Computers and Mathematics with Applications 80 (2020) 714–732

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Alternating direction ghost-fluid methods for solving the heat equation with interfaces

Chuan Li [a], Zhihan Wei [b], Guangqing Long [c], Cameron Campbell [a], Stacy Ashlyn [a], Shan Zhao [b,*]

[a] Department of Mathematics, West Chester University of Pennsylvania, West Chester, PA 19383, USA
[b] Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487, USA
[c] Department of Mathematics, Nanning Normal University, Nanning 530001, PR China

ARTICLE INFO

Article history:
Received 14 August 2019
Received in revised form 28 January 2020
Accepted 25 April 2020
Available online xxxx

Keywords:
Parabolic interface problem
Ghost fluid method
Alternating direction implicit (ADI)
Matched interface and boundary (MIB)

ABSTRACT

This work presents two new alternating direction implicit (ADI) schemes for solving parabolic interface problems in two and three dimensions. First, the ghost fluid method (GFM) is adopted for the first time in the literature to treat interface jumps in the ADI framework, which results in symmetric and tridiagonal finite difference matrices in each ADI step. The proposed GFM-ADI scheme achieves a first order of accuracy in both space and time, as confirmed by numerical experiments involving complex interface shapes and spatial–temporal dependent jumps. The GFM-ADI scheme also maintains the ADI efficiency — the computational complexity for each time step scales as $O(N)$ for a total degree of freedom N in higher dimensions. Second, a new matched interface and boundary (MIB) scheme is constructed, which downgrades the quadratic polynomial bases in the existing second order MIB to linear ones. Interestingly, the resulting MIB-ADI or mADI scheme produces the same finite difference matrices as the GFM-ADI scheme in all dimensions. Hence, the mADI scheme can be regarded as an improvement of the GFM, because it calculates tangential jumps which are omitted in the GFM. Consequently, the present mADI scheme is constantly more accurate than the GFM-ADI in all numerical examples, while keeping the same computational efficiency. Nevertheless, the mADI scheme is semi-implicit due to tangential jump approximations, while the GFM-ADI scheme is fully implicit without tangential corrections. Thus, the GFM-ADI scheme could be more stable than the mADI scheme when a huge contrast is presented in diffusion coefficients.

1. Introduction

Consider a parabolic partial differential equation (PDE)

$$\frac{\partial u}{\partial t} = \nabla \cdot (\alpha \nabla u) + f, \quad \vec{x} \in \Omega, \tag{1}$$

with appropriate initial and boundary conditions

$$\begin{aligned} u(\vec{x}, 0) &= u_0(\vec{x}) \quad \text{for all } \vec{x} \in \Omega, \\ u(\vec{x}, t) &= g(\vec{x}, t) \quad \text{for all } \vec{x} \in \partial\Omega \end{aligned} \tag{2}$$

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

https://doi.org/10.1016/j.camwa.2020.04.027
0898-1221/

over a finite domain $\Omega \subset R^1, R^2,$ or R^3. Here domain Ω is split by a closed material *interface* Γ into two subdomains, $\Omega^+ \cup \Omega^- = \Omega$ and $\Omega^+ \cap \Omega^- = \Gamma$, where Ω^- and Ω^+ denote the *inner* and *outer subdomains*, respectively. In Eq. (1), the diffusion coefficient α, in general, is naturally assumed to be spatially dependent and possibly discontinuous across the interface Γ. For the seek of simplicity, α is assumed to be piece-wisely constant in this work. That is, $\alpha = \alpha^-$ in Ω^- and $\alpha = \alpha^+$ in Ω^+, where $\alpha^-, \alpha^+ > 0$ and $\alpha^- \neq \alpha^+$. Meanwhile, the source $f(\vec{x}, t)$ is also allowed to be discontinuous, even singular, across the interface Γ.

Due to the discontinuity of α occurring on the interface Γ, the solution to above initial and boundary value problem (IBVP) (1)–(2) shall be piece-wisely defined as $u = u^-(\vec{x}, t)$ for $\vec{x} \in \Omega^-$ and $u = u^+(\vec{x}, t)$ for $\vec{x} \in \Omega^+$. Moreover, the unknown function u on the two sides of the interface Γ is related by additional *jump conditions*

$$[u] := u^+ - u^- = \phi(\vec{x}, t), \quad [\alpha u_n] := \alpha^+ \nabla u^+ \cdot \vec{n} - \alpha^- \nabla u^- \cdot \vec{n} = \psi(\vec{x}, t), \tag{3}$$

which are defined on the interface Γ by taking the limiting values from both sides of the interface. Here functions ϕ and ψ are two known functions, and $\vec{n}$ is the unit outer normal direction.

Eqs. (1)–(3) constitute a mathematical model known as the *parabolic interface problem*, which is widely adopted in multiple disciplines for studying interested physical quantities, such as heat or electrostatic potentials, propagating across the material interface Γ from one side to the other. In this work, the interface Γ is considered to be fixed with respect to time, while the spatial jumps ϕ and ψ could be time dependent.

Finding the solutions to the parabolic interface problems is nontrivial. Exact solutions in closed forms are only available in the simplest cases, for instance, when $\phi = \psi = 0$ and the interface Γ is simple and regularly shaped. In practice, solutions have to be numerically approximated for complicated interfaces. However, standard numerical procedures, which usually seek for smooth solutions to Eq. (1) over the whole domain Ω, cannot deliver accurate approximations, and often fail to converge when the interface jumps (3) are significant. It calls for specifically designed numerical procedures which can explicitly incorporate the jump conditions (3) into the discretization of Eq. (1).

In the past a few decades, a number of numerical methods have been developed to solve parabolic and other interface problems. In general, methods for solving interface problems can be roughly classified into two categories: (i) *Body-fitted interface methods* in which both finite element and finite volume methods [1–5] have been developed to employ unstructured grids to fit the material interfaces in order to provide the best flexibility for handling complex geometries; (ii) *Cartesian grid interface methods* in which sophisticated interface algorithms are indispensable to accommodate the complex-shaped interfaces and jump conditions. One of the most successful methods is the immersed interface method (IIM) [6], which manages to achieve the second order of accuracy by imposing jump conditions rigorously in the finite difference formulations via local Taylor expansions. The IIM has achieved a tremendous success in solving parabolic interface problems [7–11].

Recently, there has been a growing interest in developing interface algorithms in the framework of Alternating Direction Implicit (ADI) methods [12,13]. The most attractive feature of ADI methods is their efficiency. As implicit schemes, ADI methods typically allow using a large time step size for long time simulations. Moreover, in each time step of ADI methods, a multidimensional system will be reduced to independent one-dimensional (1D) sub-systems of tridiagonal structures, and such matrices can be efficiently solved using the Thomas algorithm [14]. From the algebra point of view, the efficiency of ADI methods over usual iterative methods in matrix inversion can be justified by the fact that ADI methods yield exact algebraic solvers which guarantee to stop within a fixed number of steps. Typically, for a large system with a total degree of freedom N, the complexity of ADI methods is of the order N, i.e., $O(N)$. Therefore, in solving parabolic interface problems, it is desired to cope with interfaces in ADI methods, without affecting their efficiency.

In 1993, Li and Mayo [15] introduced a rigorous IIM-ADI scheme for parabolic interface problems with simplified jump conditions. In this case, the accuracy of the central differences near interfaces can be recovered to second order via adding some correction terms in the ADI formulation. Later, this IIM-ADI scheme has been applied to solve other parabolic interface problems [16–18]. For the general jump conditions given in (3), the first ADI method that deliveries second order spatial accuracy while maintaining $O(N)$ efficiency is the matched ADI (mADI) introduced by Zhao in 2015 [19]. In the mADI, a tensor product decomposition is carried out to reduce jump conditions into 1D ones along Cartesian directions. Then these 1D conditions are enforced to secure a second order of convergence. Such an interface treatment is generalized from the matched interface and boundary (MIB) method [20,21] for general interface problems. Moreover, a fast algebraic algorithm combining the Gaussian Elimination procedure and the Thomas algorithm was introduced for solving perturbed tridiagonal linear systems so that the overall complexity is maintained as $O(N)$. The mADI method has been further generalized to the Peaceman–Rachford formulation [22] and for solving three-dimensional (3D) parabolic interface problems [23]. More recently, a new IIM-ADI method has been developed based on an augmented approach [24]. By introducing new auxiliary variables on the interface, the addition of correction terms is able to handle general jump conditions. The resulting ADI scheme is of second order accuracy in both space and time discretization.

It is noted that the interface treatments in the ADI methods are related to general interface algorithms, but have subtle differences. Since the jump conditions (3) are prescribed in the normal direction, which is not along Cartesian directions, the usual interface treatments naturally couple spatial derivatives in all Cartesian directions [25,26]. Nevertheless, to be applicable in the ADI formulation, interfaces have to be dealt with in a dimension-by-dimension manner. It is crucial that discretization in one alternating direction should not be simultaneously coupled with other Cartesian directions. Two major types of strategies have been developed in the literature for this purpose. In the IIM-ADI methods [15–18,24],

the jump conditions are satisfied multi-dimensionally through introducing corrections to standard central difference discretization of the Laplacian operator. Such a discrete Laplacian operator allows a direct dimensional splitting, giving rise to a new type of ADI methods. A different approach is employed in the mADI methods [19,22,23], in which the jump conditions are decomposed in a tensor product manner to each Cartesian direction. This requires jump data in tangential directions at the future time step, say t^{n+1}. This process is essentially equivalent to the general MIB interface algorithm [20,21]. However, in order to avoid unwanted coupling, the tangential jumps are estimated by using function values at the current time step t^n in the mADI methods. The numerically generated Cartesian jump conditions can then be discretized in a single direction, so that the standard ADI methods [12,13] can be corrected.

In this work, a new ADI method based on the Ghost Fluid Method (GFM) will be developed. The GFM was first proposed by Fedkiw et al. in 1999 for treating interfaces in multi-material flows [27], and was quickly adopted to solve variable coefficient Poisson equations, as well as the viscous Navier–Stokes equations [28,29], in the presence of interfaces where both variable coefficients and solutions may be discontinuous [26,30–32]. The GFM, together with its variations, such as the gas–water version GFM (GWGFM), the modified GFM (MGFM) and the real GFM (RGFM) [33–36], soon was widely used to solve the models for simulating the impact of strong shock on material interfaces [34,37,38], interface deflagration and detonation [39], liquid jet [40], compressible/incompressible multimedia flows [33,36,41–54], multiphase electrohydrodynamic (EHD) flows for a high electric Reynolds number regime [55], shallow water and ideal magnetohydrodynamics [56,57], fluid–structure interaction [58–60], and so on. Analysis of the accuracy and conservation errors of various ghost fluid type methods can be found in [61–63] by Liu et al. Recently, a second order GFM was developed for the Poisson equation through appropriately using of auxiliary virtual points [26]. However, to our best knowledge, the GFM has not been applied to solve the heat equation with interfaces.

Compared with more accurate interface algorithms, such as the IIM [6], the MIB [20,21], the Coupling Interface Method (CIM) [25], and the second order GFM [26], the standard GFM [30] employs a very simple procedure for enforcing jump conditions, albeit it only attains a first order of convergence. By completely neglecting tangential jumps, the normal jumps are simply projected to Cartesian directions in the GFM, so that the resulting finite difference matrix maintains symmetry. Such an algebraic property is not shared by other interface algorithms, and is highly desired for time stability when solving parabolic PDEs. This is one major motivation for developing a GFM-ADI method in this work. Moreover, the GFM and MIB methods share a great deal of commons to enforce the jump conditions (3) in the numerical procedure. More profound investigations revealed that the mADI method [19], as well as one simple variation, can be viewed as modifications/improvements of the GFM.

In particular, three ADI methods will be investigated in this paper. First, a new GFM-ADI method will be proposed. Moreover, a new mADI method will be developed, which is constructed by simply replacing the quadratic polynomial basis in the mADI method [19] by linear ones. This new method will be termed as MIBV1-ADI in this work, which has the same finite difference matrix as that of the GFM-ADI, except that ignored tangential jumps at interface points in the GFM now are recovered, resulting in more accurately approximated jumps enforced in the MIB-type methods. Finally, the original mADI method will be called the MIBV2-ADI scheme in this work, emphasizing its second order accuracy. More details about these three methods and their comparison will be provided in the following sections.

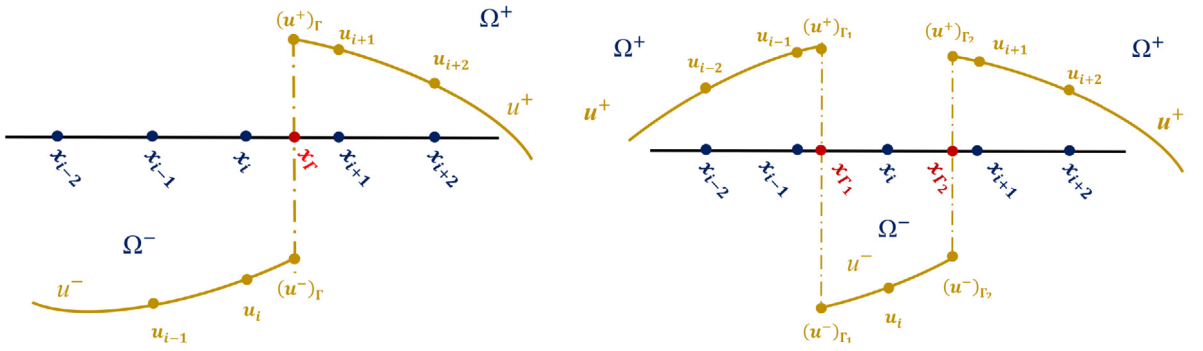
The rest of this work is organized as follows. In Section 2, the GFM and MIB methods will be demonstrated in one dimension (1D) for the seek of simplicity. In 1D, the interface Γ becomes a point and the normal direction $\overrightarrow{n}$ coincides the x-axis. In this simplest case, one will see that GFM and MIBV1 methods produce the same formulas to impose the jump conditions (3) via completely different numerical treatments, while the MIBV2 yields a higher order of accuracy. In Section 3, two-dimensional (2D) GFM-ADI and MIBV1-ADI methods are established. In 2D, the normal direction at an interface point, in general, does not coincide a grid line, resulting in different numerical formulas in these two methods. In order to quantitatively compare the performance of the new methods, as well as the original mADI or the MIBV2-ADI method, numerical experiments are provided in this section as well. In Section 4, all methods are extended once again via a technique previously reported in [23] to solve three dimensional (3D) interface problems. Concluding remarks are then provided in Section 5.

2. The GFM and MIB-type methods in one dimension

In one dimension (1D), domain $\Omega = [a, b]$ is a finite closed interval on the x-axis. This interval is then discretized by a sequence of equally spaced grids, $a = x_0 < x_1 < \cdots < x_n = b$, with uniform mesh size h. Point interface Γ is denoted by x_Γ and positioned between x_i and x_{i+1} for some i between 0 and n. The normal direction n coincides either the positive or negative direction of the x-axis.

The GFM and a MIB-type method are demonstrated here by considering the numerical treatments for approximating the term $\nabla \cdot (\alpha \nabla u)$ in Eq. (1). In 1D, $\nabla \cdot (\alpha \nabla u) = (\alpha u_x)_x$ and we denote its finite difference approximation as $\delta_{xx} u$. We assume the spatial discretization is fine enough so that only two types of interface points can occur, as shown in Fig. 1. In Fig. 1a, one interface point x_Γ (red dot) positions in the subinterval (x_i, x_{i+1}) and no other interface points occur in adjacent subintervals. This interface point is named an *irregular (interface) point*. Without losing the generality, we denote $x_\Gamma - x_i = \eta h$ and $x_{i+1} - x_\Gamma = (1 - \eta)h$ for some η between 0 and 1, and assume $u(x) = u^-(x)$ on the left side of x_Γ ($x \in \Omega^-$), while $u(x) = u^+(x)$ on the other side ($x \in \Omega^+$). At x_Γ, jump conditions,

$$[u]_\Gamma = (u^+)_\Gamma - (u^-)_\Gamma = \phi_\Gamma, \quad [\alpha u_x]_\Gamma = \alpha^+ \left(u_x^+\right)_\Gamma - \alpha^- \left(u_x^-\right)_\Gamma = \psi_\Gamma, \tag{4}$$



(a) an irregular interface point (b) a pair of corners

Fig. 1. Graphical demonstration of one dimensional jump conditions enforcement in the GFM.

are given with $\phi_\Gamma = \phi(x_\Gamma)$ and $\psi_\Gamma = \psi(x_\Gamma)$ being two (possibly nonzero) constants, and the outer normal direction n coincides the negative direction of the x-axis, pointing from subdomain Ω^+ to subdomain Ω^-.

The second type of interface points consists of a pair of interface points, demonstrated by x_{Γ_1} and x_{Γ_2} in Fig. 1b, positioning in two adjacent subintervals. Both interface points in this case are named *corners*, and the jump conditions taking place at these two corners are given by

$$[u]_{\Gamma_1} = (u^+)_{\Gamma_1} - (u^-)_{\Gamma_1} = \phi_{\Gamma_1}, \quad [\alpha u_x]_{\Gamma_1} = \alpha^+ \left(u_x^+\right)_{\Gamma_1} - \alpha^- \left(u_x^-\right)_{\Gamma_1} = \psi_{\Gamma_1}, \tag{5}$$

$$[u]_{\Gamma_2} = (u^+)_{\Gamma_2} - (u^-)_{\Gamma_2} = \phi_{\Gamma_2}, \quad [\alpha u_x]_{\Gamma_2} = \alpha^+ \left(u_x^+\right)_{\Gamma_2} - \alpha^- \left(u_x^-\right)_{\Gamma_2} = \psi_{\Gamma_2}. \tag{6}$$

We consider to construct $\delta_{xx}u$ at the five grids shown in Fig. 1. First of all, it is easy to see that the three grids, $\{x_{i-2}, x_{i-1}, x_{i+2}\}$ in Fig. 1a, and the two grids, $\{x_{i-2}, x_{i+2}\}$ in Fig. 1b, are "*regular*" in the sense that $\delta_{xx}u$ can be constructed by the central difference formula

$$[(\alpha u_x)_x]_j = \alpha_j \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + O(h^2), \tag{7}$$

for $j = i-2, i-1, i+2$ in Fig. 1a and $j = i-2, i+2$ in Fig. 1b, respectively, since these grids, as well as their left and right adjacent grids, are all on the same side of point interface x_Γ. Here α_j could be either α^+ or α^-, depending on whether x_j is in subdomain Ω^- or Ω^+.

On the other hand, the discrete operator $\delta_{xx}u$ must be corrected at the two"*irregular*" grids, $\{x_i, x_{i+1}\}$ in Fig. 1a, and the three "*corner*" grids, $\{x_{i-1}, x_i, x_{i+1}\}$ in Fig. 1b, when at least one adjacent grid positions on the opposite side of the interface point(s). In this case, jump conditions (4)–(6) must be taken into account in order to deliver accurate approximations. The GFM and MIB-type methods enforce jump conditions to correct (7) at irregular and corner grids in different manners, but result in the same formulas after algebraic simplifications in 1D. Details will be provided in below.

2.1. Enforcing 1D jump conditions in the GFM

To see the connections between GFM and MIB methods, we derive the GFM presented in [30] in a slightly different way. At irregular grids $\{x_i, x_{i+1}\}$ (Fig. 1a), we consider approximating $\left(u_x^-\right)_\Gamma$ and $\left(u_x^+\right)_\Gamma$ by

$$\left(u_x^-\right)_\Gamma = \frac{\left(u^-\right)_\Gamma - u_i}{\eta h} + O(h), \quad \text{and} \quad \left(u_x^+\right)_\Gamma = \frac{u_{i+1} - \left(u^+\right)_\Gamma}{(1-\eta)h} + O(h). \tag{8}$$

Substituting approximations (8) into jump conditions (4) yields a system of equations

$$\left(u^+\right)_\Gamma - \left(u^-\right)_\Gamma = \phi_\Gamma, \tag{9}$$

$$\alpha^+ \left(\frac{u_{i+1} - \left(u^+\right)_\Gamma}{(1-\eta)h}\right) - \alpha^- \left(\frac{\left(u^-\right)_\Gamma - u_i}{\eta h}\right) = \psi_\Gamma + O(h), \tag{10}$$

from which limit values $\left(u^-\right)_\Gamma$ and $\left(u^+\right)_\Gamma$ at x_Γ can be solved for as

$$\left(u^-\right)_\Gamma = \frac{\alpha^-(1-\eta)u_i + \alpha^+\eta(u_{i+1} - \phi_\Gamma) - \eta(1-\eta)h\psi_\Gamma}{\alpha^+\eta + \alpha^-(1-\eta)} + O(h^2), \tag{11}$$

$$\left(u^+\right)_\Gamma = \frac{\alpha^-(1-\eta)(u_i + \phi_\Gamma) + \alpha^+\eta u_{i+1} - \eta(1-\eta)h\psi_\Gamma}{\alpha^+\eta + \alpha^-(1-\eta)} + O(h^2). \tag{12}$$

This yields representations of $\left(u^-\right)_\Gamma$ and $\left(u^+\right)_\Gamma$ in terms of $\{u_i, u_{i+1}, \phi_\Gamma, \psi_\Gamma\}$.

At x_i in Ω^-, the GFM in 1D could be derived by using finite difference formula

$$\begin{aligned}[(\alpha u_x)_x]_i &= \frac{1}{h}\left(\alpha_{i+\frac{1}{2}}\ (u_x)_{i+\frac{1}{2}} - \alpha_{i-\frac{1}{2}}\ (u_x)_{i-\frac{1}{2}}\right) + O(h^2) \\ &= \frac{1}{h}\left(\alpha^-\ \left(u_x^-\right)_\Gamma - \alpha^-\ (u_x)_{i-\frac{1}{2}}\right) + O(h) \\ &= \frac{1}{h}\left(\alpha^-\left(\frac{\left(u^-\right)_\Gamma - u_i}{\eta h}\right) - \alpha^-\left(\frac{u_i - u_{i-1}}{h}\right)\right) + O(1).\end{aligned} \tag{13}$$

By substituting (11) into (13), one attains the GFM scheme at x_i

$$[(\alpha u_x)_x]_i = \frac{1}{h}\left(\hat{\alpha}\left(\frac{(u_{i+1} - \phi_\Gamma) - u_i}{h} - \frac{\psi_\Gamma(1-\eta)}{\alpha^+}\right) - \alpha^-\left(\frac{u_i - u_{i-1}}{h}\right)\right) + O(1), \tag{14}$$

where $\hat{\alpha} = \left(\alpha^+\alpha^-\right)\Big/\left(\alpha^+\eta + \alpha^-(1-\eta)\right)$. Similarly, at x_{i+1} in Ω^+, the GFM scheme is

$$[(\alpha u_x)_x]_{i+1} = \frac{1}{h}\left(\alpha^+\left(\frac{u_{i+2} - u_{i+1}}{h}\right) - \hat{\alpha}\left(\frac{u_{i+1} - (u_i + \phi_\Gamma)}{h} + \frac{\psi_\Gamma\ \eta}{\alpha^-}\right)\right) + O(1). \tag{15}$$

By correcting the discrete operator $\delta_{xx}u$ according to (14) and (15) at x_i and x_{i+1}, respectively, the GFM yields a finite difference approximation that has an $O(1)$ local truncation error. By using the second order central difference for other points, the global error of the GFM will be $O(h)$ [25]. We note that the discrete matrix of δ_{xx} is symmetric and tridiagonal. The resulting linear system of equations can then be solved by fast methods, such as the Thomas algorithm [14], for the values of u at all grids.

Correcting $\delta_{xx}u$ at the three corner grids $\{x_{i-1}, x_i, x_{i+1}\}$ (Fig. 1b) is neglected in [30]. However, it is not hard to correct $\delta_{xx}u$ at corner grids following the same fashion. Here we provide a straightforward formulation to allow the GFM enforce jump conditions at corner grids.

Following the aforementioned discussions, one can see that correcting $\delta_{xx}u$ at the left corner grid x_{i-1} and right corner grid x_{i+1} can be achieved by using jump conditions (5) at the left corner x_{Γ_1} and jump conditions (6) at the right corner x_{Γ_2}, respectively. Moreover, jump condition $[(\alpha u_x)_x]_i$ at the middle corner grid x_i can be approximated by

$$[(\alpha u_x)_x]_i = \frac{\alpha^-}{h}\left(\frac{u_i - u_{\Gamma_1}^-}{(1-\eta_1)h} - \frac{u_{\Gamma_2}^- - u_i}{\eta_2 h}\right) + O(1), \tag{16}$$

where $x_i - x_{\Gamma_1} = (1-\eta_1)h$ and $x_{\Gamma_2} - x_i = \eta_2 h$ for some η_1 and η_2 between 0 and 1. Assuming $\left(u^-\right)_{\Gamma_1}$ is represented in terms of $\{u_{i-1}, u_i, \phi_{\Gamma_1}, \psi_{\Gamma_1}\}$, and $\left(u^-\right)_{\Gamma_2}$ is represented in terms of $\{u_i, u_{i+1}, \phi_{\Gamma_2}, \psi_{\Gamma_2}\}$, formula (16) corrects $\delta_{xx}u$ at x_i and can be rewritten in terms of $\{u_{i-1}, u_i, u_{i+1}, \phi_{\Gamma_1}, \psi_{\Gamma_1}, \phi_{\Gamma_2}, \psi_{\Gamma_2}\}$, in which jump conditions at both corners are involved. Interestingly, the symmetric and tridiagonal matrix structures are still preserved, while the global approximation error is still $O(h)$.

2.2. Enforcing 1D jump conditions in the MIB method

MIB-type methods take a difference approach to enforce jump conditions in the finite difference formulation. More specifically, additional "*fictitious values*" at irregular and corner grids are introduced in order to correct $\delta_{xx}u$ at these grids. For the seek of simplicity, a simple MIB-type method, named the MIBV1, is introduced in detail to elaborate the formulations. It is interesting to point out that the MIBV1 results in the same finite difference formulas as those obtained by the GFM after algebraic simplifications.

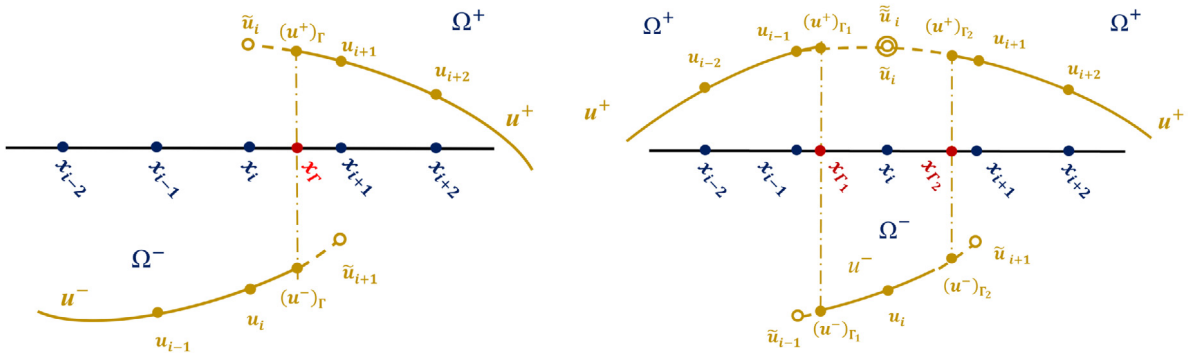
Fig. 2 considers the same 1D jumps taking place at the irregular interface point x_Γ and corners x_{Γ_1} and x_{Γ_2} as those shown in Fig. 1. In the case of irregular point (Fig. 2a), two fictitious values, $\tilde{u}_i$ and $\tilde{u}_{i+1}$, are imposed at the irregular grids x_i and x_{i+1}, respectively. One can view that the two fictitious values are values of unknown function u extended from the opposite side of the interface point x_Γ. With the newly introduced fictitious values, limit values $\left(u^+\right)_\Gamma$ and $\left(u^-\right)_\Gamma$ can then be approximated by

$$\left(u^-\right)_\Gamma = (1-\eta)u_i + \eta\tilde{u}_{i+1} + O(h^2), \quad \left(u^+\right)_\Gamma = (1-\eta)\tilde{u}_i + \eta u_{i+1} + O(h^2). \tag{17}$$

Substituting approximations (17) in jump conditions (4) yields a system of equations

$$\left((1-\eta)\tilde{u}_i + \eta u_{i+1}\right) - \left((1-\eta)u_i + \eta\tilde{u}_{i+1}\right) = \phi_\Gamma + O(h^2), \tag{18}$$

$$\alpha^+\left(\frac{u_{i+1} - \tilde{u}_i}{h}\right) - \alpha^-\left(\frac{\tilde{u}_{i+1} - u_i}{h}\right) = \psi_\Gamma + O(h), \tag{19}$$



(a) one irregular interface point (b) a pair of corners

Fig. 2. Graphical demonstration of one dimensional jump conditions enforcement in the MIBV1.

from which the two fictitious values can be solved for in terms of $\{u_i, u_{i+1}, \phi_\Gamma, \psi_\Gamma\}$ as

$$\tilde{u}_i = \frac{\hat{\alpha}}{\alpha^+}u_i + \eta\hat{\alpha}\left(\frac{1}{\alpha^-} - \frac{1}{\alpha^+}\right)u_{i+1} + \frac{\hat{\alpha}}{\alpha^+}\phi_\Gamma - \eta h\frac{\hat{\alpha}}{\alpha^+\alpha^-}\psi_\Gamma + O(h^2), \tag{20}$$

$$\tilde{u}_{i+1} = -(1-\eta)\hat{\alpha}\left(\frac{1}{\alpha^+} + \frac{1}{\alpha^-}\right)u_i + \frac{\hat{\alpha}}{\alpha^-}u_{i+1} - \frac{\hat{\alpha}}{\alpha^-}\phi_\Gamma - h(1-\eta)\frac{\hat{\alpha}}{\alpha^+\alpha^-}\psi_\Gamma + O(h^2). \tag{21}$$

With above fictitious values, central difference approximations can be obtained as

$$[(\alpha u_x)_x]_i = \alpha^-\left(\frac{u_{i-1} - 2u_i + \tilde{u}_{i+1}}{h^2}\right) + O(h^2), \tag{22}$$

$$[(\alpha u_x)_x]_{i+1} = \alpha^+\left(\frac{\tilde{u}_i - 2u_{i+1} + u_{i+2}}{h^2}\right) + O(h^2). \tag{23}$$

Substituting (20)–(21) in (22)–(23) yields the same correction formulas (14)–(15) previously obtained for the GFM.

Similarly, correcting $\delta_{xx}u$ at the three corner grids (Fig. 2b) using fictitious values can be obtained by introducing, in total, four fictitious values: one ($\tilde{u}_{i-1}$) at the left corner grid x_{i-1}, two ($\tilde{u}_i$ and $\tilde{\tilde{u}}_i$) at the middle corner grid x_i, and one ($\tilde{u}_{i+1}$) at the right corner grid. Substituting the four fictitious values in the jump conditions (5)–(6) yields a system of four equations

$$\left((1-\eta_1)u_{i-1} + \eta_1\tilde{u}_i\right) - \left((1-\eta_1)\tilde{u}_{i-1} + \eta_1 u_i\right) = \phi_{\Gamma_1} + O(h^2), \tag{24}$$

$$\alpha^+\left(\frac{u_i - \tilde{u}_{i-1}}{h}\right) - \alpha^-\left(\frac{\tilde{u}_i - u_{i-1}}{h}\right) = \psi_{\Gamma_1} + O(h), \tag{25}$$

$$\left((1-\eta_2)\tilde{\tilde{u}}_i + \eta_2 u_{i+1}\right) - \left((1-\eta_1)u_i + \eta_2\tilde{u}_{i+1}\right) = \phi_{\Gamma_2} + O(h^2), \tag{26}$$

$$\alpha^+\left(\frac{u_{i+1} - \tilde{\tilde{u}}_i}{h}\right) - \alpha^-\left(\frac{\tilde{u}_{i+1} - u_i}{h}\right) = \psi_{\Gamma_2} + O(h), \tag{27}$$

from which the four fictitious values can be solved for in terms of $\{u_{i-1}, u_i, u_{i+1}, \phi_{\Gamma_1}, \psi_{\Gamma_1}, \phi_{\Gamma_2}, \psi_{\Gamma_2}\}$. These representations can then be used in

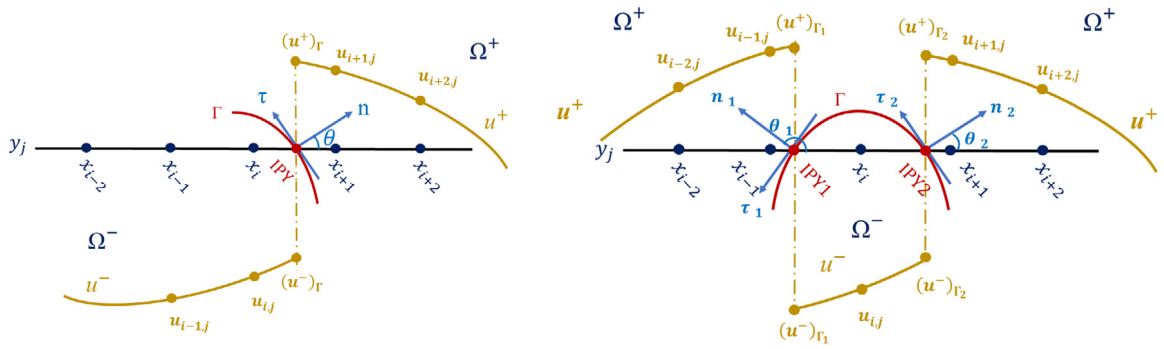
$$[(\alpha u_x)_x]_{i-1} = \alpha^+\left(\frac{u_{i-2} - 2u_{i-1} + \tilde{u}_i}{h^2}\right) + O(h^2), \tag{28}$$

$$[(\alpha u_x)_x]_i = \alpha^-\left(\frac{\tilde{u}_{i-1} - 2u_i + \tilde{u}_{i+1}}{h^2}\right) + O(h^2), \tag{29}$$

$$[(\alpha u_x)_x]_{i+1} = \alpha^+\left(\frac{\tilde{\tilde{u}}_i - 2u_{i+1} + u_{i+2}}{h^2}\right) + O(h^2). \tag{30}$$

to achieve the corrected formulas of $\delta_{xx}u$ at the three corner grids.

It is interesting to point out that the GFM and MIBV1 actually result in identical numerical formulas, even though their approaches to enforce the jump conditions are different. Therefore, one dimensional numerical examples comparing GFM and MIBV1 are omitted in this work. Interested readers are directed to [30] for 1D numerical examples.



(a) one irregular interface point (b) a pair of corners

Fig. 3. Graphical demonstration of two dimensional jump conditions enforcement in the GFM.

Identical formulas resulted in the GFM and MIBV1 methods are essentially due to the fact that linear polynomials are employed in jump condition enforcement based on either fictitious values at nodes or jump values on the interface. In the MIB literature, quadratic polynomials are usually used to ensure a local truncation error $O(h)$ and a global error $O(h^2)$ [19,22,23]. Such a MIB method in 1D will be referred to as the MIBV2 scheme in this paper. We briefly discuss the idea of the MIBV2 without offering much detail. Taking $(u^-)_\Gamma$ in Eq. (17) as an example, it will be approximated by a quadratic polynomial, or in terms of three values, $\{u_{i-1}, u_i, \tilde{u}_{i+1}\}$. The truncation error will be $O(h^3)$. In this manner, two fictitious values $\tilde{u}_i$ and $\tilde{u}_{i+1}$ will be solved in terms of $\{u_{i-1}, u_i, u_{i+1}, u_{i+2}, \phi_\Gamma, \psi_\Gamma\}$ with $O(h^3)$ error. Substituting such fictitious values in (22)–(23) yields the MIBV2 scheme, which improves the local error from $O(1)$ to $O(h)$. This leads to $O(h^2)$ global error in the maximal norm [25]. The treatment of corner points has been discussed in the mADI method [19]. We note that the order improvement in the MIBV2 breaks the symmetric and tridiagonal matrix structure. Fortunately, a modified Thomas algorithm was proposed in [19] to solve the system efficiently.

Finally, we note another point that contributes to the identical formulas achieved by the GFM and MIBV1 in 1D. This is due to the fact that the normal direction at the point interface coincides the x-axis. However, it is not the case in higher dimensions when the normal directions at interface points, in general, do not coincide grid lines. As a result, the formulas obtained by the GFM and MIBV1 become different in higher dimensions.

3. Solving parabolic interface problems in two dimensions by the GFM and MIB-type methods

The GFM and MIBV1 methods will be different in higher dimensions. We will elaborate these differences in two dimensions (2D) in this section. To this end, 2D uniform Cartesian grid meshes are deployed in both x and y directions over a finite two dimensional rectangular domain Ω. For simplicity, we assume $h = \Delta x = \Delta y$ in the following discussions.

3.1. GFM and MIB-type methods in two dimensions

Following the same numerical setups and notations in 1D, the GFM in 2D is graphically demonstrated in Fig. 3, in which both irregular and corner cases are demonstrated by a piece of a 2D closed interface Γ (red) cutting one grid line y_j. In Fig. 3a, interface Γ cuts grid line y_j at an irregular interface point IPY, where the normal vector n forms a nonzero angle θ with the positive direction of the x-grid line $y = y_j$. A coordinate transformation formula

$$\frac{\partial}{\partial n} = \cos\theta \frac{\partial}{\partial x} + \sin\theta \frac{\partial}{\partial y}, \quad \frac{\partial}{\partial \tau} = -\sin\theta \frac{\partial}{\partial x} + \cos\theta \frac{\partial}{\partial y}, \tag{31}$$

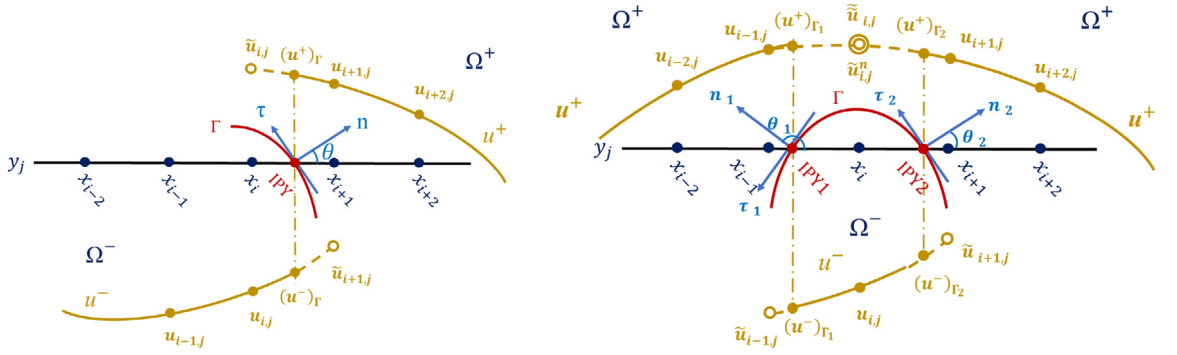
is utilized in both the GFM and MIBV1 to convert the jumps at IPY in the normal n and tangential τ directions to those in the x- and y- directions

$$[\alpha u_n]_\Gamma = \cos\theta\,[\alpha u_x]_\Gamma + \sin\theta\left[\alpha u_y\right]_\Gamma, \quad [\alpha u_\tau]_\Gamma = -\sin\theta\,[\alpha u_x]_\Gamma + \cos\theta\left[\alpha u_y\right]_\Gamma. \tag{32}$$

For computational simplicity, the GFM [30] assumes that no jump occurs in the tangential direction, i.e., $[\alpha u_\tau]_\Gamma = 0$, so that the jumps at IPY in the x-direction are obtained by

$$[u]_\Gamma = \left(u^+\right)_\Gamma - \left(u^-\right)_\Gamma = \phi_\Gamma, \quad [\alpha u_x]_\Gamma = \alpha^+(u_x^+)_\Gamma - \alpha^-(u_x^-)_\Gamma \approx \cos\theta\,[\alpha u_n] = \cos\theta\ \psi_\Gamma. \tag{33}$$

After jump conditions (33) are obtained, one can then follow the same steps (8)–(15) in one dimension to correct $\delta_{xx}u$ at the two irregular grids $\{(x_i, y_j), (x_{i+1}, y_j)\}$.



(a) one irregular interface point (b) a pair of corners

Fig. 4. Graphical demonstration of two dimensional jump conditions enforcement in the MIBV1.

Similar treatments applied to the corners in Fig. 3b lead to jump conditions at two corners as

$$[u]_{\Gamma_1} = (u^+)_{\Gamma_1} - (u^-)_{\Gamma_1} = \phi_{\Gamma_1}, \quad [\alpha u_x]_{\Gamma_1} = \alpha^+ \left(u_x^+\right)_{\Gamma_1} - \alpha^- \left(u_x^-\right)_{\Gamma_1} \approx \cos\theta_1 \psi_{\Gamma_1}, \tag{34}$$

$$[u]_{\Gamma_2} = (u^+)_{\Gamma_2} - (u^-)_{\Gamma_2} = \phi_{\Gamma_2}, \quad [\alpha u_x]_{\Gamma_2} = \alpha^+ \left(u_x^+\right)_{\Gamma_2} - \alpha^- \left(u_x^-\right)_{\Gamma_2} \approx \cos\theta_2 \psi_{\Gamma_2}, \tag{35}$$

which are used to correct $\delta_{xx}u$ at the three corner grids $\{(x_{i-1}, y_j), (x_i, y_j), (x_{i+1}, y_j)\}$ in Fig. 3b.

GFM's assumption that no jumps occurring in the tangential direction is merely a numerical simplification. More rigorous formulations [19,22] indicate that the exact jump $[\alpha u_x]$ at the irregular interface point (x_Γ, y_j) is

$$[\alpha u_x]_\Gamma = \cos\theta\ \psi_\Gamma - \sin\theta\,[\alpha u_\tau]_\Gamma \tag{36}$$

$$= \cos\theta\ \psi_\Gamma - \sin\theta(\alpha^+ - \alpha^-)\left(u_\tau^+\right)_\Gamma - \sin\theta\alpha^-\,(\phi_\tau)_\Gamma \tag{37}$$

$$= \cos\theta\ \psi_\Gamma - \sin\theta(\alpha^+ - \alpha^-)\left(u_\tau^-\right)_\Gamma - \sin\theta\alpha^+\,(\phi_\tau)_\Gamma\,, \tag{38}$$

where the subscript "τ" denotes the derivative along the positive tangential direction, and the superscript, "+" or "−", indicates the derivative is taken in the respective subdomain, Ω^+ or Ω^-.

Comparing (36)–(38) to (33), the ignored tangential jumps in the GFM are clearly revealed. The MIB-type methods adopt (36)–(38) to estimate $[\alpha u_x]_\Gamma$ for better accuracy [19]. Notice that when (37) and (38) are used, either $u_\tau^+ = \frac{\partial u^+}{\partial \tau}$ or $u_\tau^- = \frac{\partial u^-}{\partial \tau}$ shall be numerically approximated, while $\phi_\tau = \frac{\partial \phi}{\partial \tau}$ could be analytically calculated from the known ϕ function. See [19,22] for the methods to estimate $\left(u_\tau^+\right)_\Gamma$ and $\left(u_\tau^-\right)_\Gamma$. Let $\bar{\psi}_\Gamma \approx [\alpha u_x]_\Gamma$ be an approximation at IPY, a system of equations

$$\left((1-\eta)\tilde{u}_{i,j} + \eta u_{i+1,j}\right) - \left((1-\eta)u_{i,j} + \eta\tilde{u}_{i+1,j}\right) = \phi_\Gamma + O(h^2), \tag{39}$$

$$\alpha^+\left(\frac{u_{i+1,j} - \tilde{u}_{i,j}}{h}\right) - \alpha^-\left(\frac{\tilde{u}_{i+1,j} - u_{i,j}}{h}\right) = \bar{\psi}_\Gamma + O(h), \tag{40}$$

is then established in the MIBV1 method for the two fictitious values $\tilde{u}_{i,j}$ and $\tilde{u}_{i+1,j}$.

Similar treatments lead to a system of four equations

$$\left((1-\eta_1)u_{i-1,j} + \eta_1\tilde{u}_{i,j}\right) - \left((1-\eta_1)\tilde{u}_{i-1,j} + \eta_1 u_{i,j}\right) = \phi_{\Gamma_1} + O(h^2), \tag{41}$$

$$\alpha^+\left(\frac{u_{i,j} - \tilde{u}_{i-1,j}}{h}\right) - \alpha^-\left(\frac{\tilde{u}_{i,j} - u_{i-1,j}}{h}\right) = \bar{\psi}_{\Gamma_1} + O(h), \tag{42}$$

$$\left((1-\eta_2)\tilde{\tilde{u}}_{i,j} + \eta_2 u_{i+1,j}\right) - \left((1-\eta_1)u_{i,j} + \eta_1\tilde{u}_{i+1,j}\right) = \phi_{\Gamma_2} + O(h^2), \tag{43}$$

$$\alpha^+\left(\frac{u_{i+1,j} - \tilde{\tilde{u}}_{i,j}}{h}\right) - \alpha^-\left(\frac{\tilde{u}_{i+1,j} - u_{i,j}}{h}\right) = \bar{\psi}_{\Gamma_2} + O(h), \tag{44}$$

at corners shown in Fig. 4b. Here $\bar{\psi}_{\Gamma_1} \approx [\alpha u_x]_{\Gamma_1}$ and $\bar{\psi}_{\Gamma_2} \approx [\alpha u_x]_{\Gamma_2}$ at the two corners, respectively. Solving the system (41)–(44) results in four fictitious values $\tilde{u}_{i-1,j}$, $\tilde{u}_{i,j}$, $\tilde{\tilde{u}}_{i,j}$, and $\tilde{u}_{i+1,j}$ written as the linear combinations of $\{u_{i-1,j}, u_{i,j}, u_{i+1,j}, \phi_{\Gamma_1}, \bar{\psi}_{\Gamma_1}, \phi_{\Gamma_2}, \bar{\psi}_{\Gamma_2}\}$.

The main difference between the GFM and MIBV1 in 2D lies in the approximations of $[\alpha u_x]_\Gamma$ and $[\alpha u_y]_\Gamma$ at interface points. Since such approximations only affect the right hand side values in the discretized jump conditions, the finite

difference matrix structures of the GFM and MIBV1 in 2D are still the same. Moreover, as in 1D, the global truncation error of such finite difference formula is also $O(h)$. In other words, the GFM and MIBV1 are first order accurate finite difference methods in 2D. However, the impact of jump approximations to the GFM and MIBV1 methods is different. For the GFM, the approximation error underlying $[\alpha u_x]_\Gamma \approx \cos\theta \psi_\Gamma$ is $O(1)$, which could be large or small depending on actual solution. Since the MIBV1 recovers the jumps along the tangential direction and uses them for correcting $\delta_{xx}u$ at irregular and corner grids, it is natural to expect that the MIBV1 is more accurate when compared to the GFM. In particular, for the present parabolic problem, we will follow the scheme proposed in the mADI method [19,22] to estimate tangential jumps at the next future time step by using function values at the present time step. Assuming time increment is Δt, we have $[\alpha u_x]_\Gamma = \bar{\psi}_\gamma + O(h^2 + \Delta t)$ for the MIBV1 method. We thereby consider the MIBV1 an improvement of the standard GFM [30] in terms of accuracy.

3.2. Time discretization

When coupled with appropriate time evolution methods, both GFM and MIBV1 can be used to form fully-discretized methods for solving parabolic interface problems (1)–(3). In this work, two implicit time evolution methods, the implicit Euler (IE) and Douglas ADI (ADI) methods, are selected so that the equilibrium solutions of long-term simulations can be obtained within a reasonable time frame. Both time evolution methods are known to converge in first order and similarly accurate so that we expect they are going to deliver close numerical solutions. However, our previous experiments indicate that the ADI is more efficient when comparing to the IE. We will demonstrate their efficiency by comparing the CPU time when solving the same interface problem using the IE and ADI methods.

In two dimensions, solving Eq. (1) by the IE method with a uniform time step Δt yields

$$(1 - \Delta t(\delta_{xx} + \delta_{yy}))u_{i,j}^{n+1} = u_{i,j}^{n} + \Delta t f_{i,j}^{n+1}, \tag{45}$$

and solving Eq. (1) by the ADI method with the same time step Δt yields

$$(1 - \Delta t\delta_{xx})u_{i,j}^{*} = u_{i,j}^{n} + \Delta t\delta_{yy}u_{i,j}^{n} + \Delta t f_{i,j}^{n+1}, \tag{46}$$

$$(1 - \Delta t\delta_{yy})u_{i,j}^{n+1} = u_{i,j}^{*} - \Delta t\delta_{xx}u_{i,j}^{n}. \tag{47}$$

In both time evolution schemes, the finite difference operators δ_{xx} and δ_{yy} have been corrected by either the GFM or the MIBV1 at irregular and corners grids. The resulting fully discretized methods are then named as GFM-IE, GFM-ADI, MIBV1-ADI, and so on. We note that the MIBV2-ADI is actually the mADI method reported in [19]. We are interested in comparing it with newly proposed methods in this work.

It shall be pointed out that the methods involving the GFM are purely implicit by ignoring the tangential derivatives at the interface points, while those involving the MIBV1 and MIBV2 are actually semi-implicit due to approximating the tangential derivatives at the next time step t^{n+1} with its value at current time step t^n in every time evolution step [19]. This numerical treatment obviously affects the stability and accuracy of the proposed methods. Its impact on the numerical solutions will be studied using various 2D examples in the next subsection.

3.3. Two dimensional numerical experiments

In this subsection, two dimensional (2D) numerical experiments will be conducted to earn profound insights on the proposed methods for solving parabolic interface problems. To this end, an exact solution is defined

$$u(x, y, t) = \begin{cases} \sin(2x)\cos(2y)\cos(t) & \text{in } \Omega^- \\ \cos(2x)\sin(2y)\cos(t) & \text{in } \Omega^+ \end{cases} \tag{48}$$

over a finite spatial domain $\Omega = [-0.99, 0.99] \times [-0.99, 0.99]$ from an initial time $t = 0$ to a final time $t = 1$. The interface Γ is constructed by a parametric function

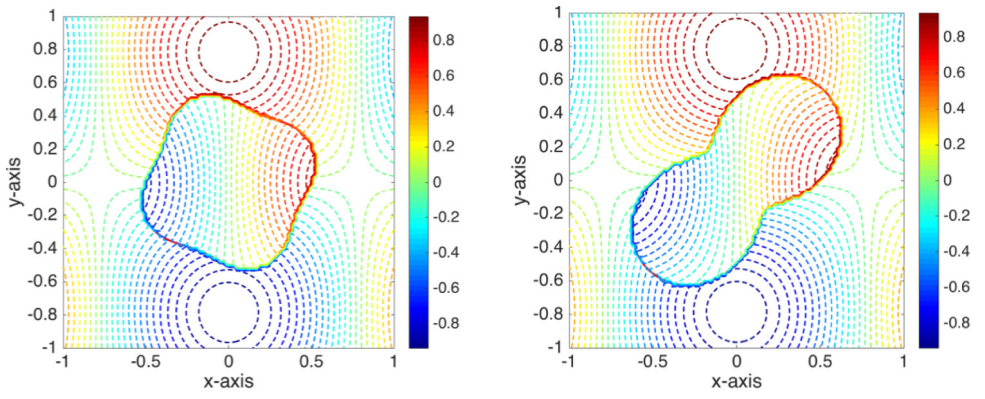
$$r = 0.5 + b\sin(k\theta), \tag{49}$$

where $b > 0$ is a real number governing the magnitude and curvature of the interface, $k > 0$ is a positive integer determining the number of "heads" of the curve, and θ is an angle in the range of $[0, 2\pi]$. Using the exact solution (48), jump conditions (3) on the interface are analytically given by

$$\begin{aligned} [u] &= (\cos(2x)\sin(2y) - \sin(2x)\cos(2y))\cos(t), \\ [\alpha u_n] &= 2(\sin(2x)\sin(2y)(\alpha^- \sin(\theta) - \alpha^+ \cos(\theta)) \\ &\quad + \cos(2x)\cos(2y)(\alpha^+ \sin(\theta) - \alpha^- \cos(\theta)))\cos(t). \end{aligned} \tag{50}$$

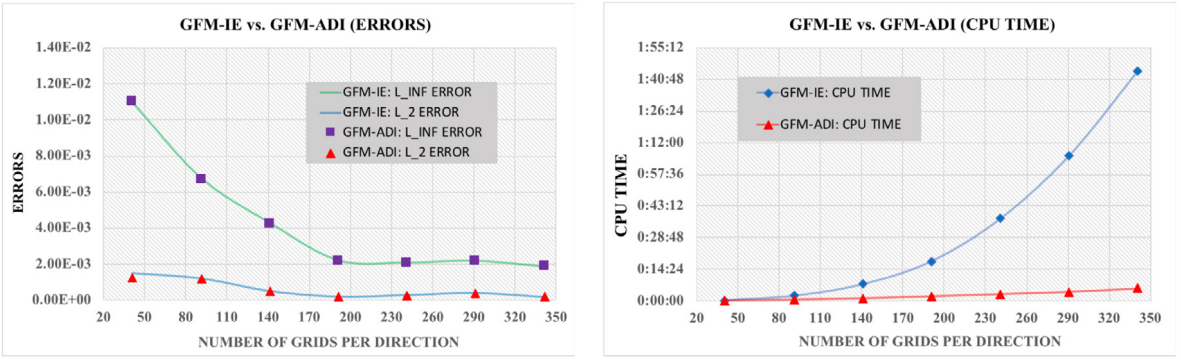
One can see that jump conditions (50) are actually time-and-space dependent, representing the most general jump conditions. In a similar fashion, source term f in Eq. (1) is found to be

$$f(x, y, t) = \begin{cases} (8\alpha^- \cos(t) - \sin(t))\sin(2x)\cos(2y), & \text{in } \Omega^- \\ (8\alpha^+ \cos(t) - \sin(t))\cos(2x)\sin(2y), & \text{in } \Omega^+ \end{cases} \tag{51}$$



(a) Example 1 (b) Example 2

Fig. 5. Analytic solutions and interfaces used in Examples 1 and 2.



(a) Numerical L_∞ and L_2 errors (b) CPU time

Fig. 6. Numerical errors and CPU time obtained in Example 1.

and the initial and boundary conditions are given by

$$u(x,y,0)=\sin(2x)\cos(2y)\ \text{ in } \Omega^-,\ \ u(x,y,0)=\cos(2x)\sin(2y)\ \text{ in } \Omega^+, \tag{52}$$

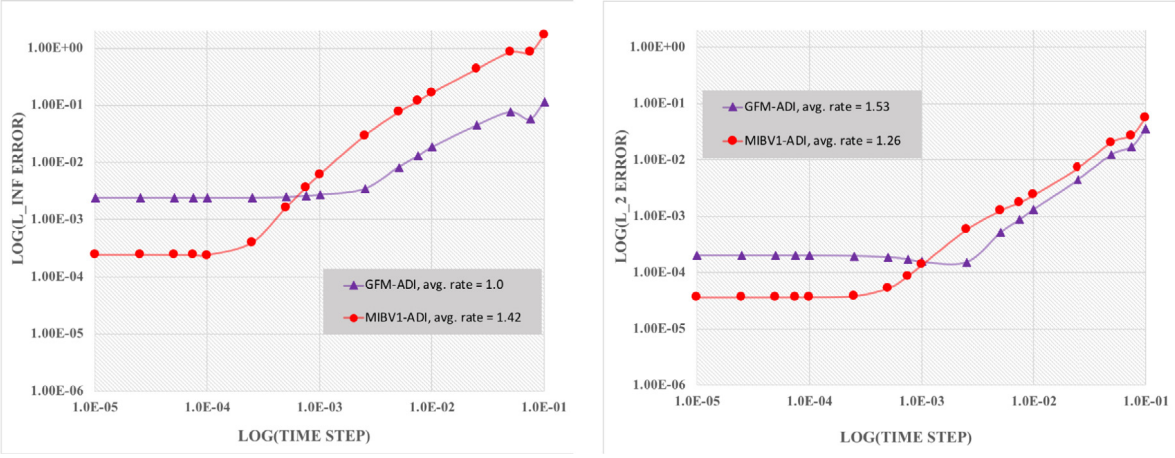
$$u(-0.99,y,t)=\cos(-1.98)\sin(2y)\cos(t),\ \ u(0.99,y,t)=\cos(1.98)\sin(2y)\cos(t), \tag{53}$$

$$u(x,-0.99,t)=\cos(2x)\sin(-1.98)\cos(t),\ \ u(x,0.99,t)=\cos(2x)\sin(1.98)\cos(t).$$

Example 1. In the first example, interface is constructed by selecting $(b,k)=(0.05,4)$, resulting in a smooth and regular "squircle" interface, demonstrated in Fig. 5a. The contour plot of the exact solution (48) is drawn in Fig. 5a, in which jumps across the interface can be easily observed.

We study the performance of the two time evolution methods in this example. To this end, GFM-IE and GFM-ADI are adopted to solve Eq. (1) from $t=0$ to $t=1$ with a fixed time step $\Delta t=1.0E{-}4$ and various spatial meshes ranging from the coarsest mesh (the number of grids per direction $N=41$) to the finest mesh ($N=341$). In the GFM-IE scheme, the linear system at each time step is solved by a biconjugate gradient iterative method implemented in the mathematical library *Slatec* https://people.sc.fsu.edu/~jburkardt/f_src/slatec/slatec.html. The obtained results are presented in Fig. 6.

One can see that the two time evolution methods are similarly accurate. The resulting L_2 and L_∞ errors are almost indistinguishable, as shown by continuous curves for the GFM-IE and discrete markers for the GFM-ADI in Fig. 6a. It is understood that the two time evolution methods differ only by a higher-order perturbation term, and the underlying spatial discretization method is the same. On the other hand, Fig. 6b shows that the GFM-ADI is much more efficient than the GFM-IE. Difference of their executed CPU time becomes more significant as N increases. In the case of the finest spatial resolution ($N=341$), GFM-IE is about 17 times slower than the GFM-ADI to achieve close accuracy at the final time. It clearly suggests that the ADI is more suitable to be used for time evolution. For this reason, we will only use the ADI in the following examples.



(a) L_∞-error (b) L_2-error

Fig. 7. Temporal accuracy and convergence rates obtained in Example 2. The average convergence rates are calculated by obtained results of time steps in the range of [2.5E−3, 1.0E−1] for the GFM-ADI and [2.5E−4, 1.0E−1] for the MIBV1-ADI, respectively.

Example 2. The second example is borrowed from [19] so that we can focus on comparing the performance of the two methods, GFM-ADI and MIBV1-ADI, while interested reader is directed to [19] for similar results obtained by the MIBV2-ADI. Interface in this example is constructed by selecting $(b, k) = (0.25, 2)$, resulting in a 2-headed interface with sharper curvature, as shown in Fig. 5b. Temporal and spatial accuracy, as well as corresponding convergence rates, are quantitatively examined in this example.

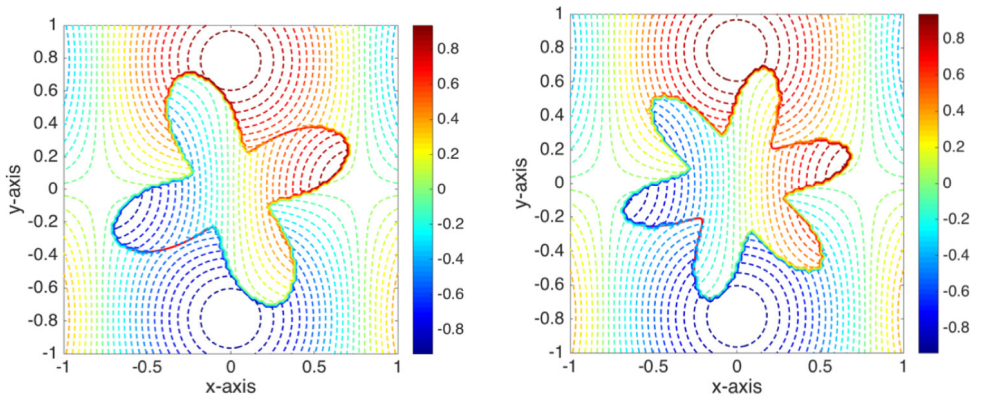
We first study the temporal accuracy and convergence rate. To this end, $N = 321$ is fixed, and the time step Δt varies from $1.0E{-}5$ to $1.0E{-}1$. The interface problem is then solved from $t = 0$ to $t = 1$. Numerical errors and convergence rates are presented in Fig. 7.

In both Figs. 7a–7b, it is observed that the GFM-ADI is able to achieve better accuracy when Δt is large, i.e., $\Delta t >$ 1.0E−3. However, both L_∞- and L_2-errors are flatten quickly after the point of $\Delta t =$ 1.0E−3. The best L_∞-error $\approx$ 2.4E−3 is achieved at $\Delta t =$ 2.5E−4 and the best L_2-error $\approx$ 1.6E−4 is achieved at $\Delta t =$ 1.0E−3, respectively. On the other hand, MIBV1-ADI starts off with larger errors at $\Delta t =$ 1.0E−1, but both errors decline in faster paces as Δt decreases. When $\Delta t \leq$ 1.0E−3, the MIBV1-ADI starts to achieve better accuracy than that obtained by the GFM-ADI. Eventually, the MIBV1-ADI achieves the best L_∞-error $\approx$ 2.4E−4 at $\Delta t =$ 1.0E−4 and the best L_2-error $\approx$ 3.7E−5 at $\Delta t =$ 1.0E−4, respectively. Numerical (average) convergence rates are also demonstrated in Figs. 7a–7b. These rates are calculated at Δt over the interval [2.5E−3, 1.0E−1] for the GFM-ADI and [2.5E−4, 1.0E−1] for the MIBV1-ADI, respectively. All temporal convergence rates are found to be greater than or equal to one.

The difference in the temporal convergence patterns of the GFM and MIBV1 methods is believed to be caused by different numerical treatments of the tangential jumps in two methods. In the GFM-ADI, errors introduced by neglecting tangential jumps at interface points are overwhelmed by errors introduced in temporal discretization when time step Δt is large. As Δt becomes smaller, the tangential errors do not change, and therefore become more significant and eventually dominate the overall errors when $\Delta t \leq$ 1.0E−3 so that no more reduction of errors is observed in both L_∞- and L_2-errors as Δt continues to decrease. On the other hand, tangential derivative u_τ^{n+1} is approximated by u_τ^n, and u_τ^n, in turn, is approximated by the numerical treatments described in Section 3.1, in each time evolution step t^n of the MIBV1-ADI. Accuracy of the former approximation obviously depends on the size of Δt. When Δt is large, i.e., $\Delta t >$ 1.0E−3, the error is significant, resulting in less accurate numerical solutions. However, this error diminishes quickly as Δt decreases, leading to more accurate numerical solutions when $\Delta t \leq$ 1.0E−3.

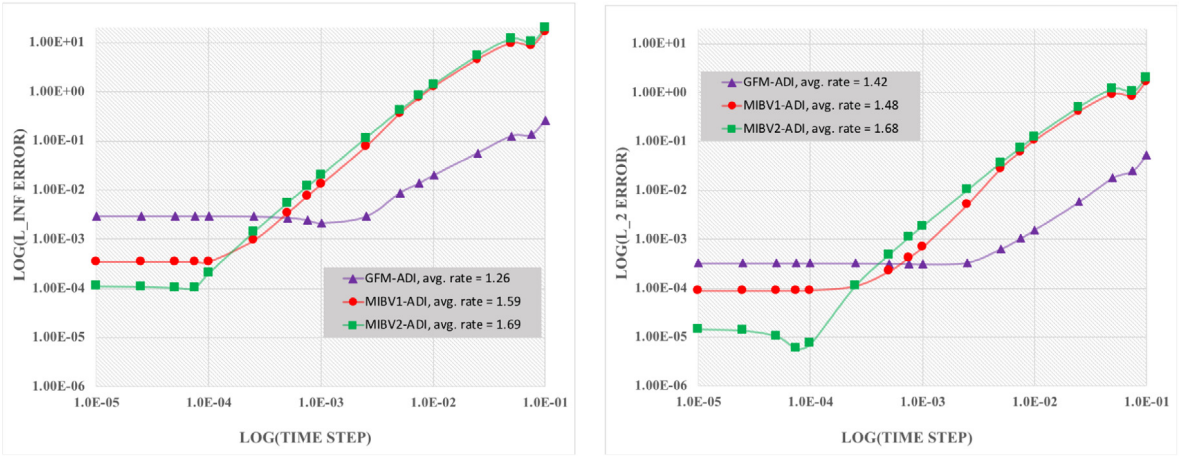
We next examine the spatial accuracy and convergence rate. A small time step $\Delta t =$ 2.5E−6 is fixed so that the temporal error can be neglected. The numerical errors for various mesh size N are presented in Table 1. Based on successive mesh refinements, numerically calculated convergence rates are reported for both L_2 and L_∞ error measurements. One can see that the spatial convergence rates obtained by both methods are mostly between one and two, while the rates obtained by the MIBV1-ADI are slightly higher than those obtained by the GFM-ADI.

Example 3. We continue to examine the performance of the two methods on an example with a more complicated 4-headed interface constructed by choosing $(b, k) = (0.25, 4)$, as shown in Fig. 8a. The MIBV2-ADI is also adopted in this example to be compared with the GFM-ADI and MIBV1-ADI. The same numerical setups in Example 2 are reused here for temporal experiments. On the other hand, the spatial experiments start with $N = 81$ in order to capture the



(a) Example 3 (b) Example 4

Fig. 8. Analytic solutions and interfaces used in Examples 3 and 4.



(a) L_∞-error (b) L_2-error

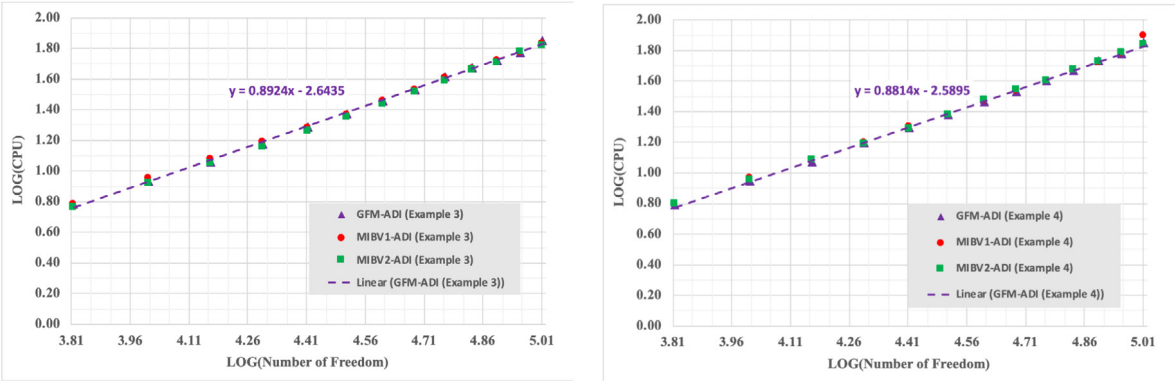
Fig. 9. Temporal accuracy and convergence rates obtained in Example 3. The average convergence rates are calculated by obtained results of time steps in the range of [2.5E−3, 1.0E−1] for the GFM-ADI, [1.0E−4, 1.0E−1] for the MIBV1-ADI, and [7.5E−5, 1.0E−1] for the MIBV2-ADI, respectively.

Table 1
Spatial accuracy and convergence rates obtained in Example 2.

N	GFM-ADI				MIBV1-ADI			
	L_∞		L_2		L_∞		L_2	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
21	2.08E−02		4.19E−03		1.09E−02		1.75E−03	
41	1.31E−02	0.69	2.04E−03	1.08	3.64E−03	1.64	5.77E−04	1.66
81	6.21E−03	1.10	7.02E−04	1.57	1.92E−03	0.93	5.25E−04	0.14
161	2.71E−03	1.21	3.41E−04	1.05	6.22E−04	1.64	1.31E−04	2.02
321	2.38E−03	0.19	2.01E−04	0.76	2.39E−04	1.39	3.66E−05	1.84

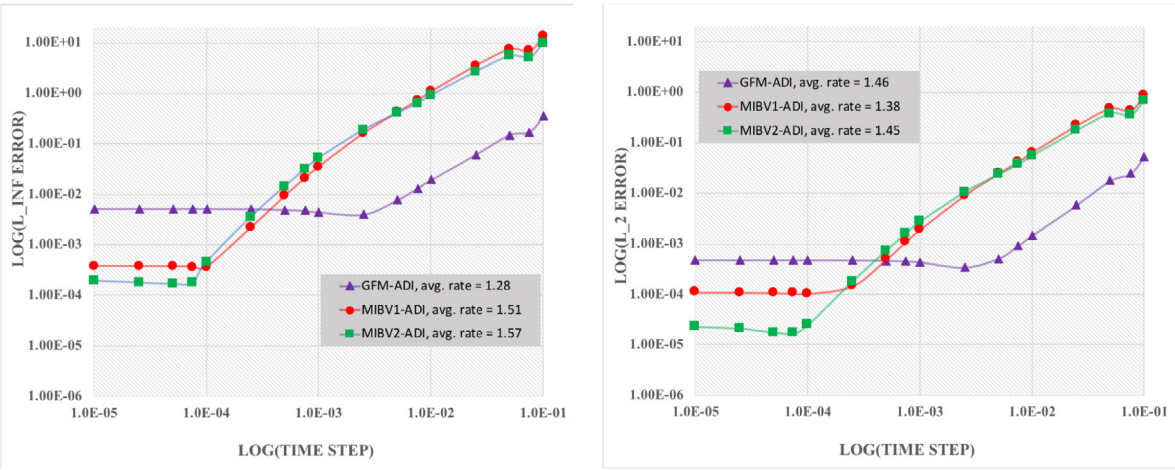
sharp change of the interface in this example. The obtained temporal results are presented in Fig. 9. One can see that all three methods are still capable of maintaining first order temporal convergence rates in this example.

The spatial results are presented in Table 2. The spatial convergence rates obtained by the GFM-ADI and MIBV1-ADI are still close the one, while the rates obtained by the MIBV2-ADI are closer to two. Results obtained in this example strongly suggest that all three methods are robust and capable of delivering accurate approximations on examples with fairly complicated interfaces.



(a) Example 3 (b) Example 4

Fig. 10. CPU time obtained in Examples 3 and 4.



(a) L_∞-error (b) L_2-error

Fig. 11. Temporal accuracy and convergence rates obtained in Example 4. The average convergence rates are calculated by obtained results of time steps in the range of [2.5E−3, 1.0E−1] for the GFM-ADI, [1.0E−4, 1.0E−1] for the MIBV1-ADI, and [7.5E−5, 1.0E−1] for the MIBV2-ADI, respectively.

Moreover, the obtained CPU time in Fig. 10a shows that all three methods are similarly efficient in practice. Here, with a fixed Δt and total time steps, the CPU time is plotted against the degree of freedom N in log–log scale. The complexities of three methods all appear to be linear with respect to N. As a demonstration, the linear trend of the data obtained for the GFM-ADI method is presented by the dashed line. The slope of this straight line is found to be 0.892. This indicates that the complexity of the GFM-ADI scales as $O(N^{0.892})$ in this example. The efficiency of the present three ADI methods is some of the fastest for solving parabolic PDEs, because the underlying Thomas algorithm in each ADI step is an $O(N)$ method.

Example 4. The most complicated 2D interface examined in this work is constructed by selecting $(b, k) = (0.2, 6)$, resulting in a 6-headed interface as shown in 8b. The obtained CPU time (Fig. 10b), temporal (Fig. 11) and spatial (Table 3) results are similar to those obtained in Example 3.

For this example, the space accuracies of the GFM-ADI and MIBV1-ADI schemes in estimating solution gradients are considered in Table 4. Here, for the numerical solution u at the final time step $t_n = 1$, the central differences are employed to approximate $\frac{\partial u}{\partial x}$ and $\frac{\partial u}{\partial y}$. Such a central difference approximation further degrades the spatial order of convergence. Thus, it can be seen in Table 4 that the GFM-ADI error is essentially divergent in L_∞ norm, while converges about $O(h^{0.5})$ in L_2 norm. By incorporating tangential jumps, the MIBV1-ADI yields a slightly better accuracy. Its gradient error is not divergent in L_∞ norm, and converges exactly on the order of 0.5 in L_2 norm. However, an additional error is involved in the present MIBV1-ADI gradient approximation, because in mADI schemes, the tangential jumps are calculated from the

Table 2
Spatial accuracy and convergence rates obtained in Example 3.

N	GFM-ADI				MIBV1-ADI				MIBV2-ADI			
	L_∞		L_2		L_∞		L_2		L_∞		L_2	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate	Error	Rate	Error	Rate
81	1.07E−02		9.72E−04		2.59E−03		4.41E−04		1.93E−03		3.28E−04	
161	7.76E−03	0.46	6.71E−04	0.54	1.31E−03	0.99	3.16E−04	0.49	5.59E−04	1.80	9.32E−05	1.83
321	2.97E−03	1.39	3.27E−04	1.04	3.54E−04	1.90	9.00E−05	1.82	1.12E−04	2.33	1.48E−05	2.66

Table 3
Spatial accuracy and convergence rates obtained in Example 4.

N	GFM-ADI				MIBV1-ADI				MIBV2-ADI			
	L_∞		L_2		L_∞		L_2		L_∞		L_2	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate	Error	Rate	Error	Rate
81	1.31E−02		1.62E−03		4.75E−03		9.11E−04		2.07E−03		3.70E−04	
161	5.78E−03	1.20	5.76E−04	1.51	1.85E−03	1.37	3.63E−04	1.34	5.26E−04	1.99	7.49E−05	2.32
321	5.19E−03	0.16	4.67E−04	0.30	3.80E−04	2.29	1.09E−04	1.74	2.02E−04	2.39	2.32E−05	1.70

Table 4
Accuracy and convergence rates for calculating gradients in Example 4.

N	GFM-ADI				MIBV1-ADI			
	L_∞		L_2		L_∞		L_2	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate
81	1.96		1.97E−01		9.91E−01		1.79E−01	
161	1.58	0.32	1.35E−01	0.54	9.91E−01	0.00	1.27E−01	0.50
321	2.99	−0.92	9.98E−02	0.44	9.91E−01	0.00	9.04E−02	0.50

Table 5
α-tests obtained in Example 4.

$\alpha^+ : \alpha^-$	GFM-ADI		MIBV1-ADI		MIBV2-ADI	
	L_∞	L_2	L_∞	L_2	L_∞	L_2
10 : 1	5.19E−03	4.67E−04	3.80E−04	1.09E−04	2.02E−04	2.32E−05
40 : 1	7.67E−03	6.44E−04	4.66E−04	1.26E−04	2.85E−04	3.19E−05
160 : 1	9.01E−03	7.18E−04	4.96E−04	1.32E−04	3.10E−04	3.49E−05
320 : 1	9.35E−03	7.34E−04	5.00E−04	1.33E−04	3.08E−04	3.53E−05
640 : 1	9.56E−03	7.42E−04	–	–	–	–
1 : 10	4.82E−03	8.44E−04	7.61E−04	1.99E−04	1.94E−04	3.59E−05
1 : 40	6.18E−03	1.31E−03	1.06E−03	2.73E−04	2.73E−04	5.28E−05
1 : 160	6.77E−03	1.52E−03	1.35E−03	3.38E−04	3.15E−04	6.07E−05
1 : 320	6.89E−03	1.57E−03	1.66E−03	4.67E−04	3.20E−04	6.20E−05
1 : 640	6.95E−03	1.59E−03	2.48E−03	9.94E−04	3.61E−04	7.50E−05

previous time step t_{n-1}, not at t_n. For this reason, the gradient improvement of the MIBV1-ADI over GFM-ADI is very slim. In particular, it is believed that the non-convergence pattern of the MIBV1-ADI in L_∞ norm is due to such a temporal error. Moreover, the gradient errors of the MIBV2-ADI are found to be exactly the same as those of the MIBV1-ADI, due to the same temporal error. Better gradient approximation techniques will be explored in the future for both GFM-ADI and MIB-ADI methods.

Besides the shape of the interface, another factor which is known to affect the performance of the proposed numerical methods is the contrast ratio of $\alpha^+ : \alpha^-$. When the ratio is extremely large or small, it has been found that MIB-type methods could be unstable, while GFM-type methods are still able to converge. In order to illustrate the impact of $\alpha^+ : \alpha^-$ on the stability of the proposed methods, we nail the smaller value of α equal to one on one side of the interface, and vary its larger value on the other side. Obtained results are presented in Table 5.

The GFM-ADI is found to converge in all tested cases shown in Table 5, while both MIB-type methods diverge in the case of $\alpha^+ : \alpha^- = 640 : 1$ or even larger ratios. It is interesting enough to see that both MIB-type methods diverge only when $\alpha^+ : \alpha^- = 640 : 1$, but converge when the ratio is reversed ($\alpha^+ : \alpha^- = 1 : 640$). We reason the divergence is caused by the numerical treatments of $[\alpha u_x]$ and $[\alpha u_y]$ in current implementations. As the matter of the fact, Eq. (37) and its analog in the y-direction are actually the ones used for estimating jumps $[\alpha u_x]$ and $[\alpha u_y]$ from the Ω^+-side of the interface, while Eq. (38) and its analog are not used at all in current MIB implementations. It points out a direction to improve the current MIB methods. If Eqs. (37)–(38), as well as their analogs in the y-direction, can be chosen wisely for approximating $[\alpha u_x]$ and $[\alpha u_y]$, the stability of the MIB methods may be improved. A development in this direction is under construction and will be reported elsewhere.

4. Solving parabolic interface problems in three dimensions by GFM and MIB-type methods

4.1. GFM and MIB-type methods in three dimensions

In three dimensions (3D), interface Γ becomes a closed hypersurface. The extension of the GFM is fairly straightforward, so it will be mentioned first.

In 3D, jump $[\alpha u_n]_\Gamma$ at an interface point is decomposed into three directional jumps in x, y, and z directions, respectively, in the GFM. Provided the hypothesis that, at an interface point, no jump occurs in the tangential plane orthogonal to the normal vector, each of above directional jumps is treated as the actual jump in that particular direction and used to correct $\delta_{xx}u$, $\delta_{yy}u$, and $\delta_{zz}u$ at irregular and corner grids near the interface point. More details can be found in [30].

On the other hand, tangential jumps are required to be estimated in MIB-type methods. In three dimensions, a particular non-orthogonal local coordinate system is proposed in [23] so that two particular tangential directions in the tangential plane are selected, and jumps in these two tangential directions are numerically approximated. The approximated tangential jumps are then used for estimating jumps in x, y and z directions. This unique treatment lowers the dimension by one for the task to estimate the jumps in x, y and z directions. This treatment has not been commonly seen in others' work. We shall elaborate this novel treatment in below.

At an interface point IPY(x_i, y_Γ, z_k) on the straight line where the plane $x = x_i$ meets the plane $z = z_k$, an outer normal vector is provided as $\xi = (n_x, n_y, n_z)$. A local coordinate system (ξ, η, ζ) is desired with two vectors, η and ζ, not necessarily orthogonal, to span the tangential plane of the interface at the point IPY. One possible choice is to select η on the straight line where the tangential plane intersects the plane $x = x_i$, and ζ on the intersection line where the tangential plane meets the plane $z = z_k$. A pair of unit vectors of η and ζ is given by

$$\eta = \left(0,\ \frac{n_z}{\sqrt{n_y^2+n_z^2}},\ -\frac{n_y}{\sqrt{n_y^2+n_z^2}}\right),\quad \zeta = \left(\frac{n_y}{\sqrt{n_x^2+n_y^2}},\ -\frac{n_x}{\sqrt{n_x^2+n_y^2}},\ 0\right). \tag{54}$$

The local coordinate system (ξ, η, ζ) can then be related to the global coordinate system (x, y, z) via a transformation formula

$$\begin{bmatrix}\xi\\ \eta\\ \zeta\end{bmatrix} = P_y \begin{bmatrix}x\\ y\\ z\end{bmatrix} = \begin{bmatrix} n_x & n_y & n_z \\ 0 & \frac{n_z}{\sqrt{n_y^2+n_z^2}} & -\frac{n_y}{\sqrt{n_y^2+n_z^2}} \\ \frac{n_y}{\sqrt{n_x^2+n_y^2}} & -\frac{n_x}{\sqrt{n_x^2+n_y^2}} & 0 \end{bmatrix} \begin{bmatrix}x\\ y\\ z\end{bmatrix}, \tag{55}$$

where the coordinate transformation matrix P_y is nonsingular, so that the desired jump conditions in x, y and z directions can be obtained by

$$\begin{pmatrix}(\psi_x)_\Gamma\\ (\psi_y)_\Gamma\\ (\psi_z)_\Gamma\end{pmatrix} := \begin{pmatrix}[\alpha u_x]_\Gamma\\ [\alpha u_y]_\Gamma\\ [\alpha u_z]_\Gamma\end{pmatrix} = P_y^{-1} \begin{pmatrix}[\alpha u_\xi]_\Gamma\\ [\alpha u_\eta]_\Gamma\\ [\alpha u_\zeta]_\Gamma\end{pmatrix}, \tag{56}$$

where

$$\begin{aligned} [\alpha u_\xi]_\Gamma &= \psi_\Gamma, \\ [\alpha u_\eta]_\Gamma &= \alpha^+ (\phi_\eta)_\Gamma + (\alpha^+ - \alpha^-)(u_\eta^-)_\Gamma = \alpha^- (\phi_\eta)_\Gamma + (\alpha^+ - \alpha^-)(u_\eta^+)_\Gamma, \\ [\alpha u_\zeta]_\Gamma &= \alpha^+ (\phi_\zeta)_\Gamma + (\alpha^+ - \alpha^-)(u_\zeta^-)_\Gamma = \alpha^- (\phi_\zeta)_\Gamma + (\alpha^+ - \alpha^-)(u_\zeta^+)_\Gamma. \end{aligned} \tag{57}$$

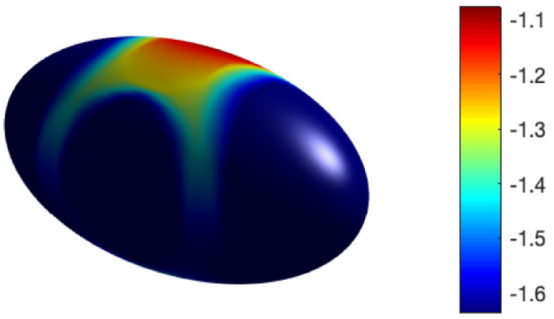
Notice that the three jumps, ψ_Γ, $(\phi_\eta)_\Gamma$ and $(\phi_\zeta)_\Gamma$, can all be obtained analytically via the imposed jump conditions at the interface point IPY, while the tangential jumps $(u_\eta^-)_\Gamma$, $(u_\eta^+)_\Gamma$, $(u_\zeta^-)_\Gamma$ and $(u_\zeta^+)_\Gamma$ in (57), must be numerically estimated, for example, by the method proposed in [23]. After approximations to directional jumps $[\alpha u_\xi]_\Gamma$, $[\alpha u_\eta]_\Gamma$, and $[\alpha u_\zeta]_\Gamma$ in (56) are obtained, one can follow similar procedure mentioned previously in one and two dimensions to correct $\delta_{xx}u$, $\delta_{yy}u$, and $\delta_{zz}u$ at the irregular and corner grids near the interface.

4.2. Time discretization

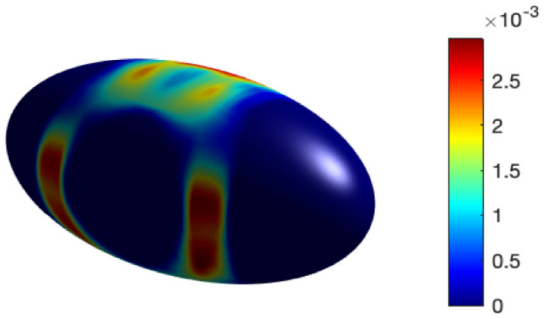
For both GFM and MIB-type methods, a Douglas-Rachford ADI method is employed for time evolution in 3D,

$$(1 - \Delta t\delta_{xx})u_{i,j,k}^* = u_{i,j,k}^n + \Delta t\delta_{yy}u_{i,j,k}^n + \Delta t\delta_{zz}u_{i,j,k}^n + \Delta t f_{i,j,k}^{n+1}, \tag{58}$$

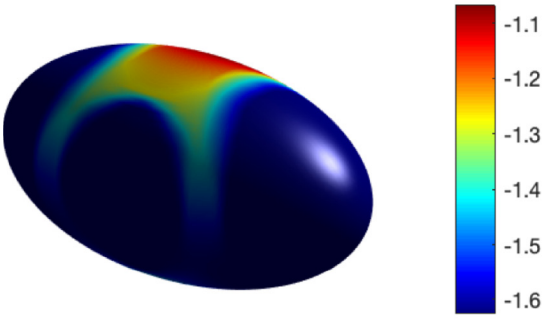
$$(1 - \Delta t\delta_{yy})u_{i,j,k}^{**} = u_{i,j,k}^* - \Delta t\delta_{yy}u_{i,j,k}^n, \tag{59}$$



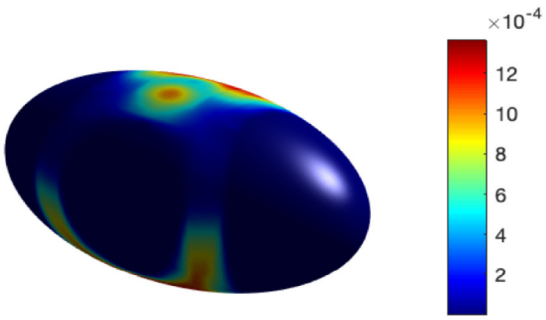
(a) Numerical solution obtained by GFM-ADI



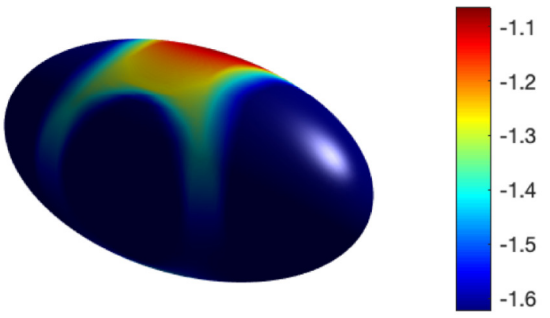
(b) Numerical errors obtained by GFM-ADI



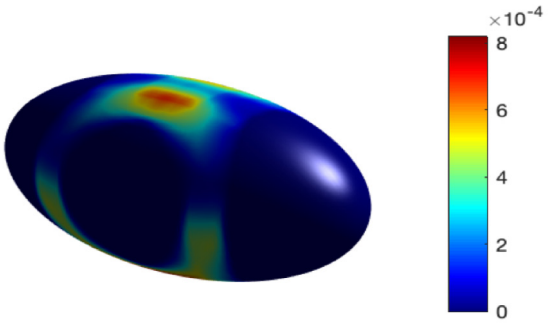
(c) Numerical solution obtained by MIBV1-ADI



(d) Numerical errors obtained by MIBV1-ADI



(e) Numerical solution obtained by MIBV1-ADI2



(f) Numerical errors obtained by MIBV2-ADI

Fig. 12. Numerical solutions and errors at the final time $t = 1$ obtained in Example 5.

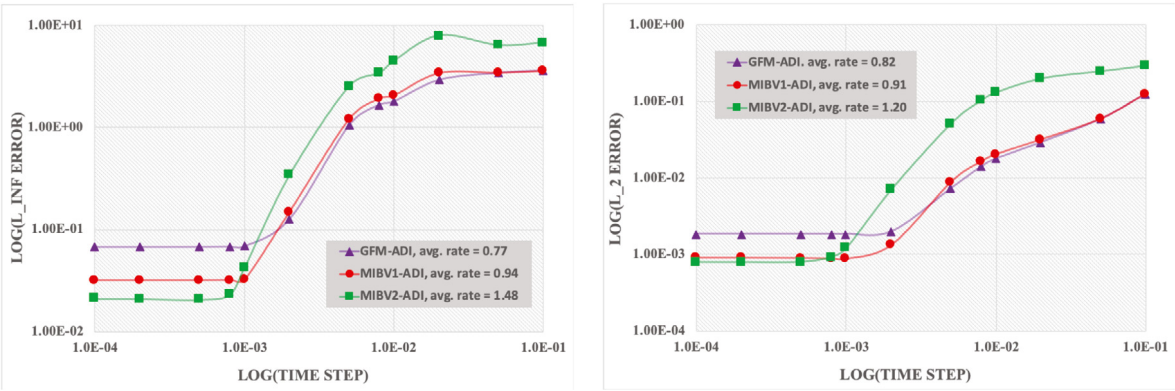
$$(1 - \Delta t \delta_{zz}) u_{i,j,k}^{n+1} = u_{i,j,k}^{**} - \Delta t \delta_{zz} u_{i,j,k}^{n}, \tag{60}$$

where u^* and u^{**} are two intermediate values. As in 2D, the temporal order of such ADI method is also one. The resulting fully discretized methods will also be named as GFM-ADI, MIBV1-ADI, and MIBV2-ADI.

4.3. Three dimensional numerical experiments

Example 5. An example is constructed to examine the capability of the three methods for solving problems with 3D interfaces. In this example, the interface is a simple smooth ellipsoid surface constructed as the zero level set of

$$S(x, y, z) = \frac{x^2}{4} + y^2 + z^2 - 1. \tag{61}$$



(a) L_∞-error (b) L_2-error

Fig. 13. Temporal accuracy and convergence rates obtained in Example 5. The average convergence rates are calculated by obtained results of time steps in the range of [8.0E−4, 2.0E−2] for all three methods.

Table 6
Spatial accuracy and convergence rates obtained in Example 5.

N	GFM-ADI				MIBV1-ADI				MIBV2-ADI			
	L_∞		L_2		L_∞		L_2		L_∞		L_2	
	Error	Rate	Error	Rate	Error	Rate	Error	Rate	Error	Rate	Error	Rate
20	2.56E−01		5.42E−02		2.97E−01		5.16E−02		3.64E−01		5.47E−02	
40	1.92E−01	0.41	1.44E−02	1.91	1.69E−01	0.82	1.34E−02	1.95	1.27E−01	1.52	1.30E−02	2.07
80	1.14E−01	0.76	4.89E−03	1.56	8.74E−02	0.95	3.41E−03	1.98	5.65E−02	1.17	3.19E−03	2.03
160	6.77E−02	0.75	1.86E−03	1.39	3.18E−02	1.46	9.17E−04	1.89	1.20E−02	2.24	1.59E−04	4.32

This interface is regular and convex, and each grid line cuts the interface at most twice. The numerical solution is constructed as

$$u(x,y,t)=\begin{cases}(\cos(2x)+\sin(2y)+\cos(2z))\sin(t) & \text{in } \Omega^-\\ (\sin(2x)+\cos(2y)+\sin(2z))\sin(t) & \text{in } \Omega^+\end{cases} \tag{62}$$

A 3D computational domain $[-4,4]\times[-4,4]\times[-4,4]$ is used. At the final time $t=1$, numerical solutions and errors on the surface obtained by the three proposed methods with $\Delta t = 1.0E-4$ and $N=160$ are graphically demonstrated in Fig. 12. More detailed temporal and spatial results are presented in Fig. 13 and Table 6, respectively. It is found that the temporal convergence rates are close to one, while the spatial convergence rates are even better than those obtained in examples with complicated 2D interfaces due to the simple shape of the interface used in this 3D example.

In summary, we conclude that all three methods can be successfully extended to solve three-dimensional interface problems. The obtained numerical accuracy and convergence rates are comparable to those obtained in two dimensional cases.

5. Conclusion

In this work, two new ADI schemes are developed and compared for solving parabolic interface problems in two and three dimensions. First, a GFM-ADI scheme is proposed for the first time in the literature. Second, a MIBV1-ADI scheme is constructed by downgrading the second order matched ADI or MIBV2-ADI method [19] to first order. Interestingly, the finite difference matrices of the GFM-ADI and MIBV1-ADI are the same in all dimensions – all being symmetric and diagonal. Moreover, both schemes maintain the ADI efficiency – the computational complexity for each time step scales as $O(N)$ for a total degree of freedom N in higher dimensions. The only difference is that the MIBV1-ADI and MIBV2-ADI schemes calculate tangential jumps which are omitted in the GFM. In other words, the MIB scheme can be regarded as higher order generalization of the GFM. In fact, high order MIB schemes, such as fourth order or even higher, have been constructed for solving general interface problems [20,21].

The performances of the GFM-ADI and MIBV1-ADI for solving parabolic interface problems with complex interfaces in two- and three-dimensions are investigated through numerical experiments. The obtained results show that both methods are capable of achieving first order of temporal convergence in ADI computations. For spatial accuracy, even though both schemes have the theoretical convergence rate being one, the MIBV1-ADI method is more accurate by taking into account

the tangential derivatives at the interface points in the formulations. Nevertheless, the GFM-ADI method is more stable due to its fully implicit nature, while the MIBV1-ADI becomes semi-implicit when calculating tangential jumps using function values in the previous time steps. The $O(N)$ complexity of both schemes is also numerically verified.

The two proposed ADI methods have great potential to be applied for solving practical interface models. For instance, the application of the GFM-ADI scheme for solving the nonlinear Poisson–Boltzmann equation has been considered in [64]. Future developments of MIB-type methods are under construction to improve their stability and will be reported in a separate work in the future.

CRediT authorship contribution statement

Chuan Li: Software, Validation, Visualization, Writing - original draft. **Zhihan Wei:** Software, Validation, Visualization, Writing - original draft. **Guangqing Long:** Writing - review & editing. **Cameron Campbell:** Validation. **Stacy Ashlyn:** Validation. **Shan Zhao:** Conceptualization, Methodology, Writing - review & editing.

Acknowledgments

This research is partially supported by the Simons Foundation, United States award 524151, the National Science Foundation (NSF) of USA under grant DMS-1812930, the Natural Science Foundation of China under grant 11461011, and the key project of Guangxi Provincial Natural Science Foundation of China under grants 2017GXNSFDA198014 and 2018GXNSFDA050014.

References

[1] C. Attanayake, D. Senaratne, Convergence of an immersed finite element method for semilinear parabolic interface problems, Appl. Math. Sci. 5 (3) (2011) 135–147.
[2] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math. 79 (2) (1998) 175–202.
[3] R.K. Sinha, B. Deka, Optimal error estimates for linear parabolic problems with discontinuous coefficients, SIAM J. Numer. Anal. 43 (2) (2005) 733–749.
[4] R.K. Sinha, B. Deka, Finite element methods for semilinear elliptic and parabolic interface problems, Appl. Numer. Math. 59 (8) (2009) 1870–1883.
[5] S. Wang, R. Samulyak, T. Guo, An embedded boundary method for elliptic and parabolic problems with interfaces and application to multi-material systems with phase transitions, Acta Math. Sci. 30 (2) (2010) 499–521.
[6] R.J. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (4) (1994) 1019–1044.
[7] L. Adams, Z. Li, The immersed interface/multigrid methods for interface problems, SIAM J. Sci. Comput. 24 (2) (2002) 463–479.
[8] F. Bouchon, G.H. Peichl, An immersed interface technique for the numerical solution of the heat equation on a moving domain, in: Numerical Mathematics and Advanced Applications 2009, Springer, 2010, pp. 181–189.
[9] F. Bouchon, G.H. Peichl, The immersed interface technique for parabolic problems with mixed boundary conditions, SIAM J. Numer. Anal. 48 (6) (2010) 2247–2266.
[10] J.D. Kandilarov, L.G. Vulkov, The immersed interface method for a nonlinear chemical diffusion equation with local sites of reactions, Numer. Algorithms 36 (4) (2004) 285–307.
[11] J.D. Kandilarov, L.G. Vulkov, The immersed interface method for two-dimensional heat-diffusion equations with singular own sources, Appl. Numer. Math. 57 (5–7) (2007) 486–497.
[12] J. Douglas Jr, On the numerical integration of $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$ by implicit methods, SIAM J. Appl. Math. 3 (1) (1955) 42.
[13] J. Douglas, D.W. Peaceman, Numerical solution of two-dimensional heat-flow problems, AIChE J. 1 (4) (1955) 505–512.
[14] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, SIAM, 2004.
[15] Z. Li, A. Mayo, ADI methods for heat equations with discontinuous along an arbitrary interface, in: Proceedings of Symposia in Applied Mathematics, Vol. 48, 1993, pp. 311–315.
[16] Z. Li, Y.-Q. Shen, A numerical method for solving heat equations involving interfaces, in: Electronic Journal of Differential Equations, Conf., Vol. 3, AMS, 1999, pp. 100–108.
[17] J. Liu, Z. Zheng, IIM-Based ADI finite difference scheme for nonlinear convection–diffusion equations with interfaces, Appl. Math. Model. 37 (3) (2013) 1196–1207.
[18] J. Liu, Z. Zheng, A dimension by dimension splitting immersed interface method for heat conduction equation with interfaces, J. Comput. Appl. Math. 261 (2014) 221–231.
[19] S. Zhao, A matched alternating direction implicit (ADI) method for solving the heat equation with interfaces, J. Sci. Comput. 63 (1) (2015) 118–137.
[20] S. Zhao, G. Wei, High-order FDTD methods via derivative matching for maxwell's equations with material interfaces, J. Comput. Phys. 200 (1) (2004) 60–103.
[21] Y. Zhou, S. Zhao, M. Feig, G.-W. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys. 213 (1) (2006) 1–30.
[22] C. Li, S. Zhao, A matched Peaceman–Rachford ADI method for solving parabolic interface problems, Appl. Math. Comput. 299 (2017) 28–44.
[23] Z. Wei, C. Li, S. Zhao, A spatially second order alternating direction implicit (ADI) method for solving three dimensional parabolic interface problems, Comput. Math. Appl. 75 (2018) 2173–2192.
[24] Z. Li, X. Chen, Z. Zhang, On multiscale ADI methods for parabolic pdes with a discontinuous coefficient, SIAM Multiscale Model. Simul. 16 (4) (2018) 1623–1647.
[25] I.-L. Chern, Y.-C. Shu, A coupling interface method for elliptic interface problems, J. Comput. Phys. 225 (2009) 2138–2174.
[26] C. Liu, C. Hu, A second order ghost fluid method for an interface problem of the Poisson equation, Commun. Comput. Phys. 22 (4) (2017) 965–996.
[27] 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 (2) (1999) 457–492.

[28] R.P. Fedkiw, X.-D. Liu, The ghost fluid method for viscous flows, in: Innovative Methods for Numerical Solution of Partial Differential Equations, World Scientific, 2002, pp. 111–143.
[29] Z. Ge, J.-C. Loiseau, O. Tammisola, L. Brandt, An efficient mass-preserving interface-correction level set/ghost fluid method for droplet suspensions under depletion forces, J. Comput. Phys. 353 (2018) 435–459.
[30] X.-D. Liu, R.P. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (1) (2000) 151–178.
[31] X.-D. Liu, T. Sideris, Convergence of the ghost fluid method for elliptic equations with interfaces, Math. Comput. 72 (244) (2003) 1731–1746.
[32] S. Hou, X.-D. Liu, A numerical method for solving variable coefficient elliptic equation with interfaces, J. Comput. Phys. 202 (2) (2005) 411–445.
[33] R.P. Fedkiw, Coupling an eulerian fluid calculation to a lagrangian solid calculation with the ghost fluid method, J. Comput. Phys. 175 (1) (2002) 200–224.
[34] T. Liu, B. Khoo, K. Yeo, Ghost fluid method for strong shock impacting on material interface, J. Comput. Phys. 190 (2) (2003) 651–681.
[35] C. Wang, T. Liu, B. Khoo, A real ghost fluid method for the simulation of multimedium compressible flow, SIAM J. Sci. Comput. 28 (1) (2006) 278–302.
[36] C. Wang, H. Tang, T. Liu, An adaptive ghost fluid finite volume method for compressible gas-water simulations, J. Comput. Phys. 227 (12) (2008) 6385–6409.
[37] S. Majidi, A. Afshari, Towards numerical simulations of supersonic liquid jets using ghost fluid method, Int. J. Heat Fluid Flow 53 (2015) 98–112.
[38] C. Farhat, A. Rallu, S. Shankaran, A higher-order generalized ghost fluid method for the poor for the three-dimensional two-phase flow computation of underwater implosions, J. Comput. Phys. 227 (16) (2008) 7674–7700.
[39] R.P. Fedkiw, T. Aslam, S. Xu, The ghost fluid method for deflagration and detonation discontinuities, J. Comput. Phys. 154 (2) (1999) 393–427.
[40] T. Ménard, S. Tanguy, A. Berlemont, Coupling level set/vof/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet, Int. J. Multiph. Flow. 33 (5) (2007) 510–524.
[41] W. Bo, J.W. Grove, A volume of fluid method based ghost fluid method for compressible multi-fluid flows, Comput. & Fluids 90 (2014) 113–122.
[42] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, J. Comput. Phys. 227 (18) (2008) 8395–8416.
[43] W. Donghong, Z. Ning, H. Ou, L. Jianming, A ghost fluid based front tracking method for multimedium compressible flows, Acta Math. Sci. 29 (6) (2009) 1629–1646.
[44] R.W. Houim, K.K. Kuo, A ghost fluid method for compressible reacting flows with phase change, J. Comput. Phys. 235 (2013) 865–900.
[45] B. Lalanne, L.R. Villegas, S. Tanguy, F. Risso, On the computation of viscous terms for incompressible two-phase flows with level set/ghost fluid method, J. Comput. Phys. 301 (2015) 289–307.
[46] T. Liu, B. Khoo, C. Wang, The ghost fluid method for compressible gas-water simulation, J. Comput. Phys. 204 (1) (2005) 193–221.
[47] T. Michael, J. Yang, F. Stern, A sharp interface approach for cavitation modeling using volume-of-fluid and ghost-fluid methods, J. Hydrodyn. Ser. B 29 (6) (2017) 917–925.
[48] V. Moureau, P. Minot, H. Pitsch, C. Bérat, A ghost-fluid method for large-eddy simulations of premixed combustion in complex geometries, J. Comput. Phys. 221 (2) (2007) 600–614.
[49] J. Qiu, T. Liu, B.C. Khoo, et al., Simulations of compressible two-medium flow by runge-kutta discontinuous Galerkin methods with the ghost fluid method, Commun. Comput. Phys. 3 (2) (2008) 479–504.
[50] H. Terashima, G. Tryggvason, A front-tracking/ghost-fluid method for fluid interfaces in compressible flows, J. Comput. Phys. 228 (11) (2009) 4012–4037.
[51] P. Trontin, S. Vincent, J.-L. Estivalezes, J.-P. Caltagirone, A subgrid computation of the curvature by a particle/level-set method. application to a front-tracking/ghost-fluid method for incompressible flows, J. Comput. Phys. 231 (20) (2012) 6990–7010.
[52] L.R. Villegas, R. Alis, M. Lepilliez, S. Tanguy, A ghost fluid/level set method for boiling flows and liquid evaporation: Application to the leidenfrost effect, J. Comput. Phys. 316 (2016) 789–813.
[53] L. Xu, T. Liu, Explicit interface treatments for compressible gas-liquid simulations, Comput. & Fluids 153 (2017) 34–48.
[54] L. Xu, C. Feng, T. Liu, Practical techniques in ghost fluid method for compressible multi-medium flows, Commun. Comput. Phys. 20 (3) (2016) 619–659.
[55] B. Van Poppel, O. Desjardins, J. Daily, A ghost fluid, level set methodology for simulating multiphase electrohydrodynamic flows with application to liquid fuel injection, J. Comput. Phys. 229 (20) (2010) 7977–7996.
[56] L. Cai, J.-H. Feng, W.-X. Xie, J. Zhou, Tracking discontinuities in shallow water equations and ideal magnetohydrodynamics equations via ghost fluid method, Appl. Numer. Math. 56 (12) (2006) 1555–1569.
[57] W.-X. Xie, L. Cai, J.-H. Feng, Tracking entropy wave in ideal mhd equations by weighted ghost fluid method, Appl. Math. Model. 31 (11) (2007) 2503–2514.
[58] X. Zeng, C. Farhat, A systematic approach for constructing higher-order immersed boundary and ghost fluid methods for fluid–structure interaction problems, J. Comput. Phys. 231 (7) (2012) 2892–2923.
[59] L. Xu, T. Liu, Modified ghost fluid method as applied to fluid-plate interaction, Adv. Appl. Math. Mech. 6 (1) (2014) 24–48.
[60] T. Liu, A. Chowdhury, B.C. Khoo, The modified ghost fluid method applied to fluid-elastic structure interaction, Adv. Appl. Math. Mech. 3 (5) (2011) 611–632.
[61] L. Xu, T. Liu, Optimal error estimation of the modified ghost fluid method, Commun. Comput. Phys. 8 (2010) 403–426.
[62] L. Xu, T. Liu, Accuracies and conservation errors of various ghost fluid methods for multi-medium riemann problem, J. Comput. Phys. 230 (12) (2011) 4975–4990.
[63] T. Liu, B. Khoo, The accuracy of the modified ghost fluid method for gas–gas riemann problem, Appl. Numer. Math. 57 (5–7) (2007) 721–733.
[64] S. Ahmed Ullah, S. Zhao, Pseudo-transient ghost fluid methods for the poisson-boltzmann equation with a two-component regularization, Appl. Math. Comput. 380 (2020) 125267.