_I, y_J, z_K) is from the other side of the interface. In the MIB method, a fictitious value will be assumed at (x_I, y_J, z_K), which can be regarded as the extension of the solution from the Ω^- side, but is determined rigorously according to jump conditions [60]. Therefore, in order to recover fourth order central difference approximation at all irregular points near Γ_1, four layers of fictitious points surrounding Γ_1 will be needed along each Cartesian grid line. Referring to Fig. 1 (b), four fictitious points near Γ_1 are highlighted along y-direction grid line, two inside and two outside. Other fictitious points along x direction or z direction can be generated in a similar fashion.

Secondly, irregular points near interface Γ_2 can be defined as the grid nodes on Γ_2, two layers of grid nodes outside Γ_2, and the nearest one layer of grid points inside Γ_2. Similarly, four layers of fictitious values can be assumed near Γ_2, two inside and two outside. However, the two layers inside are actually not needed in our computation, because the corresponding fictitious values trivially equal to zero for the zero-padding solution u^e. Thus, only two fictitious points outside Γ_2 are highlighted along y-direction grid line in Fig. 1 (b). These two layers of fictitious values will be determined according to boundary conditions [57].

2.2.1. *Fictitious values near interface Γ_1*

In the present study, a novel ray-casting MIB scheme will be developed for handling the curved interface Γ_1 in 3D. The new scheme consists of two main stages. First, the jump conditions (3) and (4) are imposed along the normal direction to determine necessary fictitious values. Second, the involved points will be interpolated on 2D Cartesian planes so that the fictitious values can be expressed by means of function values of u on a set of grid nodes. The ray-casting MIB scheme can be similarly formulated for treating 2D interfaces.

In the first stage of the ray-casting MIB scheme, we determine the fictitious values around Γ_1. A particular case of the ray-casting MIB scheme is illustrated in Fig. 2. Consider an irregular point (x_i, y_j, z_{k-1}) in Ω^-. Assume that the fourth order finite difference approximation at (x_i, y_j, z_{k-1}) along the z direction needs two fictitious values at (x_i, y_j, z_k) and (x_i, y_j, z_{k+1}) in Ω^+. Here, details will be provided on how to determine the first layer fictitious value outside Γ_1, i.e., at

$M_1^+ = (x_i, y_j, z_k)$ as shown in Fig. 2. The second layer fictitious value outside Γ_1, e.g. at (x_i, y_j, z_{k+1}), and two layers of fictitious values inside Γ_1 can be similarly generated.

For the fictitious node M_1^+, one first finds a normal line of Γ_1 that passes $M_1^+ = (x_i, y_j, z_k)$, similar to the immersed interface method (IIM) [38]. For a smooth interface Γ_1, such a normal direction can always be found, and we assume the normal line intersects Γ_1 at an interface point P_{Γ_1}. Along the inward normal direction, the normal line or auxiliary line will first meet a 2D Cartesian grid plane, which could be xy plane, xz plane, or yz plane. Without the loss of generality, let us assume that the normal line first meets a xy plane $z = z_{k-1}$ and denote the intersection point as M_1^-, see Fig. 2. Extending the auxiliary line in both directions, it will meet other xy planes, yielding three auxiliary nodes M_l^+ for $l = 2, 3, 4$ on three horizontal xy plane $z = z_{k+l}$ for $l = 1, 2, 3$ positioned in the same side of Γ_1 as M_1^+, i.e., Ω^+. Similarly four auxiliary nodes M_l^- for $l = 1, 2, 3, 4$ on four horizontal xy plane $z = z_{k-l}$ for $l = 1, 2, 3, 4$ are positioned in the other side of Γ_1, i.e., Ω^-.

We next discretize the zeroth and first jump conditions (3) and (4) at P_{Γ_1} by using eight auxiliary points $\{M_4^-, M_3^-, M_2^-, M_1^-, M_1^+, M_2^+, M_3^+, M_4^+\}$. With two conditions, this discretization allows us to determine two unknowns, i.e., two fictitious values at M_1^+ and M_1^-, which will be denoted as $\hat{u}_{M_1^+} = \hat{u}_{i,j,k}$ and $\hat{u}_{M_1^-}$, respectively. Note that $\hat{u}_{i,j,k}$ is the fictitious value needed in the fourth order central difference approximation at (x_i, y_j, z_{k-1}). The additional fictitious value $\hat{u}_{M_1^-}$ is not used in this study, but may be employed in future studies to accommodate more complex geometries. Denote the function values at M_l^+ and M_l^- as $u_{M_l^+}$ and $u_{M_l^-}$ for $l = 1, 2, \cdots, 4$. Each limiting value in the interface conditions (3) and (4) is approximated by using four function values from the same side and one fictitious value from the other side. Neglecting the truncation errors on the order of $O(h^4)$, this yields two algebraic equations

$$w_{0,1}^+ \hat{u}_{M_1^-} + \sum_{l=1}^{4} w_{0,l+1}^+ u_{M_l^+} - (w_{0,1}^- \hat{u}_{i,j,k} + \sum_{l=1}^{4} w_{0,l+1}^- u_{M_l^-}) = \phi(P_{\Gamma_1}), \tag{12}$$

$$\beta^+ (w_{1,1}^+ \hat{u}_{M_1^-} + \sum_{l=1}^{4} w_{1,l+1}^+ u_{M_l^+}) - \beta^- (w_{1,1}^- \hat{u}_{i,j,k} + \sum_{l=1}^{4} w_{1,l+1}^- u_{M_l^-}) = \psi(P_{\Gamma_1}), \tag{13}$$

where $w_{m,l+1}^+$ and $w_{m,l+1}^-$ for $m = 0, 1$ and $l = 0, 1, \cdots, 4$ represent the finite difference weights. Here the subscript m stands for zeroth $(m = 0)$ or first order derivative $(m = 1)$ approximation at the interface point P_{Γ_1}. The superscripts $-$ and $+$ in w signify the Ω^- and Ω^+ domain separated by the interface Γ_1. Therefore, by solving two equations (12) and (13) algebraically, one obtains two fictitious values $\hat{u}_{i,j,k}$ and $\hat{u}_{M_1^-}$ by means of a linear combination of $u_{M_l^+}$ and $u_{M_l^-}$ for $l = 1, 2, \cdots, 4$, and two jump values at the interface point P_{Γ_1}, i.e., $\phi(P_{\Gamma_1})$ and $\psi(P_{\Gamma_1})$.

In the second stage of the ray-casting MIB scheme, the seven auxiliary points, M_l^- for $l = 1, 2, 3, 4$ and M_l^+ for $l = 2, 3, 4$, have to be interpolated by using nearby grid node values, because these auxiliary points are usually off-grid. Note that the function value at the auxiliary point M_1^+ is on-grid, i.e., $u_{M_1^+} = u_{i,j,k}$. To guarantee the accuracy, each of the seven auxiliary points will be interpolated by using grid nodes from the same side of the interface Γ_1. Moreover, such grid nodes are also on the same 2D Cartesian plane, i.e., xy plane in the present discussion. In this manner, seven 2D auxiliary regions are identified as shown in Fig. 2.

We next consider the interpolation of an auxiliary point within the corresponding 2D auxiliary region. A set of appropriate grid nodes in the auxiliary region will be chosen so that the interpolation error is on the order of $O(h^4)$. Without the loss of generality, let us take for instance the interpolation of the auxiliary point M_1^- restricted in a 2D auxiliary region Ω_{k-1}^- on the horizontal plane $z = z_{k-1}$. To better illustrate the interpolation process, a 2D plot is given in Fig. 3. Here the auxiliary point M_1^- is assumed to have coordinate values (x_o, y_o, z_{k-1}) for $x_{m-1} < x_o < x_m$ and $y_{l+1} < y_o < y_{l+2}$. The 2D interpolation will be carried out through two 1D interpolation steps. In the first step, we need to choose an interpolation line from $x = x_o$ or $y = y_o$, based on the coordinate values x_o and y_o. For the case shown in Fig. 3, the line $y = y_o$ is chosen. Then five interpolation points along this interpolation line, which are closest to M_1^- while within Ω_{k-1}^-, are selected, i.e., (x_{m-1}, y_o, z_{k-1}), (x_m, y_o, z_{k-1}), (x_{m+1}, y_o, z_{k-1}), (x_{m+2}, y_o, z_{k-1}), (x_{m+3}, y_o, z_{k-1}) in Fig. 3. Next, each of these five interpolation point will be interpolated or extrapolated by five closest grid nodes along y direction within Ω_{k-1}^-. This totally selects 25 grid nodes to interpolate M_1^- within Ω_{k-1}^-, as shown in Fig. 3. The interpolation represents $u_{M_1^-}$ by means of 25 function values $u_{I,J,K}$ at 25 grid nodes. The interpolation of other auxiliary values can be conducted similarly.

With the seven auxiliary function values approximated properly, the formulation of fictitious value $\hat{u}_{i,j,k}$ can finally take a general form as

$$\hat{u}_{i,j,k} = \sum_{(x_I, y_J, z_K) \in \mathbb{S}_{i,j,k}} W_{I,J,K} u_{I,J,K} + W_0 \phi(P_{\Gamma_1}) + W_1 \psi(P_{\Gamma_1}), \tag{14}$$

where $\mathbb{S}_{i,j,k}$ represents the set of grid nodes selected in the ray-casting MIB scheme, and $W_{I,J,K}$ is the weight associated with each point in the set. In particular, this set involves 25 grid nodes for each of seven auxiliary points, as well as $M_1^+ = (x_i, y_j, z_k)$. Thus, in total, the fictitious value $\hat{u}_{i,j,k}$ is a linear combination of 176 function values on 176 Cartesian nodes, and two known jump values $\phi(P_{\Gamma_1})$ and $\psi(P_{\Gamma_1})$. This fictitious value representation is guaranteed to have an accuracy on

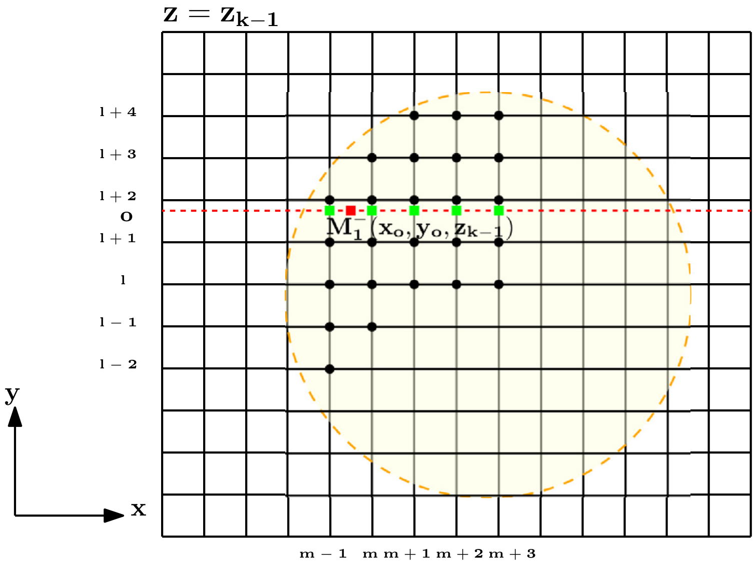


Fig. 3. The interpolation of an auxiliary point M_1^- within the 2D auxiliary region Ω_{k-1}^-. The 2D interpolation is carried out through two 1D steps. First, M_1^- (red square) is interpolated by using five interpolation points (green squares) at (x_{m-1}, y_o, z_{k-1}), (x_m, y_o, z_{k-1}), (x_{m+1}, y_o, z_{k-1}), (x_{m+2}, y_o, z_{k-1}), and (x_{m+3}, y_o, z_{k-1}). Second, each interpolation point is interpolated in y direction by five grid nodes (black filled circles).

the order of $O(h^4)$. In a similar manner, one can generate all necessary fictitious values around Γ_1, two layers inside and two layers outside. We note that the generation processes of all fictitious values are decoupled or independent.

The present ray-casting MIB scheme is formulated such that the 176 Cartesian nodes in $\mathbb{S}_{i,j,k}$ are chosen to minimize the involved interpolation and derivative approximation errors, while maintaining the $O(h^4)$ requirement. Some details about the selection of $\mathbb{S}_{i,j,k}$ for different interface geometries are discussed below.

1. For each normal line or auxiliary line, there are three scenarios depending on whether the auxiliary line intersects with the xy plane or xz plane or yz plane first. Consider the vector $\vec{v} = \overrightarrow{P_{\Gamma_1} M_1^+}$, and normalize it to calculate its x, y, and z components, i.e., $\frac{\vec{v}}{|\vec{v}|} = (d_x, d_y, d_z)^T$. Computationally, these three scenarios can be identified by examining the maximal value among $|d_x|$, $|d_y|$, and $|d_z|$. For example, if $|d_z|$ is the largest one, the auxiliary line will first meet the xy plane, as shown in Fig. 2. Alternatively, the auxiliary line first meets the yz plane and xz plane, respectively, if $|d_x|$ and $|d_y|$ is the largest value.
2. The auxiliary regions Ω_l^- and Ω_l^+ for $l = 1, 2, 3, 4$ are selected adaptively in the ray-casting MIB algorithm. As shown in Fig. 3, 25 grid nodes will be chosen from each auxiliary region to interpolate the auxiliary point. However, for complex geometries, there are situations in which no enough grid nodes are available for interpolating the auxiliary point within the auxiliary region. For the geometry shown in Fig. 2, this could happen for certain Ω_l^- in Ω^-. In such a case, the auxiliary point M_l^- shall be skipped. Then, a new auxiliary point will be considered, which is taken as the intersection point of the normal line with the next xy plane below $z = z_{k-4}$, and the corresponding auxiliary region will be determined. Such a down shifting process is allowed to be conducted at most twice to guarantee that there are enough nodes available at each auxiliary region for interpolating the auxiliary points. In order to deal with more complicated geometry, we plan to explore other strategies in the future, such as conducting the down shifting for more layers, or conditionally accepting an auxiliary region with less than 25 nodes for interpolating the auxiliary point.
3. The adaptive procedure could also be considered in calculating the second layer of fictitious values. For example, image in Fig. 2 that there is a fictitious point (x_i, y_j, z_{k+1}) in Ω^+. The closest auxiliary point generated in the inward normal direction could be still in Ω^+. Since the corresponding auxiliary region can be created for this auxiliary point, it will not be skipped. Only another two auxiliary points in Ω^+ are required above (x_i, y_j, z_{k+1}). For auxiliary points in Ω^-, an adaptive down shifting process shall be carried out to seek for four auxiliary regions in Ω^-.
4. For the interpolation in the auxiliary region, the order of two 1D interpolation steps should be decided. In this work, the decision depends on the location of the auxiliary point, e.g. (x_o, y_o, z_{k-1}) in Fig. 3. In particular, if $|x_o| \geq |y_o|$, the interpolation of M_1^- is carried out along x direction first. Otherwise, M_1^- is interpolated along y direction first.
5. Unlike the previous ray-casting MIB scheme for solving elliptic boundary value problems over irregular domains [46], the present ray-casting MIB scheme is free of an accuracy reduction issue. In [46], the MIB accuracy has been found to be seriously reduced in treating Dirichlet boundary conditions, if there is an auxiliary point that is very close to the boundary. Following the idea of Gibou et al. [24], all auxiliary nodes are shifted one position to restore the accuracy in [46]. Such an accuracy reduction is associated with the interpolation only, because the discretization of Neumann and

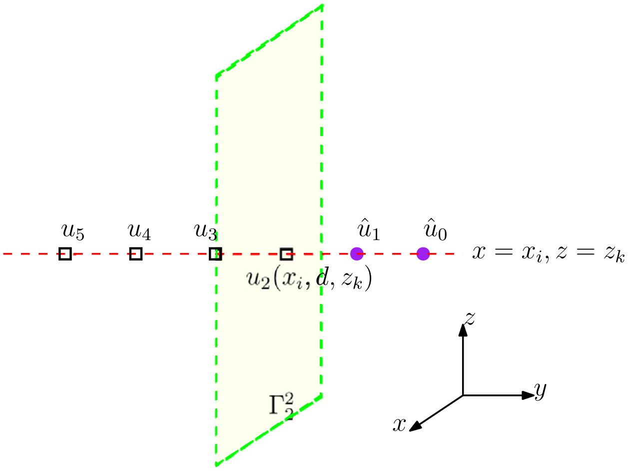


Fig. 4. The MIB boundary treatment along a y grid line with $x = x_i$ and $z = z_k$ at $y = d$. Two fictitious values $\hat{u}_0$ and $\hat{u}_1$ will be represented by using four function values u_2, u_3, u_4, and u_5, where u_2 is located on the boundary Γ_2^2.

> Robin conditions is free of this issue. Similarly, for the present interface problem, the discretization of jump conditions involves derivative approximations, not just interpolation. Thus, such kind of accuracy reduction has not been observed in the present ray-casting MIB scheme.

A comparison of the proposed ray-casting MIB scheme and the existing MIB scheme [60,54,55] is in order. The ray-casting MIB scheme keeps some nice features of the classical MIB scheme. First, both MIB schemes attain the fourth order of accuracy in treating curved interfaces. Second, both schemes only use zeroth and first order jump conditions for fictitious value generation, and no higher order jump conditions are needed. It is known that the higher order jump conditions include higher order derivatives and mix derivatives, whose numerical discretization is not trivial. Finally, both MIB interface treatments are independent of the governing PDE, so that they can be applied to solve various PDE interface problems, including elliptic, hyperbolic and parabolic types.

The ray-casting MIB scheme is different from the Cartesian MIB scheme [60,54,55] in several aspects. First, two jump conditions are implemented along the normal direction in the ray-casting scheme. In the original MIB scheme, two jump conditions together with further derived tangential jump conditions will be decomposed into x, y, and z directions for finite difference approximations. The present 1D implementation is much easier than the simultaneous 3D approximations in [54,55]. Second, because the normal direction makes it possible to reach more grid points inside/outside the interface, compared to the Cartesian directions, the ray-casting MIB scheme becomes more robust in handling complex geometry. Third, in the classical fourth order MIB scheme, four fictitious values along one Cartesian direction (two inside and two outside the interface) are usually generated together. This often runs into a corner problem, i.e., the Cartesian line will cut the interface twice within the range of four fictitious values [54,55]. With a sufficiently high grid resolution, the corner problem will not occur in the normal direction or in the ray-casting MIB scheme. In summary, the proposed ray-casting MIB scheme is simpler and more robust than the classical MIB scheme for PDE interface problems.

2.2.2. *Fictitious values near interface* Γ_2

For the interface Γ_2, the MIB boundary closure approach introduced in [57] will be simply applied to generate fictitious values. The MIB formulation for arbitrarily high order central difference is considered for various boundary conditions in [57]. Recall that for the present fourth order central difference, two layers of fictitious values are required outside Γ_2. Here, the fictitious value generation near to Γ_2^2 of Γ_2 is demonstrated in Fig. 1, while one can follow the same arguments to generate fictitious values for other sides of Γ_2. The MIB boundary scheme [57] can handle Dirichlet, Robin, and Neumann boundary conditions, and their mixed combination.

On the boundary $y = d$, we first consider a Robin boundary condition

$$\alpha u + \frac{\partial u}{\partial y} = g(x, d, z). \tag{15}$$

Take for instance the formulation of fictitious values right to Γ_2^2 along $x = x_i$ and $z = z_k$. Let us drop the function dependence on x and z and reduce the derivation into a 1D problem as shown in Fig. 4. Utilizing the fictitious values $\hat{u}_1$ and true function values u_2, u_3, u_4, u_5 along the y grid line, one can discretize the condition (15) as

$$\alpha u_2 + C_1^{(1)} \hat{u}_1 + \sum_{m=2}^{5} C_m^{(1)} u_m = g(x_i, z_k), \tag{16}$$

where $C_m^{(1)}$ are finite difference weights in approximating first order derivative at $y = d$ by using a stencil $\{\hat{u}_1, u_2, u_3, u_4, u_5\}$. Solving Eq. (16), one obtains the fictitious value $\hat{u}_1$. Note that this procedure works for a Neumann boundary condition by taking $\alpha = 0$ in Eq. (15).

For a Dirichlet boundary condition $u(x, d, z) = g(x, d, z)$ imposed at Γ_2^2, it will be combined with the equation $-\beta^+ \triangle u + \kappa\beta^+ u = f(x, y, z)$ in Ω^+. This gives rise to another condition:

$$u_{yy}(x, d, z) = -\frac{f(x, d, z)}{\beta^+} - g_{xx}(x, d, z) - g_{zz}(x, d, z) + \kappa g(x, d, z) \triangleq G(x, z). \tag{17}$$

By discretizing u_{yy} in the same manner, one can also obtain the fictitious value $\hat{u}_1$.

The fictitious value $\hat{u}_1$ calculated from Dirichlet, Neumann, or Robin boundary condition can be rewritten as:

$$\hat{u}_1 = \sum_{m=2}^{5} W_m u_m + W_0 G(x_i, z_k) + W_1 g(x_i, z_k). \tag{18}$$

To generate more fictitious values to support high order central differences, the MIB boundary scheme [57] will iteratively enforce the boundary conditions. In particular, by taking $\hat{u}_1$ as a known value, the discretization of boundary conditions by using a stencil $\{\hat{u}_0, \hat{u}_1, u_2, u_3, u_4, u_5\}$ determines another fictitious value $\hat{u}_0$. Referring to [57], $\hat{u}_0$ can also be represented in a form like Eq. (18). Other fictitious values outside Γ_2 from x direction or z direction can be formulated similarly.

2.3. Fourth order central difference discretization

In the present work, an augmented MIB (AMIB) method will be constructed for solving 3D elliptic interface problems, which is generalized from the recently developed AMIB scheme for various PDE interface and boundary value problems [16,18,17,46,35]. Different from the classical MIB methods [60,57], the fictitious values calculated above will not be directly used to modify the Laplacian approximation at irregular points. Instead, in the augmented formulation, the standard fourth order central difference will be used at irregular points near Γ_1 and Γ_2, subject to certain corrections by means of Cartesian derivative jumps $[u^{(k)}]$ at various intersection points. Then, the derivative jumps $[u^{(k)}]$ will be reconstructed with aid of fictitious values near Γ_1 and Γ_2. More details are provided in the following subsections.

2.3.1. Correcting fourth order central difference

Taking advantage that the Laplacian approximation can be carried out in a tensor product manner, one can discuss the ideas of the proposed AMIB method by taking one direction into account first. For simplicity, we may ignore the function dependence on x and z at the moment, by denoting $u = u(y)$. The following theorem provides the fourth order corrected finite difference for a function $u(y)$ across an interface.

Theorem 1 *(Corrected fourth order finite differences). Let $y_j \le \alpha \le y_{j+1}$, $h^- = y_j - \alpha$, and $h^+ = y_{j+1} - \alpha$. Suppose $u \in C^6[y_j - 2h, \alpha) \bigcap C^6(\alpha, y_{j+1} + 2h]$, with derivative extending continuously up to the interface $y = \alpha$. Then the following approximations hold to $O(h^4)$ when $K = 5$:*

$$u_{yy}(y_{j-1}) \approx \frac{1}{h^2}[-\frac{1}{12}u(y_{j-3}) + \frac{4}{3}u(y_{j-2}) - \frac{5}{2}u(y_{j-1}) + \frac{4}{3}u(y_j) - \frac{1}{12}u(y_{j+1})] + \frac{1}{12h^2}\sum_{k=0}^{K}\frac{(h^+)^k}{k!}[u^{(k)}],$$

$$\begin{aligned} u_{yy}(y_j) \approx{}& \frac{1}{h^2}[-\frac{1}{12}u(y_{j-2}) + \frac{4}{3}u(y_{j-1}) - \frac{5}{2}u(y_j) + \frac{4}{3}u(y_{j+1}) - \frac{1}{12}u(y_{j+2})] \\ & - \frac{4}{3h^2}\sum_{k=0}^{K}\frac{(h^+)^k}{k!}[u^{(k)}] + \frac{1}{12h^2}\sum_{k=0}^{K}\frac{(h+h^+)^k}{k!}[u^{(k)}], \end{aligned}$$

$$\begin{aligned} u_{yy}(y_{j+1}) \approx{}& \frac{1}{h^2}[-\frac{1}{12}u(y_{j-1}) + \frac{4}{3}u(y_j) - \frac{5}{2}u(y_{j+1}) + \frac{4}{3}u(y_{j+2}) - \frac{1}{12}u(y_{j+3})] \\ & + \frac{4}{3h^2}\sum_{k=0}^{K}\frac{(h^-)^k}{k!}[u^{(k)}] - \frac{1}{12h^2}\sum_{k=0}^{K}\frac{(h^- - h)^k}{k!}[u^{(k)}], \end{aligned}$$

$$u_{yy}(y_{j+2}) \approx \frac{1}{h^2}[-\frac{1}{12}u(y_j) + \frac{4}{3}u(y_{j+1}) - \frac{5}{2}u(y_{j+2}) + \frac{4}{3}u(y_{j+3}) - \frac{1}{12}u(y_{j+4})] - \frac{1}{12h^2}\sum_{k=0}^{K}\frac{(h^-)^k}{k!}[u^{(k)}],$$

where $[u^{(k)}] = [u^{(k)}]_\alpha = \lim_{y\to\alpha^+} u^{(k)}(y) - \lim_{y\to\alpha^-} u^{(k)}(y)$.

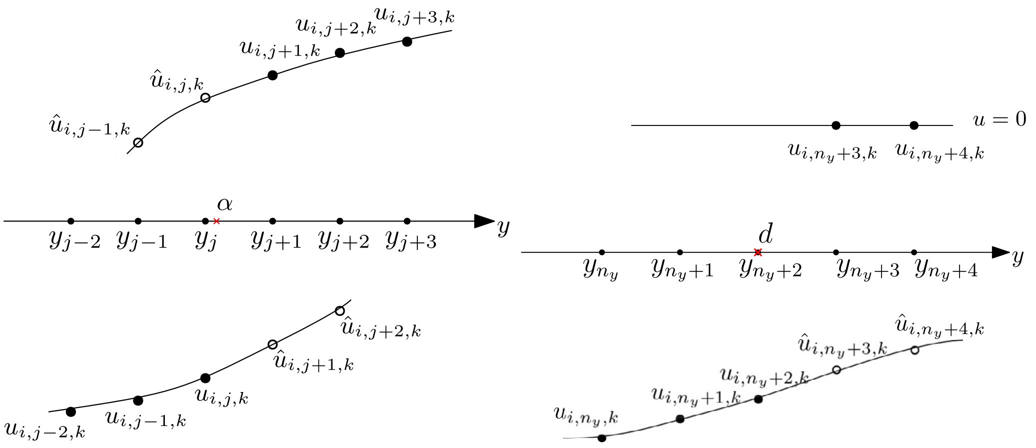


Fig. 5. An illustration of the fourth order numerical approximation to Cartesian derivative jumps at Γ_1 and Γ_2. In both charts, crosses stand for the interface point, while filled circles represent real function values and empty circles denote fictitious values. The left chart is a general case for jump values at $y=\alpha$ for Γ_1, while the right one is at boundary point $y=d$ for the side Γ_2^2 of Γ_2.

Bearing the above y-oriented corrected difference in mind, we can easily generalize the idea into 3D by combining corrected differences from different directions to approximate the Laplacian operator dimension by dimension. In all corrected finite differences, the standard central difference is preserved and the correction terms contain jump quantities at interface points. When away from interfaces, the correction terms vanish for regular points, so that the corrected finite difference reduces to the standard central difference. Thus, the corrected finite difference can be applied to all grid nodes. In order to achieve fourth order accuracy, K needs to be at least 5 for a fourth order truncation error $O(h^4)$. However, in practice, one can retain fourth order accuracy if we only take $K=4$ with a third order local truncation error $O(h^3)$ at irregular points for approximation to Laplacian, because global fourth order accuracy can still be realized.

The corrected fourth order finite differences will be applied to handle the interior interface Γ_1 and the exterior interface Γ_2. To be more specific, for Γ_1, a general case is usually taken, in which one comes across an interface point α positioned between two of its adjacent grid points, y_j and y_{j+1} with $h^-=y_j-\alpha$, and $h^+=y_{j+1}-\alpha$. For Γ_2, Theorem 1 is also applicable when each side of Γ_2 aligns with a grid line and the interface point α coincides with a grid point. This can be realized by defining $h^-=-h$ and $h^+=0$ for the lower side Γ_2^1, the left side Γ_2^4, and the front side Γ_2^5, and $h^-=0$ and $h^+=h$ for the upper side Γ_2^3, the right side Γ_2^2, and the back side Γ_2^6.

2.3.2. *Cartesian jump values reconstruction*

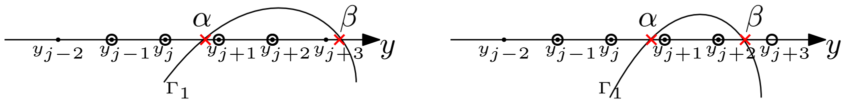
The fourth order central differences are corrected at various interface points, which are intersection points between a Cartesian grid line and Γ_1 or Γ_2. At such interface point, the Cartesian derivative jumps usually cannot be obtained from the given PDE problem. We next consider how to numerically approximate Cartesian derivative jumps at α, i.e., $[u^{(m)}]_\alpha$ $(m=0,1,\cdots,4)$ with the aid of MIB fictitious values near Γ_1 and Γ_2 calculated above.

Instead of a numerical approximation, the zeroth derivative jump is the function jump across the interface, which can be explicitly obtained by the analytical jump conditions Eq. (3) or (10). In order to reconstruct other jump values for m up to 4 in the corrected difference, two Lagrange polynomials of degree four can be built to give limiting derivatives, resulting in the approximated jumps up to fourth order derivative. Take a jump reconstruction along y-direction for an illustration here. Two layers of fictitious values are available on each side of the interface Γ_1 and outside Γ_2, see Fig. 5. We can combine three real function values together with two fictitious values on the other side of the interface to obtain the Lagrange polynomial for one-side subdomain.

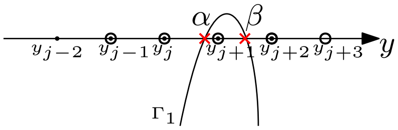
By taking derivatives on these one-sided polynomials, the jump values at an intersection point α with regard to y-directions will take a general form in 3D as below

$$\begin{aligned}[\frac{\partial^m u}{\partial y^m}]|_{y=\alpha} \approx\ & (w^m_{i,j-1,k}\hat{u}_{i,j-1,k}+w^m_{i,j,k}\hat{u}_{i,j,k}+\sum_{l=1}^{3} w^m_{i,j+l,k}u_{i,j+l,k}) \\ & -(\sum_{l=1}^{3} w^m_{i,j-3+l,k}u_{i,j-3+l,k}+w^m_{i,j+1,k}\hat{u}_{i,j+1,k}+w^m_{i,j+2,k}\hat{u}_{i,j+2,k}),\end{aligned} \tag{19}$$

where $w^m_{p,q,r}$ denotes the finite difference weights for each function value at (x_p,y_q,z_r) after taking m^{th} derivative of the Lagrange polynomial at $y=\alpha$. For Γ_2, only approximation from the inside of Γ_2 is involved at the intersection point $\alpha=d$. In other words, the limit from outside of Γ_2 is trivially zero. To be more clear, the first polynomial at the intersection point



(a) Type 1. (b) Type 2.



(c) Type 3.

Fig. 6. All possible corner scenarios in correcting the fourth order central difference. Empty circles denote the fictitious values at the grid points, while black dots indicate real function values.

d on the right hand side of Eq. (19) is zero because of the zero-padding function values beyond Γ_2, see the right chart of Fig. 5 for an illustration.

2.3.3. Corner treatments

As discussed above, the ray-casting MIB scheme is naturally free of corner issue in generating fictitious values for the interface Γ_1. Since Γ_2 is flat, no corner point could occur along Cartesian directions. However, the correction of fourth order central difference along Cartesian directions near Γ_1 may involve corner issues, which deserve further study. In order to reconstruct the derivative jumps in a safe manner, the local interface topology has to be monitored exclusively for Γ_1, as demonstrated in [18]. Because of the tensor product nature of the AMIB scheme, one just needs to focus on the corner treatment in 1D. In practice, a Cartesian grid line may intersect with the interface twice within a relatively short distance, yielding two adjacent intersection points α and β, see Fig. 6. If both intersection points are within the grid range considered in Theorem 1, a new correction has to be developed for the fourth order finite difference.

Recently, a ray-casting AMIB scheme has been developed in [35] for solving initial boundary value problems of the diffusion equation over irregular domains. In [35], all possible corner cases occurred in correcting fourth order central difference for a boundary value problem (BVP) have been examined, and the corresponding corrections have been successfully constructed. Interested readers are directed to Subsection 3.2.2 of Ref. [35] for a detailed description of different treatments for BVPs. Essentially, all corner cases occurred in interface problems can be corrected as in [35], except that the reconstruction of Cartesian derivative jumps needs function values from both outside and inside of Γ_1, while for BVPs [35], the solution outside an irregular domain is trivially zero.

Recall that the reconstruction of derivative jumps in finite difference correction usually needs three function values on one side of the interface, and two fictitious values on the other side, as shown in Fig. 5. Thus, three scenarios should be considered for possible corner issues, depending on the distance between two intersection points α and β, see Fig. 6.

The first case is that at least three grid points are available between α and β, such as Fig. 6 (a). One names this corner situation as Type 1. In this case, the finite difference correction and derivative jump reconstruction at α and β are actually decoupled, and Eq. (19) can be safely used without modification. In the case of only two grid point between α and β, the finite difference correction used in [35] can be applied, while Eq. (19) needs to be modified. We name this corner situation as Type 2. For instance, to approximate jump values at α shown in Fig. 6 (b), we may simply use fictitious value $\hat{u}_{i,j+3,k}$ instead of real function value $u_{i,j+3,k}$ in Eq. (19). Thus, the Cartesian derivative jumps in Type 2 are approximated by

$$[\frac{\partial^m u}{\partial y^m}]|_{y=\alpha} \approx (w^m_{i,j-1,k}\hat{u}_{i,j-1,k} + w^m_{i,j,k}\hat{u}_{i,j,k} + \sum_{l=1}^{2} w^m_{i,j+l,k}u_{i,j+l,k} + w^m_{i,j+3,k}\hat{u}_{i,j+3,k})$$
$$- (\sum_{l=1}^{3} w^m_{i,j-3+l,k}u_{i,j-3+l,k} + w^m_{i,j+1,k}\hat{u}_{i,j+1,k} + w^m_{i,j+2,k}\hat{u}_{i,j+2,k}). \quad (20)$$

Furthermore, as the case becomes more extreme, we refer to Fig. 6 (c), in which only one grid point is positioned in between α and β. Such a case is called corner Type 3. The fourth order central difference can be corrected as in [35]. Here, for the purpose of constructing derivative jumps at α, a simple treatment is applied by replacing real function values $u_{i,j+2,k}, u_{i,j+3,k}$ with fictitious values $\hat{u}_{i,j+2,k}, \hat{u}_{i,j+3,k}$, and substituting the function value $u_{i,j+2,k}$ for $\hat{u}_{i,j+2,k}$ in Eq. (19). In such a case, Eq. (19) can be rewritten as

$$[\frac{\partial^m u}{\partial y^m}]|_{y=\alpha} \approx (w^m_{i,j-1,k}\hat{u}_{i,j-1,k} + w^m_{i,j,k}\hat{u}_{i,j,k} + +w^m_{i,j+1,k}u_{i,j+1,k} + \sum_{l=2}^{3} w^m_{i,j+l,k}\hat{u}_{i,j+l,k})$$

$$-(\sum_{l=1}^{3} w^m_{i,j-3+l,k} u_{i,j-3+l,k} + w^m_{i,j+1,k}\hat{u}_{i,j+1,k} + w^m_{i,j+2,k} u_{i,j+2,k}). \tag{21}$$

The treatment at β can be similarly formulated. With these corner treatments, the corrections to the fourth order central difference at various corner types can be accurately constructed.

2.4. Formulation of augmented system and FFT inversion

In the present study, an augmented linear system will be formulated for 3D elliptic interface problems, following the previous AMIB methods [16,18,17,46,35]. In particular, the Cartesian derivative jumps at various interface points on Γ_1 and Γ_2 will be introduced as auxiliary variables. Consequently, the 3D discrete Laplacian can be efficiently inverted by the fast Fourier transform (FFT) algorithm, while the auxiliary variables can be solved iteratively in the Schur complement procedure.

2.4.1. Augmented system

In order to construct the augmented system, we first consider the derivative jumps reconstruction Eq. (19) or Eq. (20) or Eq. (21) and rewrite it into an equivalent matrix form. Let $N_1 = (n_x+3)\times(n_y+3)\times(n_z+3)$ be the total number of interior grids of the whole cuboid domain D and N_2 be the total number of the interface points in the x-, y- and z-directions.

In the preceding subsection, derivative jumps are reconstructed in terms of fictitious values at each intersection point between a grid line and Γ_1 or Γ_2. By introducing those derivative jumps as auxiliary variables, plugging the fictitious value representation Eq. (14) or Eq. (18) into Eq. (19) or Eq. (20) or Eq. (21) yields a generic linear equation

$$\sum_{(x_I,y_J,z_K)\in\mathbb{S}_{i,j,k}} C_{I,J,K} u_{I,J,K} + [\frac{\partial^m u}{\partial y^m}] = C_o G + C_0\phi + C_1\psi, \tag{22}$$

where $C_{I,J,K}$ is the corresponding weights of function value $u_{I,J,K}$ in approximation to jump quantity $[\frac{\partial^m u}{\partial y^m}]$, and G, ϕ, and ψ are the known interface data. As a matter of fact, one can obtain formulas similar to Eq. (22) for all interface points in x-, y- and z-directions.

Furthermore, one denotes the 1D column vector Q of dimension $5N_2\times 1$ as a vector of the introduced auxiliary variables $[\frac{\partial^m u}{\partial x^m}]_i, [\frac{\partial^m u}{\partial y^m}]_j$, and $[\frac{\partial^m u}{\partial z^m}]_k, m = 0, 1, \cdots, 4$ for $i = 1, 2, \cdots,$ $j = 1, 2, \cdots,$ and $k = 1, 2, \cdots,$ at total N_2 intersection points between grid lines and Γ_1 or Γ_2. Unknown function values at N_1 interior grids within the domain D are formed in a $1D$ column vector U of dimension $N_1 \times 1$. After generalizing Eq. (22) from one interface point to all interface points, the matrix form of Eq. (22) is then written as

$$CU + IQ = \Phi, \tag{23}$$

where C is a sparse matrix of dimension $5N_2 \times N_1$, I is the identity matrix of dimension $5N_2 \times 5N_2$, and Φ is a column vector of dimension $5N_2 \times 1$ composed of the known interface quantities.

For the 3D elliptic problem (9), denote $U_{i,j,k}$ as the discrete solution at (x_i, y_j, z_k). Making use of the corrected differences at all interior mesh grids and taking correction terms as auxiliary variables, the PDE (9) can be discretized as

$$L_h U_{i,j,k} + C_{i,j,k} = f_{i,j,k}, \quad 1 \le i \le n_x+3, \quad 1 \le j \le n_y+3, \quad 1 \le k \le n_z+3 \tag{24}$$

where $C_{i,j,k}$ is the correction term, and $L_h U_{i,j,k}$ is the standard fourth order central difference approximation to the Laplacian operator with a degree of freedom N_1. Consequently, the resulting equation in the matrix form is given by

$$(\bar{A} - \kappa I)U + BQ = F, \tag{25}$$

where B is a sparse matrix of dimension N_1 by $5N_2$ consisting of coefficients from correction terms, and F is a vector of dimension $N_1 \times 1$ with entries being $f_{i,j,k}$. The symmetric and diagonally dominant matrix $(\bar{A} - \kappa I)$ consist of coefficients from discretizing the 3D differential operator $\Delta - \kappa$ obtained via the fourth order central difference.

From now on, let us denote $\bar{A} - \kappa I$ as A. Coupling (25) and (23) yields an augmented system,

$$KW = R, \tag{26}$$

where

$$K = \begin{pmatrix} A & B \\ C & I \end{pmatrix}, W = \begin{pmatrix} U \\ Q \end{pmatrix}, \quad \text{and} \quad R = \begin{pmatrix} F \\ \Phi \end{pmatrix}.$$

2.4.2. FFT inversion

The inversion of matrix A can be carried out by 3D FFT fast Poisson solver. As suggested in [18], an anti-symmetric property at ∂D is trivially satisfied in the proposed AMIB method. Moreover, taking advantage of the corrected central differences and auxiliary variables, the solution discontinuities will not be sensed in the FFT computation at both Γ_1 and Γ_2. By the tensor product approach, the generalization to 3D FFT inversion of A can be carried out based on 1D FFT inversion.

With parameters P, Q, T being n_x+3, n_y+3, n_z+3 respectively, consider a 3D linear system

$$Au = f, \tag{27}$$

where A is a constant matrix resulting from fourth order central difference discretization. The FFT inversion of (27) is obtained via the 3D fast Sine transform (FST) as described below:

1. Relabel f to be an equivalent 3D array $[f_{p,q,t}]$ for $p = 1, \cdots, P, q = 1, \cdots, Q, t = 1, \cdots, T$, and compute the Sine transform for $f_{p,q,t}$ via 3D inverse fast Sine transform (IFST)
   $\hat{f}_{l,m,n} = \frac{8}{(P+1)(Q+1)(T+1)} \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{t=1}^{T} f_{p,q,t} \sin(\frac{lp\pi}{P+1}) \sin(\frac{mq\pi}{Q+1}) \sin(\frac{nt\pi}{T+1})$,
   for $l = 1, \cdots, P,\ m = 1, \cdots, Q,\ n = 1, \cdots, T$.
2. Compute
   $\lambda_l^x = -\frac{1}{3h^2}[\cos(\frac{l\pi}{P+1}) - 1][\cos(\frac{l\pi}{P+1}) - 7]$,
   $\lambda_m^y = -\frac{1}{3h^2}[\cos(\frac{m\pi}{Q+1}) - 1][\cos(\frac{m\pi}{Q+1}) - 7]$,
   $\lambda_n^z = -\frac{1}{3h^2}[\cos(\frac{n\pi}{T+1}) - 1][\cos(\frac{n\pi}{T+1}) - 7]$,
   $\hat{u}_{l,m,n} = \frac{\hat{f}_{l,m,n}}{\lambda_l^x + \lambda_m^y + \lambda_n^z - \kappa}$, for $l = 1, 2, \cdots, P,\ m = 1, \cdots, Q,\ n = 1, \cdots, T$.
3. Compute $u_{p,q,t}$ via 3D fast Sine transform (FST):
   $u_{p,q,t} = \sum_{l=1}^{P} \sum_{m=1}^{Q} \sum_{n=1}^{T} \hat{u}_{l,m,n} \sin(\frac{lp\pi}{P+1}) \sin(\frac{lq\pi}{Q+1}) \sin(\frac{lt\pi}{T+1})$,
   for $p = 1, \cdots, P,\ q = 1, \cdots, Q,\ t = 1, \cdots, T$. The resulting 3D array $[u_{p,q,t}]$ is then reshaped to a $1D$ vector for the solution of the linear system.

2.4.3. Schur complement

A Schur complement method is used to eliminate U from the augmented system (26), forming a linear system with dimension $5N_2 \times 5N_2$ for Q

$$(I - CA^{-1}B)Q = \Phi - A^{-1}F. \tag{28}$$

The detailed explanation of implementation on Schur complement system (28) with a biconjugate gradient iteration is presented as below:

1. The calculation of the right side $\hat{F} = \Phi - CA^{-1}F$ involves performing FFT on F for $A^{-1}F$ and additional matrix-vector multiplication and vector-vector subtraction.
2. The left hand multiplication of matrix by vector $(I - CA^{-1}B)Q$ is achieved by $IQ - CA^{-1}BQ$. To be more clear, BQ is obtained by direct matrix-vector multiplication. A FFT is again utilized on $A^{-1}BQ$ and it is followed with additional direct matrix-vector manipulation and vector-vector subtraction for $IQ - CA^{-1}(BQ)$. The transpose of $(I - CA^{-1}B)^T Q$ is done on $IQ - B^T A^{-1} C^T Q$ by following the same strategy.
3. An initial guess $Q = (0, 0, \cdots, 0)^T$ starts the biconjugate gradient iteration, and Q will be updated to start a new iteration until either the maximal iteration number 5000 or error tolerance equaling to 10^{-12} is reached.

After the auxiliary vector Q is solved from (28), the numerical solution U could be solved by one more FFT inversion

$$AU = F - BQ.$$

The iterative solution of Eq. (28) is known to be quite efficient in the existing AMIB methods [16,18,17,46,35], due to two reasons. First, dimension $5N_2$ of Q is one dimensionally smaller than that of U, i.e., $N_1 = (n_x+3) \times (n_y+3) \times (n_z+3)$ or the total degree of freedom in 3D. Second, iterative number in the biconjugate gradient solution of Eq. (28) usually just weakly depends on N_2 in the AMIB method. Consequently, the major computational cost is consumed by the FFT inversion of 3D discrete Laplacian, so that the total complexity of the AMIB method for solving 3D elliptic interface problems is $O(N_1 \log N_1)$.

3. Numerical experiments

In this section, we will examine the accuracy and efficiency of the proposed ray-casting AMIB scheme solving two or three dimensional elliptic equations with piecewise constant β. Moreover, the proposed ray-casting MIB scheme will also be examined, in which when the fourth order central difference stencil is crossing the interface, the fictitious values will be simply supplied. In solving 2D interface problems, the Cartesian MIB and the corresponding AMIB introduced in [18] will also be employed for a comparison. While all four methods are fourth order of accurate, their robustness and efficiency could be different.

For simplicity, the domain D will be assumed to be a cubic domain with a uniform number of subintervals in each direction, i.e., $n = n_x = n_y = n_z$. The accuracy and convergence of the numerical solution in 3D problems are tested by measuring errors in the maximum norm and L_2 norm

$$L_\infty = \max_{(x_i,y_j,z_k)\in\Omega^-\cup\Omega^+} |u(x_i,y_j,z_k) - u_h(x_i,y_j,z_k)|,$$

$$L_2 = \sqrt{\frac{1}{(n+1)^3}\sum_{(x_i,y_j,z_k)\in\Omega^-\cup\Omega^+} |u(x_i,y_j,z_k) - u_h(x_i,y_j,z_k)|^2},$$

where $u(x_i,y_j,z_k)$ and $u_h(x_i,y_j,z_k)$ are respectively analytical and numerical solution. The error norms in 2D can be similarly defined.

The accuracy in approximating solution gradients or fluxes will be examined for some examples. The fourth order central difference is employed for $\frac{\partial u_h}{\partial x}$, $\frac{\partial u_h}{\partial y}$, and $\frac{\partial u_h}{\partial z}$ in the computation of the discrete gradient $\nabla u_h(x_i,y_j,z_k)$. For instance,

$$\frac{\partial u_h}{\partial x}(x_i,y_j,z_k) \triangleq \frac{1}{12h}u_h(x_{i-2},y_j,z_k) - \frac{8}{12h}u_h(x_{i-1},y_j,z_k) + \frac{8}{12h}u_h(x_{i+1},y_j,z_k) - \frac{1}{12h}u_h(x_{i+2},y_j,z_k), \tag{29}$$

and $\frac{\partial u_h}{\partial y}$ and $\frac{\partial u_h}{\partial z}$ can be computed similarly. When the fourth order central difference stencil crosses the interface, the function values are replaced with the corresponding fictitious values in Eq. (29). Then, the errors of gradient approximation can be considered in the maximum norm and L_2 norm defined as

$$\begin{aligned}L_\infty = \max_{(x_i,y_j,z_k)\in\Omega^-\cup\Omega^+} \max\{&|\frac{\partial u}{\partial x}(x_i,y_j,z_k) - \frac{\partial u_h}{\partial x}(x_i,y_j,z_k)|, |\frac{\partial u}{\partial y}(x_i,y_j,z_k) - \frac{\partial u_h}{\partial y}(x_i,y_j,z_k)|,\\ &|\frac{\partial u}{\partial y}(x_i,y_j,z_k) - \frac{\partial u_h}{\partial y}(x_i,y_j,z_k)|\},\end{aligned}$$

$$L_2 = \sqrt{\frac{1}{(n+1)^3}\sum_{(x_i,y_j,z_k)\in\Omega^-\cup\Omega^+} ||\nabla u(x_i,y_j,z_k) - \nabla u_h(x_i,y_j,z_k)||_2^2},$$

where $\nabla u(x_i,y_j,z_k)$ and $\nabla u_h(x_i,y_j,z_k)$ are respectively analytical and numerical gradient.

The convergence rate will be measured by the formula

$$\text{order} = \frac{\log(||E_1||/||E_2||)}{\log(h_1/h_2)},$$

where $||E_i||$ is the error based a mesh spacing h_i for $i = 1, 2$, using the above defined norms on $(n+1)\times(n+1)\times(n+1)$ mesh for the interested domain $\Omega^-\cup\Omega^+$. The solution calculation is facilitated by a FFT subroutine from Numerical Recipes [45] with 2^k summation. Due to the restriction of partition number equaling to 2^k in the subroutine, non-bisectional mesh refinement is usually conducted in the error analysis.

All the experiments are carried out by using a single core on a Dell PowerEdge R920 in The University of Alabama High-Performance Computer (UAHPC) (https://oit.ua.edu/services/research/) with Intel® Xeon® CPU E7-8891 v2 operating at 3.20 GHz clock speed.

3.1. Numerical examples in 2D

We first consider two examples in 2D so that the proposed ray-casting AMIB scheme can be compared with the existing MIB schemes.

Example 1. This example focuses on a 2D Poisson's equation

$$(\beta u_x)_x + (\beta u_y)_y = q(x,y), \tag{30}$$

Table 1
Example 1 – $\beta^+ = 10^{-4}, \beta^- = 1$; circular interface.

$[n, n]$	Ray-casting AMIB					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[60, 60]	1.182E-2	–	5.537E-3	–	66	0.113
[124, 124]	4.962E-4	4.5744	2.344E-4	4.5622	39	0.267
[252, 252]	3.284E-5	3.9176	1.557E-5	3.9117	56	1.338
[508, 508]	3.159E-6	3.3770	1.501E-6	3.3748	123	10.650
$[n, n]$	Ray-casting MIB					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[60, 60]	1.753E-2	–	8.212E-3	–	579	0.176
[124, 124]	4.353E-4	5.3320	2.056E-4	5.3200	1335	1.307
[252, 252]	2.610E-5	4.0595	1.238E-5	4.0537	3254	11.979
[508, 508]	2.133E-6	3.6130	1.012E-6	3.6121	15140	213.963
$[n, n]$	Cartesian AMIB [18]					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[60, 60]	–	–	–	–	–	–
[124, 124]	9.327E-4	–	4.406E-4	–	41	0.276
[252, 252]	5.898E-5	3.9831	2.797E-5	3.9772	56	1.396
[508, 508]	3.321E-6	4.1505	1.578E-6	4.1477	97	8.395
$[n, n]$	Cartesian MIB [18]					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[60, 60]	2.087E-2	–	9.776E-3	–	898	0.329
[124, 124]	8.824E-4	4.5639	4.168E-4	4.5518	1442	1.867
[252, 252]	5.713E-5	3.9491	2.710E-5	3.9432	4002	19.679
[508, 508]	–	–	–	–	–	–

over in a square $[-\frac{\pi}{3.5}, \frac{\pi}{3.5}] \times [-\frac{\pi}{3.5}, \frac{\pi}{3.5}]$ with a circular interface Γ defined by $r^2 := x^2 + y^2 = 0.3^2$. The exact solution to this problem is prescribed as

$$u(x, y) = \begin{cases} \cos(kx)e^y, & r \le 0.3, \\ e^x(x^2 \sin(y) + y^2), & \text{otherwise}, \end{cases} \tag{31}$$

with the diffusion coefficient

$$\beta = \begin{cases} 1, & r \le 0.3, \\ 10^{-4}, & \text{otherwise}, \end{cases}$$

and the parameter k is chosen to be 3. Here we call β as β^+ when it is outside the interface Γ, and β^- when it is inside the interface Γ. The same principle is followed for the below notations. The source term $q(x, y)$ is related to the above designated solution,

$$q(x, y) = \begin{cases} (1-k^2)\beta^- \cos(kx)e^y, & r \le 0.3, \\ \beta^+ e^x(2 + y^2 + (4x+2)\sin y), & \text{otherwise}. \end{cases}$$

On the boundary of the domain $[-\frac{\pi}{3.5}, \frac{\pi}{3.5}] \times [-\frac{\pi}{3.5}, \frac{\pi}{3.5}]$, a Dirichlet boundary condition is assumed with the boundary data derived by the analytical solution. For treating such a Dirichlet boundary condition, the introduction of an extended domain Ω^e is necessary in the ray-casting AMIB and Cartesian AMIB schemes. Consequently, the mesh size is taken to be $(n+5) \times (n+5)$ for the whole domain $D = \Omega^- \cap \Omega^+ \cap \Omega^e$ in two AMIB schemes. Without the extended domain Ω^e, the mesh size of the ray-casting MIB and Cartesian MIB schemes will be $(n+1) \times (n+1)$ so that the grid spacing h is the same.

Taking $\beta^+ = 10^{-4}$ and $\beta^- = 1$, the numerical results of the ray-casting AMIB, ray-casting MIB, Cartesian AMIB, and Cartesian MIB methods are presented in Table 1. It can be seen that all four methods achieve fourth order of convergence, and two ray-casting methods are slightly more accurate. Two failure cases are encountered in this example. First, the Cartesian AMIB scheme fails on a coarse mesh with $[n, n] = [60, 60]$. Note that the underlying Cartesian MIB scheme is able to handle the interface Γ based on the same mesh resolution. Thus, the failure is due to the lack of corner treatments in the augmented formulation for correcting the fourth order central differences. By using the corner treatments discussed in Section 2.3.3, the ray-casting AMIB scheme is free of corner issues, and yields a satisfactory result for the coarse mesh with $[n, n] = [60, 60]$. Second, the iterative solution of the Cartesian MIB scheme fails to converge for a dense mesh with $[n, n] = [508, 508]$, while the Cartesian AMIB scheme works for this dense mesh. Since both schemes use the same set of fictitious values, this means that such fictitious values are actually correct, and the failure is simply due to iterative solution process for a large matrix with irregular structure. It is noted that if a more robust algebraic solver is used, the Cartesian MIB scheme may still produce a sound numerical solution at $[508, 508]$. The present study indicates that with an augmented formulation and the FFT inversion, the AMIB scheme becomes more robust than the MIB scheme. This is one of motivations for developing the AMIB schemes [16–18,46,35].

The efficiency of the four methods is also reported in Table 1 by means of the iteration number and CPU time. It can be seen that two AMIB schemes are much more efficient than the corresponding MIB schemes, and the Cartesian AMIB method is slightly faster than the ray-casting AMIB method in this example. Focusing on the ray-casting schemes, when the mesh size is doubled for three times, the iteration number of the AMIB is doubled, while the iteration number of the MIB becomes 26 times larger. Consequently, on a mesh 513 by 513, the ray-casting AMIB scheme is about 21 times faster than the ray-casting MIB.

We next examine the dependence of accuracy and efficiency on the PDE coefficient β. Following [4,18], the impact of highly contrasted coefficients is investigated, by changing β^+ from 10^{-4} to 10^9 with $\beta^- = 1$. A dense mesh with $[n, n] = [508, 508]$ is used, which keeps the spacing h to be the same in the four methods. The L_∞ error, L_2 error, iteration number, and CPU time of the four methods are plotted against β^+ in Fig. 7. For the Cartesian MIB, extrapolations in jump condition discretization can be carried out either from positive or negative side of Γ. An upwinding procedure was suggested in [18], in which the extrapolation direction depends on whether $\beta^- \leq \beta^+$ or $\beta^- < \beta^+$, so that the accuracy and condition number of the Cartesian MIB and AMIB schemes could be improved. We note that in the proposed ray-casting MIB method, no extrapolation is needed, and the same procedure is utilized no matter β^- is larger or smaller than β^+.

For two error subplots in Fig. 7, it can be seen that errors increase significantly as β^+ goes to zero, while they do not grow as β^+ increases. It is well known that as the PDE coefficient β^+ approaches to zero, the elliptic problem becomes a singular perturbation one, which is very challenging to solve numerically. This is why all four methods produce much large errors as β^+ becomes smaller, and the Cartesian MIB scheme fails when $\beta^+ = 10^{-4}$. For $\beta^+ > 1$, we can see that two Cartesian MIB schemes yield the same errors, except at 10^3, while the errors of two ray-casting MIB schemes are different for different β^+ values. Moreover, ray-casting MIB schemes are more accurate than Cartesian MIB schemes, even though an adaptive extrapolation is used in the two Cartesian MIB schemes. In terms of efficiency, in all cases, two AMIB methods need much smaller iteration numbers, and consume much less computational time. Also, the Cartesian AMIB scheme is slightly more efficient than the ray-casting AMIB scheme for extreme β^+ values.

Example 2. The second example concerns the Poisson equation (30) on a square domain $[-\frac{\pi}{3.5}, \frac{\pi}{3.5}] \times [-\frac{\pi}{3.5}, \frac{\pi}{3.5}]$, and the interface Γ is given as

$$r = 0.5 + b \sin(5\theta),$$

with $b = 0.12$. This results in a five-leaf interface as shown in Fig. 8. In this example, the exact solution is defined by

$$u(x, y) = \begin{cases} e^{-x^2 - 0.5y^2}, & \text{inside } \Gamma, \\ \sin(kx)\sin(ky), & \text{otherwise}, \end{cases} \tag{32}$$

with $k = 3$ and the diffusion coefficient is given as

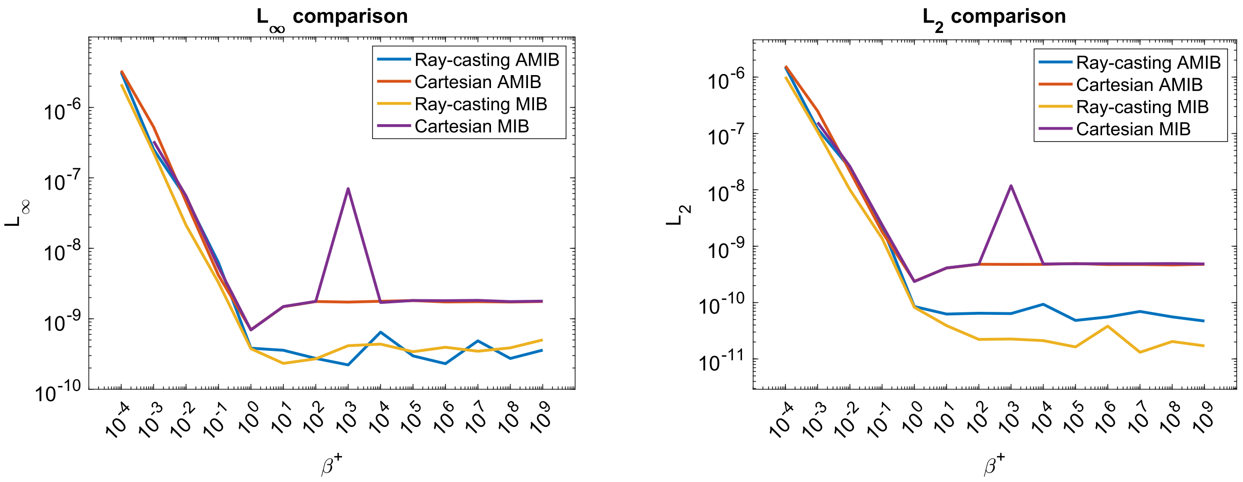
$$\beta = \begin{cases} 10, & \text{inside } \Gamma, \\ 1, & \text{otherwise}. \end{cases}$$

The source term $q(x, y)$ is related to the above designated solution,

$$q(x, y) = \begin{cases} (4x^2 + y^2 - 3)\beta^- e^{-x^2 - 0.5y^2}, & \text{inside } \Gamma, \\ -2k^2 \beta^+ \sin(kx)\sin(ky), & \text{otherwise}. \end{cases}$$

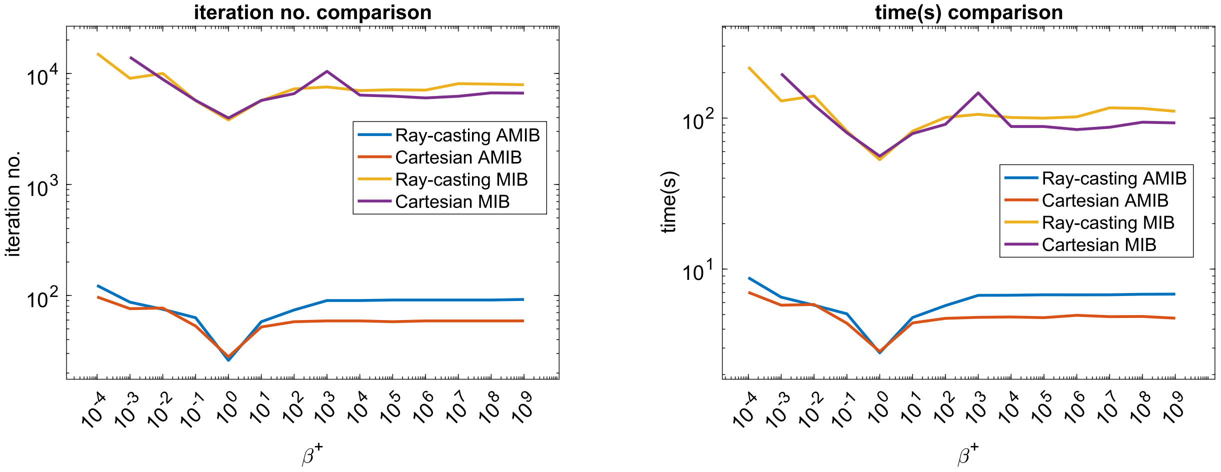
Two boundary conditions are studied for the present example, i.e., a Dirichlet condition and a Robin condition, and the boundary data is derived based on the analytical solution. For simplicity, only the ray-casting and Cartesian AMIB schemes will be tested. The numerical results are listed in Table 2 and Table 3. In terms of accuracy, the results of two boundary conditions are quite similar. In particular, the Cartesian AMIB scheme is slightly more accurate than the ray-casting AMIB. However, when $[n, n] = [252, 252]$, the Cartesian AMIB scheme is experiencing some corner issue, so that it fails in both cases with different boundary conditions. Here the corner issue is due to that one Cartesian grid line cuts the interface within a length of $2h$. Unfortunately, for a complicated interface like the present Γ, such corner issue could happen for any mesh resolution, no matter coarse or dense. In the present case, the failure occurs at $n = 252$, but the Cartesian AMIB scheme works for other n values. With the proposed corner treatments, the ray-casting AMIB scheme maintains the fourth order accuracy for $[n, n] = [252, 252]$. The numerical solution and error of the ray-casting AMIB method are depicted in Fig. 8 by considering the Dirichlet boundary condition and taking $n = 252$. It is observed that the maximal errors occur around the five-leaf interface Γ, while the errors near boundary are almost the same as regular points.

In terms of efficiency, it can be seen from Table 2 and Table 3 that the iteration numbers for the Robin condition are larger than those for the Dirichlet condition. Thus, the CPU time becomes at least doubled for each case when the Dirichlet



(a) L_∞ error

(b) L_2 error



(c) Iteration number

(d) CPU time

Fig. 7. Impact of highly contrasted coefficients to the ray-casting MIB, ray-casting AMIB, Cartesian MIB, and Cartesian AMIB methods for the 2D circular interface problem with a fixed β^- being 1 and β^+ ranging from 10^{-4} to 10^9.

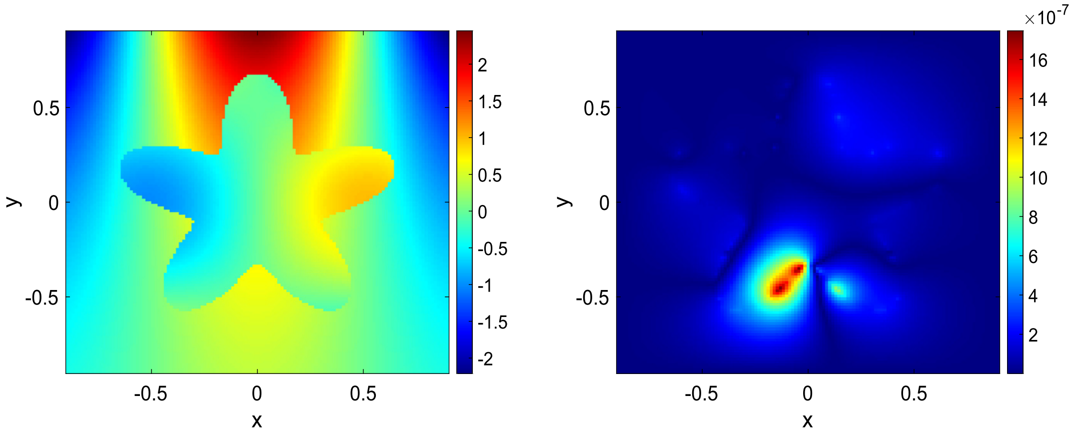


Fig. 8. The numerical solution (left) and error (right) of the ray-casting AMIB method for Example 2a on a mesh with $n = 252$.

Table 2
Example 2a – $\beta^+ = 1, \beta^- = 10$; five-leaf interface; Dirichlet boundary condition.

$[n, n]$	Ray-casting AMIB					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[60, 60]	3.005E-4	–	7.832E-5	–	48	0.104
[124, 124]	1.559E-5	4.2688	3.044E-6	4.6856	53	0.308
[252, 252]	1.310E-6	3.5728	2.163E-7	3.8146	99	1.541
[508, 508]	1.644E-7	2.9948	2.747E-8	2.9772	88	5.014
$[n, n]$	Cartesian AMIB [18]					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[60, 60]	1.348E-4	–	3.897E-5	–	46	0.134
[124, 124]	7.093E-6	4.2487	1.628E-6	4.5810	53	0.278
[252, 252]	–	–	–	–	–	–
[508, 508]	3.082E-8	3.9232	7.153E-9	3.9152	75	4.331

Table 3
Example 2b – $\beta^+ = 1, \beta^- = 10$; five-leaf interface; Robin boundary condition.

$[n, n]$	Ray-casting AMIB					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[60, 60]	3.338E-4	–	1.093E-4	–	58	9.066E-2
[124, 124]	1.628E-5	4.3580	3.823E-6	4.8386	76	0.342
[252, 252]	1.363E-6	3.5782	2.731E-7	3.8070	144	2.255
[508, 508]	1.739E-7	2.9702	3.915E-8	2.8024	172	11.365
$[n, n]$	Cartesian AMIB [18]					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[60, 60]	1.446E-4	–	5.018E-5	–	59	0.104
[124, 124]	7.671E-6	4.2366	2.007E-6	4.6444	80	0.430
[252, 252]	–	–	–	–	–	–
[508, 508]	3.164E-8	3.9608	9.567E-9	3.8562	180	13.534

condition is replaced with the Robin condition, owning to the difficult nature of the Robin condition. On the other hand, the ray-casting AMIB scheme is more efficient than the Cartesian AMIB scheme in case of a Robin boundary, while it is slower for the Dirichlet case.

3.2. Numerical examples in 3D

We next validate the ray-casting MIB and AMIB schemes in 3D. The Cartesian AMIB scheme [18] was developed only for 2D elliptic interface problems. Moreover, the above 2D studies show that the Cartesian AMIB scheme suffers from corner issues, while such issues are rigorously treated in the ray-casting AMIB scheme. Since the corner issues happen more frequently in 3D, we will only consider ray-casting schemes for 3D examples. For simplicity, we will refer to the ray-casting MIB and AMIB as the MIB and AMIB, respectively. In each 3D test case, the number of corner points which have been treated in our computation will be reported.

Example 3. Consider a 3D Poisson's equation

$$(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z = q(x, y, z), \tag{33}$$

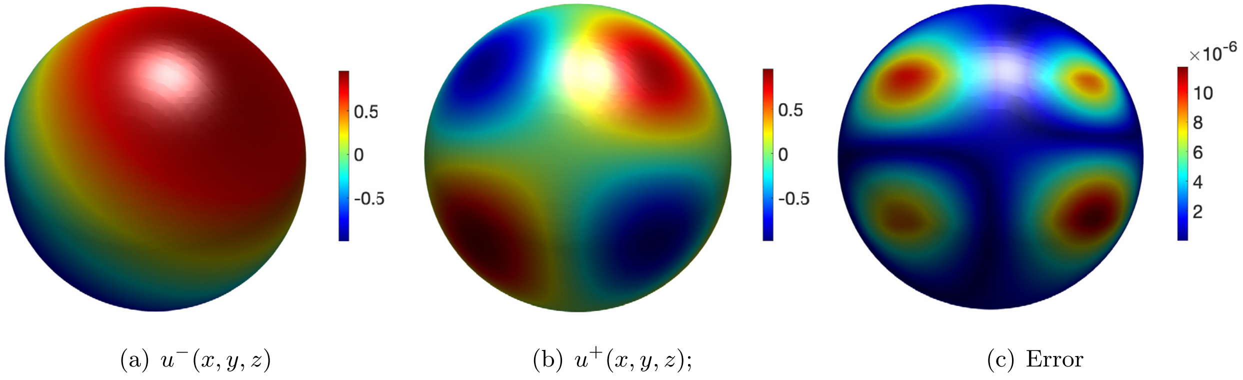
over a cubic domain $[-\frac{\pi}{3}, \frac{\pi}{3}] \times [-\frac{\pi}{3}, \frac{\pi}{3}] \times [-\frac{\pi}{3}, \frac{\pi}{3}]$ with a spherical interface defined by

$$\Gamma : x^2 + y^2 + z^2 = 0.4^2.$$

The analytical solution to this problem is constructed to be

$$u(x, y, z) = \begin{cases} \sin(x + y + z), & \text{inside } \Gamma, \\ \sin(kx)\sin(ky)\sin(kz), & \text{otherwise,} \end{cases} \tag{34}$$

with $k = 3$. The source term $q(x, y, z)$ is related to the above designated solution,



(a) $u^-(x,y,z)$ (b) $u^+(x,y,z)$; (c) Error

Fig. 9. The plots of exact solutions and numerical error of the ray-casting AMIB scheme by mapping onto the spherical surface Γ, based on a mesh with $n = 128$.

$$q(x,y,z) = \begin{cases} -3\beta^- \sin(x+y+z), & \text{inside } \Gamma, \\ -3k^2\beta^+ \sin(kx)\sin(ky)\sin(kz), & \text{otherwise.} \end{cases}$$

On the boundary of the cubic domain, a Dirichlet boundary condition is assumed with the boundary data derived by the analytical solution.

By considering $\beta^+ = 100$ and $\beta^- = 1$, the analytical solutions $u^-(x,y,z)$ and $u^+(x,y,z)$ are depicted in Fig. 9. In particular, the graphs are generated by mapping the solutions onto the spherical surface Γ from inside and outside, respectively, for $u^-(x,y,z)$ and $u^+(x,y,z)$. While two solutions are smooth in their subdomains, the entire solution $u(x,y,z)$ is obviously discontinuous across Γ.

In our computation, since the solution $u^+ = \sin(kx)\sin(ky)\sin(kz)$ naturally satisfies the anti-symmetry property across the boundary of the cubic domain $[-\frac{\pi}{3}, \frac{\pi}{3}] \times [-\frac{\pi}{3}, \frac{\pi}{3}] \times [-\frac{\pi}{3}, \frac{\pi}{3}]$, there is no need to introduce an extended domain Ω^e and additional interface Γ_2. Without using the MIB boundary closure [57,17], this allows us to solely focus on the performance of the ray-casting scheme for treating the surface Γ. For both MIB and AMIB schemes, the same mesh size is used, i.e., $(n+1) \times (n+1) \times (n+1)$.

The numerical results of the MIB and AMIB schemes are reported in Table 4, in which the number of corner points is reported for each tested n value. It can be seen that the AMIB scheme attains a fourth order of accuracy, while the MIB scheme converges slower for $n = 128$ and fails for $n = 256$. Similar to Example 1 in 2D, the AMIB method is well conditioned such that the iteration number only weakly depends on the mesh size n. Thus, the AMIB method is much faster than the MIB method. Because the iteration number increases dramatically for the MIB scheme, the MIB computation on a mesh $257 \times 257 \times 257$ was terminated by the system due to an excessive long runtime. In Table 4, the gradient approximation by the AMIB scheme is also reported. Based on the numerical solutions, the gradient can be simply estimated, and the corresponding numerical order is also four. The numerical error of the AMIB scheme at $n = 128$ is shown in Fig. 9 (c). It is seen that large errors occur in the places where the solutions have large magnitudes.

We next increase the diffusion coefficient to be $\beta^+ = 10^4$ and $\beta^- = 1$, while keeping the other parameters unchanged. In Table 5, the numerical results of the AMIB are displayed. By comparing with Table 4, we found that the changes in accuracies and orders are very minor, but the iteration number becomes larger at the densest mesh. This means that a high jump ratio may affect the efficiency of the AMIB method when a very dense mesh is used in 3D.

Example 4. We consider a topologically shaped interface in this example. Because the MIB scheme is too slow for 3D computations, we will only consider the ray-casting AMIB scheme from now on. The 3D Poisson's equation (33) is studied with a tanglecube surface defined as

$$\Gamma : 8x^4 + 8y^4 + 8z^4 - 10x^2 - 10y^2 - 10z^2 = -5,$$

embedded in a computational domain of dimension $[-\frac{2\pi}{3}, \frac{2\pi}{3}] \times [-\frac{2\pi}{3}, \frac{2\pi}{3}] \times [-\frac{2\pi}{3}, \frac{2\pi}{3}]$. Two exact solutions of Poisson equation are designated by

$$u(x,y,z) = \begin{cases} \cos(kx)\sin(ky)\cos(kz), & \text{inside } \Gamma, \\ \sin(kx)\sin(ky)\sin(kz), & \text{otherwise,} \end{cases} \tag{35}$$

with $k = 3$. The corresponding source terms are

$$q(x,y,z) = \begin{cases} -3k^2\beta^- \cos(kx)\sin(ky)\cos(kz), & \text{inside } \Gamma, \\ -3k^2\beta^+ \sin(kx)\sin(ky)\sin(kz), & \text{otherwise.} \end{cases} \tag{36}$$

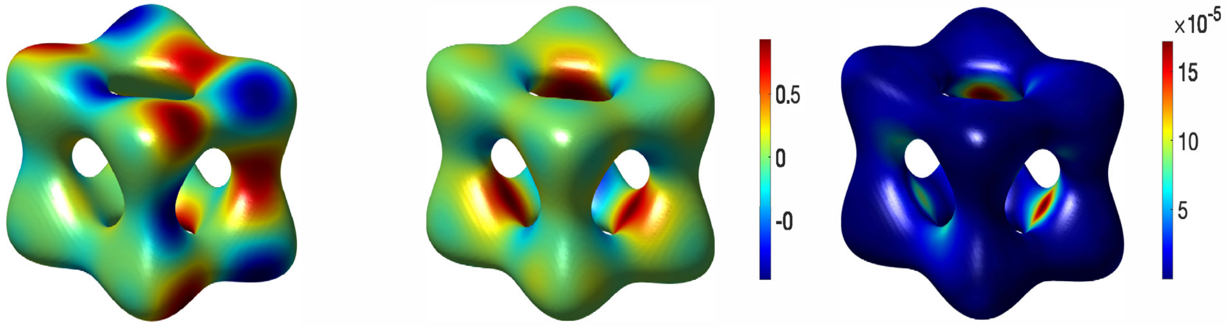
By taking $\beta^+ = 20$ and $\beta^- = 1$, the analytical solutions $u^-(x,y,z)$ and $u^+(x,y,z)$ are depicted in Fig. 10. It can be seen that the tanglecube surface is of a complicated shape, and the solution $u(x,y,z)$ is discontinuous.

Table 4
Example 3a – $\beta^+ = 100, \beta^- = 1$; Spherical interface. The number of corner points is 24, 24, 0 and 0, respectively, for n =32, 64, 128, and 256.

$[n, n, n]$	AMIB Solution					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[32, 32, 32]	3.846E-4	–	3.228E-5	–	19	0.845
[64, 64, 64]	3.948E-5	3.2839	1.748E-6	4.2068	23	5.112
[128, 128, 128]	1.745E-6	4.5002	8.203E-8	4.4133	50	60.637
[256, 256, 256]	7.582E-8	4.5243	4.120E-9	4.3153	78	654.356
$[n, n, n]$	AMIB Gradient					
	L_∞		L_2			
	Error	Order	Error	Order		
[32, 32, 32]	5.692E-3	–	5.502E-4	–		
[64, 64, 64]	7.936E-4	2.8424	2.815E-5	4.2888		
[128, 128, 128]	3.527E-5	4.4919	1.251E-6	4.4926		
[256, 256, 256]	3.576E-6	3.3021	6.041E-8	4.3717		
$[n, n, n]$	MIB Solution					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[32, 32, 32]	1.732E-2	–	2.613E-4	–	455	15.322
[64, 64, 64]	2.846E-4	5.9274	3.361E-6	6.2807	1546	257.489
[128, 128, 128]	1.253E-4	1.1837	5.368E-8	2.6466	5074	5399.483
[256, 256, 256]	–	–	–	–	–	–

Table 5
Example 3b – $\beta^+ = 10^4, \beta^- = 1$; Spherical interface. The number of corner points is 24, 24, 0 and 0, respectively, for n =32, 64, 128, and 256.

$[n, n, n]$	AMIB Solution					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[32, 32, 32]	4.007E-4	–	3.366E-5	–	19	0.753
[64, 64, 64]	4.370E-5	3.1968	1.939E-6	4.1178	23	6.067
[128, 128, 128]	1.832E-6	4.5761	8.636E-8	4.4891	55	73.805
[256, 256, 256]	8.105E-8	4.4985	4.404E-9	4.2933	186	1462.9773



(a) $u^-(x, y, z)$ (b) $u^+(x, y, z)$; (c) Error

Fig. 10. The plots of exact solutions and numerical error of the ray-casting AMIB scheme by mapping onto the tanglecube surface Γ, based on a mesh with $n = 128$.

A simple Dirichlet zero boundary condition can be assumed for this example so that the anti-symmetry property is satisfied again at the boundary. The mesh size of the AMIB scheme is $(n+1) \times (n+1) \times (n+1)$. The numerical results of the AMIB scheme are reported in Table 6. Obviously, the AMIB scheme achieves fourth order convergence in both solution and gradient. In terms of efficiency, the iteration number of the AMIB is larger than Example 3a on the same mesh size, even if the ratio of β^+ over β^- is not that large. This is mainly because the tanglecube surface is much more complicated than spherical surface such that the AMIB in this example involves more auxiliary variables at the interface. The numerical error of the AMIB scheme is demonstrated in Fig. 10 (c). It is found that the error distribution pattern is quite concentrated, due to the fact that the large errors usually occur on or near the interface.

Table 6
Example 4 – $\beta^+ = 20, \beta^- = 1$; Tanglecube interface; The number of corner points is 72, 48, 120 and 24, respectively, for $n = 64$, 128, and 256, 512.

$[n, n, n]$	AMIB Solution					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[64, 64, 64]	1.808E-3	–	1.159E-4	–	40	13.53
[128, 128, 128]	1.742E-4	3.3757	9.787E-6	3.5657	53	99.15
[256, 256, 256]	8.880E-6	4.2938	5.433E-7	4.1712	85	929.29
[512, 512, 512]	4.404E-7	4.3336	3.008E-8	4.1750	90	9514.02

$[n, n, n]$	AMIB Gradient			
	L_∞		L_2	
	Error	Order	Error	Order
[64, 64, 64]	5.694E-2	–	1.080E-3	–
[128, 128, 128]	4.231E-3	3.7503	6.637E-5	4.0243
[256, 256, 256]	3.441E-4	3.6200	3.184E-6	4.3819
[512, 512, 512]	6.255E-5	2.4600	1.649E-7	4.2707

Table 7
Example 5 – $\beta^+ = 20, \beta^- = 1$; Flower interface. The number of corner points is 27, 57, 37, 73, respectively, for $n = 32$, 64, 128, and 256.

$[n, n, n]$	AMIB Solution					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[32, 32, 32]	1.656E-2	–	1.309E-3	–	41	0.856
[64, 64, 64]	9.682E-4	4.0960	5.457E-5	4.5846	62	8.821
[128, 128, 128]	2.231E-4	2.1172	8.238E-6	2.7277	117	113.313
[256, 256, 256]	8.529E-6	4.7095	4.337E-7	4.2475	140	1046.385

Example 5. Reusing the exact solution (35) and source term (36) with $k = 6$, one more example is considered to solve 3D Poisson's equation (33) with a non-smooth flower-like interface [55]

$$\Gamma : r = 0.6 + 0.12 \sin 5\theta, \quad -0.3 \leqslant z \leqslant 0.3,$$

where $r = \sqrt{x^2 + y^2}$ and $\theta = \arctan(\frac{x}{y})$. The computational domain is set to $[-\frac{\pi}{3}, \frac{\pi}{3}] \times [-\frac{\pi}{3}, \frac{\pi}{3}] \times [-\frac{\pi}{3}, \frac{\pi}{3}]$, and the diffusion coefficients β^+ and β^- are chosen as 20 and 1, respectively.

With a Dirichlet boundary condition, the anti-symmetry property is satisfied. The numerical results are presented in Table 7. It is noted that the horizontal cross-section of the present 3D shape is similar to the 2D five-leaf interface studied in Example 2. Thus, many fictitious values can be generated by using ray-casting lines completely contained in some xy planes. In other words, they are generated in a 2D manner. For the 3D flower interface, the AMIB scheme achieves the fourth-order convergence in Table 7. The flower interface is depicted in Fig. 11 with surface maps of exact solutions and numerical errors. It is found that large numerical errors occur where there are large curvatures.

Example 6. In this example, a Poisson-Boltzmann type problem

$$(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z - \beta u = q(x, y, z),$$

is studied with a smooth molecular surface of two atoms defined by

$$\Gamma : (x^2 + y^2 + z^2 + \frac{3}{5})^2 - \frac{5}{2} y^2 = \frac{3}{5}.$$

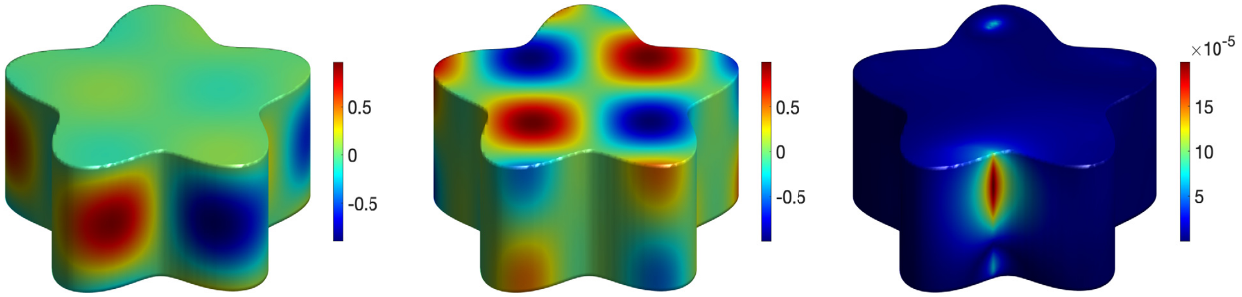
With the parameter $k = 3$, the exact solutions are constructed to be

$$u(x, y, z) = \begin{cases} e^{-x^2 - y^2/2 - z^2/2} & \text{inside } \Gamma, \\ \sin(kx)\sin(ky)\sin(kz) & \text{otherwise,} \end{cases}$$

and the corresponding source terms are

$$q(x, y, z) = \begin{cases} (-4 + 4x^2 + y^2 + z^2)\beta^- e^{-x^2 - y^2/2 - z^2/2} & \text{inside } \Gamma, \\ -3k^2 \beta^+ \sin(kx)\sin(ky)\sin(kz) & \text{otherwise.} \end{cases}$$

The plots of solutions u^- and u^+ mapped onto the molecular surface are given in Fig. 12.



(a) $u^-(x,y,z)$ (b) $u^+(x,y,z)$; (c) Error

Fig. 11. The plots of exact solutions and numerical error of the ray-casting AMIB scheme by mapping onto the flower surface Γ, based on a mesh with $n = 128$.

Table 8
Example 6 – $\beta^+ = 20$, $\beta^- = 1$; Molecular interface of two atoms. The number of corner points is 8, 8, 0 and 8, respectively, for n =32, 64, 128, and 256.

$[n, n, n]$	AMIB Solution					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[32, 32, 32]	2.193E-3	–	1.882E-4	–	26	0.774
[64, 64, 64]	1.740E-4	3.6559	8.652E-6	4.4429	31	6.295
[128, 128, 128]	1.790E-5	3.2811	4.572E-7	4.2422	49	61.904
[256, 256, 256]	2.019E-6	3.1485	2.758E-8	4.0513	79	664.513

The numerical results are presented in Table 8 for $\beta^+ = 20$ and $\beta^- = 1$. Again, a simple Dirichlet boundary condition is used, with the anti-symmetry property satisfied. Table 8 shows that the AMIB method is equally capable of solving 3D Poisson-Boltzmann equation. The fourth order convergences are again confirmed in the solution. The efficiency of the AMIB is further validated. The error plot in Fig. 12 shows that the maximum errors occur in the regions where the largest jumps take place across the interface.

Example 7. We next consider a 3D Helmholtz equation

$$(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z + \beta u = q(x, y, z).$$

The interface is a smooth torus surface

$$\Gamma : (0.9 - \sqrt{x^2 + y^2})^2 + z^2 = 0.2,$$

restricted on a cubic domain $[-\frac{2\pi}{3}, \frac{2\pi}{3}] \times [-\frac{2\pi}{3}, \frac{2\pi}{3}] \times [-\frac{2\pi}{3}, \frac{2\pi}{3}]$. Two exact solutions of Poisson equation are designated by

$$u(x, y, z) = \begin{cases} \cos(x + y + z) & \text{inside } \Gamma, \\ \sin(kx)\sin(ky)\sin(kz) & \text{otherwise,} \end{cases}$$

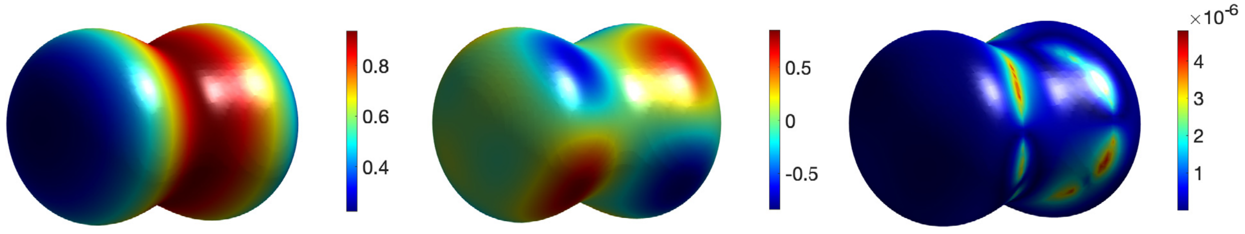
with $k = 3$. The corresponding source terms are

$$q(x, y, z) = \begin{cases} -3\beta^- \cos(x + y + z) & \text{inside } \Gamma, \\ -3k^2\beta^+ \sin(kx)\sin(ky)\sin(kz) & \text{otherwise.} \end{cases}$$

The torus surface and solutions u^- and u^+ are shown in Fig. 13.

By taking $\beta^+ = 1$ and $\beta^- = 20$, the numerical results are presented in Table 9. The fourth order convergence is attained for both solution and gradient. The iteration numbers of the AMIB are larger than the previous examples on the same mesh size, probably due to the fact that the β contrast ratio is reversed. Nevertheless, the grow rate of the iteration number with respect to the mesh size n is still moderate, so that the efficiency of the AMIB is not reduced. The numerical error is shown in Fig. 13. It can be seen that the largest errors occur at regions where the solution changes rapidly. (See Fig. 13.)

Example 8. In the last example, we solve the 3D Poisson's equation (33) with different boundary conditions. The interface is a simple cylinder surface, which is defined as



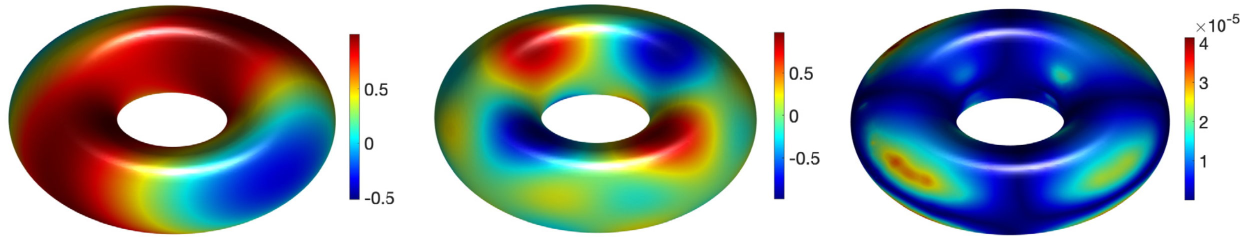
(a) $u^-(x,y,z)$ (b) $u^+(x,y,z)$; (c) Error

Fig. 12. The plots of exact solutions and numerical error of the ray-casting AMIB scheme by mapping onto the molecular surface Γ, based on a mesh with $n = 128$.

Table 9
Example 7 – $\beta^+ = 1, \beta^- = 20$; Torus interface. The number of corner points is 16, 8, 8 and 0, respectively, for n =64, 128, 256, and 512.

$[n,n,n]$	AMIB Solution					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[64, 64, 64]	5.451E-4	–	3.916E-5	–	75	13.673
[128, 128, 128]	4.481E-5	3.6044	3.405E-6	3.5236	76	97.767
[256, 256, 256]	3.101E-6	3.8531	2.195E-7	3.9555	92	813.712
[512, 512, 512]	1.825E-7	4.0865	1.317E-8	4.0588	109	8678.912

$[n,n,n]$	AMIB Gradient			
	L_∞		L_2	
	Error	Order	Error	Order
[64, 64, 64]	1.807E-2	–	3.924E-4	–
[128, 128, 128]	2.240E-3	3.0115	2.443E-5	4.0057
[256, 256, 256]	7.199E-4	1.6377	1.448E-6	4.0761
[512, 512, 512]	3.238E-5	4.4746	7.922E-8	4.1924



(a) $u^-(x,y,z)$ (b) $u^+(x,y,z)$; (c) Error

Fig. 13. The plots of exact solutions and numerical error of the ray-casting AMIB scheme by mapping onto the torus surface Γ, based on a mesh with $n = 128$.

$$\Gamma : x^2 + y^2 = 0.7^2, \quad -0.7 \leqslant z \leqslant 0.7.$$

The analytical solution to this problem is constructed to be

$$u(x,y,z) = \begin{cases} \cos(kx)\sin(ky)\cos(kz), & \text{inside } \Gamma, \\ \sin(kx)e^{y+z}, & \text{otherwise}, \end{cases} \tag{37}$$

with $k = 3$. The source term $q(x,y,z)$ is related to the above designated solution,

$$q(x,y,z) = \begin{cases} -3k^2\beta^- \cos(kx)\sin(ky)\cos(kz), & \text{inside } \Gamma, \\ (2-k^2)\beta^+ \sin(kx)e^{y+z}, & \text{otherwise}. \end{cases}$$

The computational domain is fixed to be $[-\frac{2\pi}{3.5}, \frac{2\pi}{3.5}] \times [-\frac{2\pi}{3.5}, \frac{2\pi}{3.5}] \times [-\frac{2\pi}{3.5}, \frac{2\pi}{3.5}]$. The diffusion coefficient is chosen as $\beta^+ = 10$ and $\beta^- = 1$.

We note that for the present $u^+(x,y,z)$ and domain, the anti-symmetric property is not satisfied across the boundary. Consequently, the boundary will be treated as an immersed interface Γ_2 with an extended domain Ω^e as the zero-padding

Table 10
Example 8a – $\beta^+ = 10, \beta^- = 1$; Cylinder interface; Dirichlet boundary condition. The number of corner points is 0, 0, 0 and 0, respectively, for n =28, 60, 124, and 252.

$[n, n, n]$	AMIB					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[28, 28, 28]	2.790E-2	–	2.228E-3	–	27	0.648
[60, 60, 60]	1.727E-3	4.0138	1.126E-4	4.3062	22	3.383
[124, 124, 124]	3.680E-4	2.2307	1.098E-5	3.3589	52	51.735
[252, 252, 252]	3.903E-6	6.5587	3.610E-7	4.9264	52	385.434

Table 11
Example 8b – $\beta^+ = 10, \beta^- = 1$; Cylinder interface; mixed boundary conditions. The number of corner points is 0, 0, 0 and 0, respectively, for n =28, 60, 124, and 252.

$[n, n, n]$	AMIB					
	L_∞		L_2		iter no.	CPU time (s)
	Error	Order	Error	Order		
[28, 28, 28]	1.147E-2	–	1.536E-3	–	32	0.792
[60, 60, 60]	7.840E-4	3.8709	1.126E-4	3.7699	39	6.081
[124, 124, 124]	6.348E-5	3.6263	7.655E-6	3.8791	83	85.114
[252, 252, 252]	2.919E-6	4.4428	4.056E-7	4.2383	144	1047.582

zone [17]. The mesh size for the entire domain D is taken as $(n+5) \times (n+5) \times (n+5)$. Two types of boundary conditions will be studied for this example, i.e., a simple Dirichlet condition and mixed conditions. For the mixed boundary value problem, a Robin boundary condition will be assumed on four sides of Γ_2, while a Neumann boundary condition is used for the other two sides,

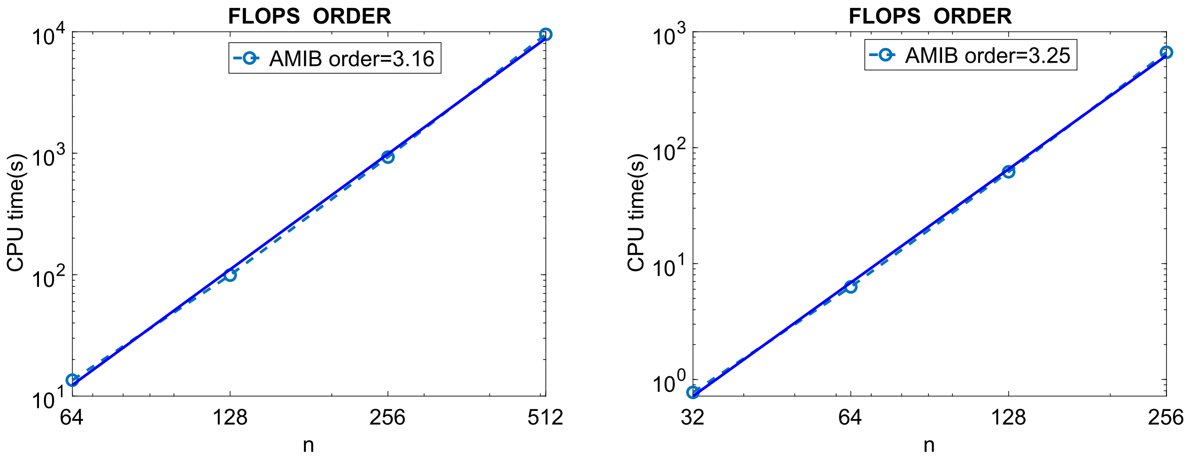
$$\begin{aligned} u + \frac{\partial u}{\partial n} &= g_1(x, y, z), \qquad \text{on} \;\; \Gamma_2^1, \Gamma_2^3, \Gamma_2^4, \Gamma_2^6 \\ \frac{\partial u}{\partial n} &= g_2(x, y, z), \qquad \text{on} \;\; \Gamma_2^5, \Gamma_2^2 \end{aligned}$$

where $\vec{n}$ denotes the outward normal direction. The boundary data, such as $g_1(x, y, z)$ and $g_2(x, y, z)$, are determined by the analytical solution.

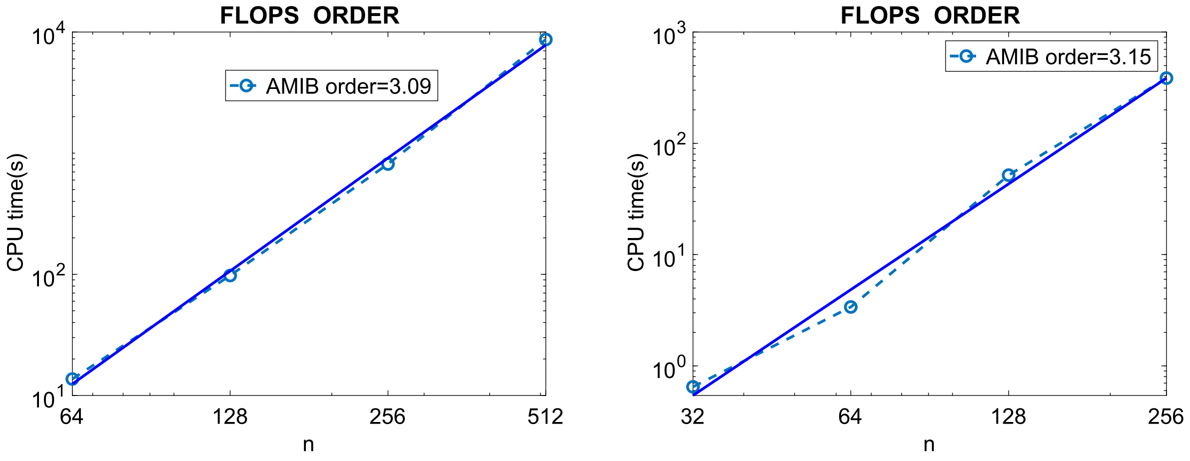
The numerical results of the proposed AMIB scheme are listed in Table 10 and Table 11, respectively, for the Dirichlet problem and mixed boundary problem. It can be observed that the AMIB scheme achieves fourth order of convergence in both cases. In case of the Dirichlet boundary condition, the iteration number of the AMIB scheme does not grow too much as the mesh is refined. Similar to the previous 2D study, because of the impact of Robin boundary conditions, the iteration number of the AMIB scheme becomes larger in the mixed problem. Nevertheless, the AMIB scheme still maintains a high efficiency for this challenging problem. In fact, even without the material interface Γ_1, the numerical solution of elliptic boundary value problems with mixed boundary conditions is still a great challenge to other numerical methods. The AMIB scheme is the only known method that attains a fourth order accuracy and FFT efficiency [17,46].

3.3. Computational efficiency

In this subsection, the computational efficiency of the AMIB method for 3D elliptic interface problems will be further investigated. In our previous study [18], the Cartesian AMIB scheme for 2D elliptic interface problems can deliver a computational complexity of $O(n^2 \log n)$ on a $n \times n$ mesh. For 3D problems, our brief complexity analysis given at the end of Section 2 indicates that the proposed ray-casting AMIB scheme can maintain the $O(n^3 \log n)$ efficiency of the FFT algorithm. Based on the computational time of the AMIB method reported in the 3D examples above, the complexity of AMIB method will be numerically quantified. Note that the mesh size is $(n+5) \times (n+5) \times (n+5)$ in Example 8, while in all other 3D examples, it is $(n+1) \times (n+1) \times (n+1)$. For simplicity, we regard the degree of freedom in each direction as n. The CPU time of the AMIB scheme is depicted against n in log-log scale in Fig. 14 for four examples. It can be seen that in all cases, the log-log plots are almost linear. A least squares fitting is conducted to compute a flop order r in $O(n^r)$ setting, and the corresponding order r is shown in the legend of each subfigure. From Fig. 14, it can be observed that the flop order r for the ray-casting AMIB method is slightly above 3. One can conclude that the computational cost of the AMIB method in 3D is roughly on the order of $O(n^3 \log n)$.



(a) Example 4. (b) Example 6.



(c) Example 7. (d) Example 8a.

Fig. 14. Flops order in CPU time is examined for several 3D examples. The numerical flop order r is slightly larger than 3 in all examples.

4. Conclusion

In this work, we proposed a fast and fourth order accurate finite difference method to solve 3D elliptic interface problems. A novel ray-casting MIB scheme is proposed to generate fictitious values for treating a smoothly curved interface in two and three dimensions. Outside the boundary, two layers of zero-padding solutions are introduced, and the MIB boundary closure scheme [57,17] is employed to generate fictitious values in dealing with Dirichlet, Neumann, Robin boundary conditions, and their mix combinations. Based on the fictitious values generated near interfaces and boundaries, a systematic approach is constructed to correct the fourth order central difference discretization of Laplacian operator, so that the discrete Laplacian can be efficiently inverted by the FFT Poisson solver. In the Schur complement solution of the augmented problem, the iteration number of the AMIB method only weakly depends on the mesh size n. Thus, the proposed ray-casting AMIB method not only achieves a fourth order of accuracy for complex interfaces and boundaries, but also delivers an overall complexity of $O(n^3 \log n)$ on a $n \times n \times n$ mesh. Moreover, with a little additional computation, the AMIB method can produce fourth order accurate approximation of solution gradients.

Comparing with the fourth order FFT-AMIB algorithm for 2D elliptic interface problems [18], the novelty of the present development lies in two aspects.

- First, the present AMIB scheme is based on a novel interface algorithm. In [18], the classical MIB scheme [60,54,55] is employed, in which the interface jump conditions are decomposed into Cartesian directions and the resulted Cartesian jump conditions are discretized simultaneously in all involved Cartesian directions. In the proposed ray-casting scheme, the jump conditions are simply discretized along the normal direction for fictitious value generation, and the involved auxiliary points are interpolated along 2D planes into grid nodes. Note that 2D interpolation/extrapolation is also needed in the classical MIB scheme [60,54,55] after discretizing jump conditions. Thus, comparing with the classical MIB, the ray-casting MIB scheme is simpler and easier to implement in 3D. Moreover, because the normal direction makes it possible to reach more grid points inside/outside the interface, compared to the classical MIB, the ray-casting MIB scheme can handle more complicated geometry.

- Second, the corner issues are rigorously treated in the proposed AMIB method. As shown in our 2D computations, the existing AMIB method [18] fails when corner points are presented. Unfortunately, corner points are frequently encountered in 3D, so that the corner treatment becomes indispensable in solving 3D elliptic interface problems. In the proposed ray-casting MIB scheme, by using a sufficiently dense grid, the corner issue can be naturally bypassed, because the normal line locally will not cut the interface for a second time. In correcting the fourth order central difference, all corner cases have been examined, and the corresponding correcting formulas involving multiple interface cuts have been developed in the proposed ray-casting AMIB scheme. Therefore, the ray-casting AMIB scheme becomes more robust than the previous AMIB scheme [18].

In the future, the extension of the proposed ray-casting AMIB method to more challenging interface problems will be explored. Similar to the classical MIB scheme [60,54,55] and the existing AMIB scheme [16–18,46,35], the ray-casting AMIB scheme is independent of the governing PDE. This will allow us to generalize the ray-casting AMIB algorithm for solving parabolic and hyperbolic PDE interface problems. The parallelization is a challenging issue for the ray-casting AMIB scheme, because a fictitious value is related to many surrounding grid points. For a domain-decomposition type parallelization, the partition of the grid could be cumbersome, because a lot of communication processes are needed. Alternative approaches have to be explored for the parallelization.

CRediT authorship contribution statement

Yiming Ren: Methodology, Software, Validation, Visualization, Writing – original draft. **Shan Zhao:** Conceptualization, Methodology, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This research is partially supported by the National Science Foundation (NSF) of USA grant DMS-2110914.

References

[1] L. Adams, Z.L. Li, The immersed interface/multigrid methods for interface problems, SIAM J. Sci. Comput. 24 (2002) 463–479.
[2] I. Babuška, The finite element method for elliptic equations with discontinuous coefficients, Computing 5 (1970) 207–213.
[3] J. Bedrossian, J.H. von Brecht, S.W. Zhu, E. Sifakis, J.M. Teran, A finite element method for interface problems in domains with smooth boundaries and interfaces, J. Comput. Phys. 229 (2010) 6405–6426.
[4] D. Bochkov, F. Gibou, Solving elliptic interface problems with jump conditions on Cartesian grids, J. Comput. Phys. 407 (2020) 109269.
[5] J. Bramble, J. King, A finite element method for interface problems in domains with smooth boundaries and interfaces, Adv. Comput. Math. 6 (1996) 109–138.
[6] P.A. Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, J. Comput. Phys. 197 (2004) 364–386.
[7] G. Brandstetter, S. Govindjee, A high-order immersed boundary discontinuous-Galerkin method for Poisson's equation with discontinuous coefficients and singular sources, Int. J. Numer. Methods Eng. 101 (2015) 847–869.
[8] T. Chen, J. Strang, Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems, J. Comput. Phys. 227 (2008) 7503–7542.
[9] X. Chen, X. Feng, Z. Li, A direct method for accurate solution and gradient computations for elliptic interface problems, Numer. Algorithms 80 (2019) 709–740.
[10] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math. 79 (1998) 175–202.
[11] I.-L. Chern, Y.-C. Shu, A coupling interface method for elliptic interface problems, J. Comput. Phys. 225 (2007) 2138–2174.
[12] C. Chu, I.G. Graham, T. Hou, A new multiscale finite element method for high-contrast elliptic interface problems, Math. Comput. 79 (2010) 1915–1955.
[13] A. Coco, G. Russo, Second order finite-difference ghost-point multigrid methods for elliptic problems with discontinuous coefficients on an arbitrary interface, J. Comput. Phys. 361 (2018) 299–330.
[14] R. Egan, F. Gibou, xGFM: recovering convergence of fluxes in the ghost fluid method, J. Comput. Phys. 409 (2020) 109351.
[15] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[16] H. Feng, G. Long, S. Zhao, An augmented matched interface and boundary (MIB) method for solving elliptic interface problem, J. Comput. Appl. Math. 361 (2019) 426–433.

[17] H. Feng, S. Zhao, FFT-based high order central difference schemes for three-dimensional Poisson's equation with various types of boundary conditions, J. Comput. Phys. 410 (2020) 109391.

[18] H. Feng, S. Zhao, A fourth order finite difference method for solving elliptic interface problems with the FFT acceleration, J. Comput. Phys. 419 (2020) 109677.

[19] Q.W. Feng, B. Han, P. Minev, Sixth order compact finite difference scheme for Poisson interface problem with singular sources, Comput. Math. Appl. 99 (2021) 2–25.

[20] Q.W. Feng, B. Han, P. Minev, A high order compact finite difference scheme for elliptic interface problems with discontinuous and high-contrast coefficients, Appl. Math. Comput. 431 (2022) 127314.

[21] J. Fernández-Fidalgo, S. Clain, L. Ramírez, I. Colominas, X. Nogueira, Very high-order method on immersed curved domains for finite difference schemes with regular Cartesian grids, Comput. Methods Appl. Mech. Eng. 360 (2020) 112782.

[22] T.-P. Fries, T. Belytschko, The intrinsic XFEM: a method for arbitrary discontinuities without additional unknowns, Int. J. Numer. Methods Eng. 68 (2006) 1358–1385.

[23] W. Geng, S. Zhao, A two-component Matched Interface and Boundary (MIB) regularization for charge singularity in implicit solvation, J. Comput. Phys. 351 (2017) 25–39.

[24] F. Gibou, R.P. Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. 202 (2005) 577–601.

[25] H. Guo, X. Yang, Gradient recovery for elliptic interface problem: II. Immersed finite element methods, J. Comput. Phys. 338 (2017) 606–619.

[26] J.L. Hellrung Jr., L.M. Wang, E. Sifakis, J.M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions, J. Comput. Phys. 231 (2012) 2015–2048.

[27] T.Y. Hou, Z.L. Li, S. Osher, H. Zhao, A hybrid method for moving interface problems with application to the Hele-Shaw flow, J. Comput. Phys. 134 (1997) 236–252.

[28] X. He, T. Lin, Y. Lin, Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions, Int. J. Numer. Anal. Model. 8 (2011) 284–301.

[29] H. Ji, J. Dolbow, On strategies for enforcing interfacial constraints and evaluating jump conditions with the extended finite element method, Int. J. Numer. Methods Eng. 61 (2004) 2508–2535.

[30] H. Ji, J. Chen, Z. Li, A high-order source removal finite element method for a class of elliptic interface problems, Appl. Numer. Math. 130 (2018) 112–130.

[31] G. Jo, D.Y. Kwak, Geometric multigrid algorithms for elliptic interface problems using structured grids, Numer. Algorithms 81 (2019) 211–235.

[32] G. Jo, D.Y. Kwak, A semi-uniform multigrid algorithm for solving elliptic interface problems, Comput. Methods Appl. Math. 21 (2021) 127–143.

[33] R.J. LeVeque, Z.L. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (1994) 1019–1044.

[34] R.J. LeVeque, Z.L. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (1997) 709–735.

[35] C. Li, Y. Ren, G. Long, E. Boerman, S. Zhao, A fast Sine transform accelerated high order finite difference method for parabolic problems over irregular domains, J. Sci. Comput. (2022), revised.

[36] J. Li, J.M. Melenk, B. Wohlmuthc, J. Zou, Optimal a priori estimates for higher order finite elements for elliptic interface problems, Appl. Numer. Math. 60 (2010) 19–37.

[37] Z.L. Li, A fast iterative algorithm for elliptic interface problem, SIAM J. Numer. Anal. 35 (1998) 230–254.

[38] Z. Li, K. Ito, The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains, SIAM, 2006.

[39] Z.L. Li, H. Ji, X. Chen, Accurate solution and gradient computation for elliptic interface problems with variable coefficients, SIAM J. Numer. Anal. 55 (2017) 670–697.

[40] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157–192.

[41] C. Liu, C. Hu, A second order ghost fluid method for an interface problem of the Poisson equation, Commun. Comput. Phys. 22 (2017) 965–996.

[42] K. Liu, Q. Zou, Analysis of a special immersed finite volume method for elliptic interface problems, Int. J. Numer. Anal. Model. 16 (2019) 964–984.

[43] X. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (2000) 151–178.

[44] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220–252.

[45] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes in Fortran: the Art of Scientific Computing, 2nd edition, Cambridge University Press, 1992.

[46] Y. Ren, H. Feng, S. Zhao, A FFT accelerated high order finite difference method for elliptic boundary value problems over irregular domains, J. Comput. Phys. 448 (2022) 110762.

[47] Y. Shu, I. Chern, C. Chang, Accurate gradient approximation for complex interface problems in 3D by an improved coupling interface method, J. Comput. Phys. 275 (2014) 642–661.

[48] F. Tong, W. Wang, X. Feng, J. Zhao, Z. Li, How to obtain an accurate gradient for interface problems?, J. Comput. Phys. 405 (2020) 109070.

[49] J. Towers, Finite difference methods for discretizing singular source terms in a Poisson interface problem, Contemp. Math. 526 (2010) 359–389.

[50] A. Wiegmann, K.P. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, SIAM J. Numer. Anal. 37 (2004) 827–862.

[51] K. Xia, G.W. Wei, A Galerkin formulation of the MIB method for three dimensional elliptic interface problems, Comput. Math. Appl. 68 (2014) 719–745.

[52] Y.N. Xing, L.N. Song, X.M. He, C.X. Qiu, A generalized finite difference method for solving elliptic interface problems, Math. Comput. Simul. 178 (2020) 109–124.

[53] M. Xu, L. Zhang, E. Tohidi, A fourth-order least-squares based reproducing kernel method for one-dimensional elliptic interface problems, Appl. Numer. Math. 162 (2021) 124–136.

[54] S. Yu, Y. Zhou, G.W. Wei, Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces, J. Comput. Phys. 224 (2007) 729–756.

[55] S. Yu, G. Wei, Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities, J. Comput. Phys. 227 (2007) 602–632.

[56] C. Zhang, Z. Li, X. Yue, An acceleration technique for the augmented IIM for 3D elliptic interface problems, Numer. Math., Theory Methods Appl. 14 (2021) 773–796.

[57] S. Zhao, G.W. Wei, Matched interface and boundary (MIB) for the implementation of boundary conditions in high order central finite differences, Int. J. Numer. Methods Eng. 77 (2009) 1690–1730.

[58] S. Zhao, A fourth order finite difference method for waveguides with curved perfectly conducting boundaries, Comput. Methods Appl. Math. 199 (2010) 2655–2662.

[59] X. Zhong, A new high-order immersed interface method for solving elliptic equations with imbedded interface of discontinuity, J. Comput. Phys. 225 (2007) 1066–1099.

[60] Y.C. Zhou, S. Zhao, M. Feig, G.W. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular source, J. Comput. Phys. 213 (2006) 1–30.