Applied Mathematics and Computation 299 (2017) 28–44

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A matched Peaceman–Rachford ADI method for solving parabolic interface problems

Chuan Li[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

ARTICLE INFO

MSC:
65M06
65M12
35K05

Keywords:
Heat equation
Parabolic interface problem
Matched alternating direction implicit (ADI) method
Peaceman–Rachford ADI scheme
Matched interface and boundary (MIB)

ABSTRACT

A new Peaceman–Rachford alternating direction implicit (PR-ADI) method is proposed in this work for solving two-dimensional (2D) parabolic interface problems with discontinuous solutions. The classical ADI schemes are known to be inaccurate for handling interfaces. This motivates the development of a matched Douglas ADI (D-ADI) method in the literature, in which the finite difference is locally corrected according to the jump conditions. However, the unconditional stability of the matched ADI method cannot be maintained if the D-ADI is simply replaced by the PR-ADI. To stabilize the computation, the tangential derivative approximations in the jump conditions decomposition are substantially improved in this paper. Moreover, a new temporal discretization is adopted for formulating the PR-ADI method, which involves less perturbation terms. Stability analysis is conducted through eigenvalue spectrum analysis, which demonstrates the unconditional stability of the proposed method. The matched PR-ADI method achieves second order of accuracy in space in all tested problems with complex geometries and jumps, while maintaining the efficiency of the ADI. The proposed PR-ADI method is found to be more accurate than the D-ADI method in time integration, even though its formal temporal order is limited in the matched ADI framework.

1. Introduction

This work overcomes the difficulty of the recently developed matched alternating direction implicit (ADI) method [28] in constructing an unconditionally stable Peaceman–Rachford ADI (PR-ADI) discretization for solving two-dimensional (2D) parabolic equations with interfaces. Consider a heat equation

$$\frac{\partial u}{\partial t} = \nabla \cdot (\alpha \nabla u) + f, \text{ in } \Omega \subset R^2, \tag{1}$$

with some boundary conditions prescribed for u on the boundary $\partial\Omega$. The finite domain Ω is assumed to be split by a closed interface, $\Gamma = \Omega^- \cap \Omega^+$, into two sub-domains, $\Omega = \Omega^- \cup \Omega^+$. Here $u(\vec{x}, t)$ is a function of interest, α is the diffusion coefficient which can be discontinuous across the interface Γ, and $f(\vec{x}, t)$ is the source which may be even singular across

[*] Corresponding author. Fax: +1 205 348 7067.
E-mail addresses: szhao@ua.edu, szhao@bama.ua.edu (S. Zhao).

http://dx.doi.org/10.1016/j.amc.2016.11.033
0096-3003/© 2016 Elsevier Inc. All rights reserved.

the interface. On the interface Γ, the general jump conditions

$$[u] = u^+ - u^- = \phi(s,t), \ \ [\alpha u_n] = \alpha^+ \frac{\partial u^+}{\partial n} - \alpha^- \frac{\partial u^-}{\partial n} = \psi(s,t), \tag{2}$$

hold, which analytically relate the solution u on both sides of the interface Γ. In (2), s is the arc-length parameterization of the interface Γ, and n is the unit outer normal direction. The superscripts, $-$ and $+$, denote the limiting value of a function from one side or the other of the interface.

Parabolic interface problems (1) and (2) provide a general math model for many physical and engineering applications, such as the continuous casting in the metallurgical industry, the freezing process of perishable foodstuffs in the food engineering, and the magnetic fluid hypothermia treatment of cancer. The jump conditions (2) are of a quite general form, while for many relatively simple applications, $\phi = \psi = 0$.

Delivering accurate and efficient numerical solutions of parabolic interface problems (1) and (2) is not a trivial task. The major difficulties arise from two factors: First, the standard numerical methods, without treating the interface jump conditions carefully, often perform poorly (lower rate of accuracy) or even fail to converge, due to the fact that the physical solution of (1) and (2) is usually non-smooth or even discontinuous across the interface Γ. In order to restore the accuracy near the interface, jump conditions (2) have to be accounted for in the numerical discretization. Moreover, sophisticated numerical treatments are called for handling the complicated geometry associated with curved interfaces. Second, in practical applications where long-term simulations or stead state solutions are required, the implicit time stepping schemes are preferred over explicit ones so that the stability constrain associated with the explicit methods can be bypassed. Unconditional stability is particularly desired which permits the use of a larger time increment so that the equilibrium could be achieved in a faster pace. However, the efficiency is a big concern in the implicit time stepping, because one has to solve a linear algebraic system in each time step. For the present interface problems with discontinuous solutions, fast algebraic solvers with preconditioning are usually difficult to develop and implement.

In the past two decades, numerous numerical methods have been developed in the literature for solving parabolic and other interface problems. They can be roughly classified into two categories: unstructured grid interface methods and Cartesian grid interface methods. In unstructured grid interface methods, body-fitted grids are employed to fit to the material interfaces in order to provide the best flexibility for handling complex geometries. Based on such grids, finite element and finite volume methods [2,6,21,22,25] have been designed for solving parabolic interface problems. What are more related to the present study are Cartesian grid interface methods, in which a sophisticated interface algorithm is indispensable to accommodate the complex interface geometers and jump conditions. A famous example of such methods is the immersed interface method (IIM) [14], which achieves the second order of accuracy by rigorous imposing jump conditions in finite difference discretizations via local Taylor expansions. The IIM method has achieved a tremendous success in solving parabolic interface problems [1,4,5,12,13].

All numerical methods for parabolic interface problems face the same challenge in implicit time stepping, i.e., how to efficiently solve a large linear system at each time step. For unstructured grid interface methods, multigrid methods [11], particularly geometric multigrid methods, offer some of the best algebraic solvers with computation time of the order $O(N)$ in each time step, where N is the total spatial degree of freedom. Nevertheless, the construction of the restriction and prolongation in a multigrid cycle is nontrivial, especially with the presence of interfaces. An exceptional example is the fast multigrid IIM method developed by Adams and Li [1]. From the linear algebraic point of view, the ADI methods [7,8,19] offer a different approach to the problem, by reducing a multidimensional system to sets of independent one-dimensional (1D) systems. This is particularly attractive to Cartesian grid finite difference methods, because the resulting 1D systems are tridiagonal without considering jump conditions [3,10,24,27], and can be efficiently solved by using the Thomas algorithm [23]. The complexity of the ADI methods is also of the order of N, i.e., $O(N)$, similar to the multigrid methods.

However, the enforcement of the jump conditions in the ADI framework without compromising stability and spatial accuracy is known to be extremely difficult. One pioneer study in this direction is the IIM-ADI method developed by Li and Mayo [15]. For simplified jump conditions with α being continuous across the interface, the IIM-ADI method achieves second order of accuracy in space by adding some correction terms into the classical ADI scheme for irregular points near the interface [15–18]. For general jump conditions with α being a piecewise constant, the first ADI method that attains a spatial second order of accuracy is the matched ADI method recently developed in [28] for 2D heat equations. The success of the matched ADI method lies in two new contributions: First, a novel tensor-product decomposition of jump conditions (2) was proposed, which decouples 2D jump conditions into essentially 1D ones, in the same spirit of the ADI method. Then, these 1D conditions are enforced by the matched interface and boundary (MIB) scheme [26,29] in the matched ADI framework [28]. Second, fast algebraic solvers were developed for solving perturbed tridiagonal linear systems, as efficient as the Thomas algorithm.

In the matched ADI method [28], the Douglas ADI (D-ADI) scheme [7,8] is employed for time stepping, which yields an unconditionally stable time integration. Numerically, the temporal order of the D-ADI scheme is one. For a better precision, it is desired to consider another popular 2D ADI method, the Peaceman–Rachford ADI (PR-ADI) [19], whose formal temporal order is two. However, when the D-ADI is replaced by the PR-ADI in the existing matched ADI method with the same spatial discretizations, the computation becomes conditionally stable [28], despite the fact that the PR-ADI scheme is proved to be unconditionally stable for solving parabolic equations without interfaces.

The goal of this work is to improve the matched ADI framework so that an unconditionally stable PR-ADI scheme can be realized. Novel changes will be introduced in both temporal and spatial approximations, besides the replacement of the D-ADI by the PR-ADI. In particular, a more rigorous ADI formulation will be adopted in the temporal discretization so that the present formulation avoids some extra perturbation terms. Moreover, a key observation in our study is that if tangential derivatives at interface points, which are needed in the tensor product decomposition of the 2D jump conditions, are not numerically approximated, but are given analytically, the unconditional stability of the PR-ADI can be guaranteed. This motivates us to design several approximation and interpolation schemes to treat cases that are not considered in the original matched ADI method, e.g., grid points where the shape of the interface changes dramatically. With a high fidelity approximation to tangential derivatives, the unconditional stability of the PR-ADI scheme is restored in the new matched ADI method. Nevertheless, we note the temporal order of the proposed method is limited to be one, because the spatial discretization of the matched ADI method inevitably introduces an error on the order of the time increment, i.e., $O(\Delta t)$.

The rest of this work is organized as follows. After a brief introduction to the matched ADI method, Section 2 is devoted mainly to new numerical treatments. A stability analysis of the PR-ADI method will also be conducted. The stability and convergence of the proposed method will be examined experimentally in Section 3. Finally, conclusions and future developments are presented in Section 4.

2. Theory and algorithm

Consider a rectangular computational domain Ω with appropriate boundary conditions imposed on the boundary $\partial\Omega$. A uniform mesh is employed with a spacing $\Delta x = \Delta y = h$ in both x- and y- directions, yielding N_x and N_y nodes, respectively. Without the loss of the generality, each grid line, $x = x_i$ or $y = y_j$, is assumed to cut the interface Γ at most twice, and the diffusion coefficient α takes piecewisely defined constant values, α^+ in Ω^+ and α^- in Ω^-. The time increment is denoted by Δt, and the notation for the numerical solution $u_{i,j}^k = u(x_i, y_j, t_k)$ is used.

2.1. Classical time stepping methods

We first rewrite the governing Eq. (1) to a form that allows an easier formulation for the matched alternating direction implicit (ADI) method. To this end, dividing α on both sides of Eq. (1) yields

$$\frac{1}{\alpha}\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{f}{\alpha}, \quad \text{in} \quad \Omega^- \quad \text{or} \quad \Omega^+. \tag{3}$$

For α being a piecewisely defined constant, the numerical solution of (3) is identical to that of (1), provided that jump conditions (2) are rigorously enforced in the numerical discretization.

We consider the time stepping of u at a point (x, y) in the time interval $[t_k, t_{k+1}]$. In the existing matched ADI method [28], the implicit Euler semi-discretization is studied

$$\frac{u^{k+1} - u^k}{\alpha\Delta t} = \frac{\partial^2}{\partial x^2}u^{k+1} + \frac{\partial^2}{\partial y^2}u^{k+1} + \frac{f^{k+1}}{\alpha}, \tag{4}$$

while in the present study, the Crank–Nicolson semi-discretization is considered

$$\frac{u^{k+1} - u^k}{\alpha\Delta t} = \frac{\partial^2}{\partial x^2}u^{k+\frac{1}{2}} + \frac{\partial^2}{\partial y^2}u^{k+\frac{1}{2}} + \frac{f^{k+\frac{1}{2}}}{\alpha}. \tag{5}$$

Neglecting the spatial approximation at the moment, the following Douglas ADI (D-ADI) scheme

$$\begin{aligned} \frac{1}{\alpha}u^* - \Delta t\frac{\partial^2}{\partial x^2}u^* &= \frac{1}{\alpha}u^k + \Delta t\frac{\partial^2}{\partial y^2}u^k + \frac{\Delta t}{\alpha}f^{k+1}, \\ \frac{1}{\alpha}u^{k+1} - \Delta t\frac{\partial^2}{\partial y^2}u^{k+1} &= \frac{1}{\alpha}u^* - \Delta t\frac{\partial^2}{\partial y^2}u^k, \end{aligned} \tag{6}$$

is used in [28] as a one-dimensional (1D) implementation of the implicit Euler scheme (4). Similarly, a Peaceman–Rachford ADI (PR-ADI) scheme will be employed in this work to implement the Crank–Nicolson scheme (5)

$$\begin{aligned} \frac{1}{\alpha}u^* - \frac{\Delta t}{2}\frac{\partial^2}{\partial x^2}u^* &= \frac{1}{\alpha}u^k + \frac{\Delta t}{2}\frac{\partial^2}{\partial y^2}u^k + \frac{\Delta t}{2\alpha}f^{k+\frac{1}{2}}, \\ \frac{1}{\alpha}u^{k+1} - \frac{\Delta t}{2}\frac{\partial^2}{\partial y^2}u^{k+1} &= \frac{1}{\alpha}u^* - \frac{\Delta t}{2}\frac{\partial^2}{\partial x^2}u^* + \frac{\Delta t}{2\alpha}f^{k+\frac{1}{2}}. \end{aligned} \tag{7}$$

Obviously, the PR-ADI scheme (7) maintains a nice symmetric in x and y direction, in comparing with the D-ADI scheme (6). Both ADI schemes can be shown to be a higher order perturbation of the underlying time stepping method. The temporal order of the PR-ADI scheme is two, while that of the D-ADI scheme is one.

2.2. Spatial approximations of the matched ADI method

The matched ADI method incorporates the interface jump conditions (2) into the finite difference formulation in the procedure of spatial discretization [28]. To this end, the spatial nodes are classified into two categories based on their relative locations to the interface Γ. For a regular node (x_i, y_j) which stays away from the interface, the standard central difference formula is utilized to define the discrete operators δ_{xx} and δ_{yy}

$$\frac{\partial^2}{\partial x^2}u(x_i,y_j) \approx \delta_{xx}u_{i,j} := \frac{1}{h^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}), \tag{8}$$

$$\frac{\partial^2}{\partial y^2}u(x_i,y_j) \approx \delta_{yy}u_{i,j} := \frac{1}{h^2}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}). \tag{9}$$

The usual finite difference boundary treatment will be used if the regular node is on the boundary $\partial\Omega$. For an irregular node near the interface Γ, (8) and (9) must be corrected by approximately imposing jump conditions (2). To unify the notation, the corrected finite difference formula at the irregular node (x_i, y_j) is also denoted as δ_{xx} or δ_{yy}.

A key concept of the matched ADI method [28] is introducing a novel tensor-product decomposition of jump conditions (2), which decouples 2D conditions into essentially 1D ones. Consider an interface point which is an intersection point of the interface Γ and a x or y grid line. Denote the outer normal and tangential direction at this point as n and τ, respectively. The angle formed by n and the x-axis is denoted as θ. Differentiating along the τ direction, one more jump condition can be derived

$$\frac{\partial u^+}{\partial \tau} - \frac{\partial u^-}{\partial \tau} = [u_\tau] = \frac{\partial [u]}{\partial \tau} = \frac{\partial \phi}{\partial \tau} := \phi_\tau. \tag{10}$$

Jump conditions (2) and (10) cannot be applied in the ADI framework, because the normal and tangential directions are normally neither x nor y directions.

Through a careful study, new 1D jump conditions are proposed in [28], i.e.,

$$[u] = \phi, \quad [\alpha u_x] = \cos\theta\psi - \sin\theta(\alpha^+ - \alpha^-)u_\tau^+ - \sin\theta\alpha^-\phi_\tau := \bar{\psi} \tag{11}$$

in x-direction, and

$$[u] = \phi, \quad [\alpha u_y] = \sin\theta\psi + \cos\theta(\alpha^+ - \alpha^-)u_\tau^+ + \cos\theta\alpha^-\phi_\tau := \hat{\psi} \tag{12}$$

in y-direction, where u_τ^+ represents the tangential derivative of u on the Ω^+ side of the interface Γ. Note that u_τ^+ is the only unknown in (11) and (12), which has to be numerically approximated. Once u_τ^+ is accurately estimated, both $\bar{\psi}$ and $\hat{\psi}$ are known. We thus have essentially 1D jump conditions in x- and y- directions for the matched ADI method.

Assuming that jump conditions (11) and (12) have been given with a high fidelity, the matched interface and boundary (MIB) scheme [26,29] can be utilized to correct finite difference operators δ_{xx} and δ_{yy}. Taking δ_{yy} as an example, the corrected finite difference formulas at nodes (x_i, y_j) and (x_i, y_{j+1}) become

$$\delta_{yy}u_{i,j} = \frac{1}{h^2}(u_{i,j-1} - 2u_{i,j} + \tilde{u}_{i,j+1}), \quad \delta_{yy}u_{i,j+1} = \frac{1}{h^2}(\tilde{u}_{i,j} - 2u_{i,j+1} + u_{i,j+2}), \tag{13}$$

where $\tilde{u}_{i,j}$ and $\tilde{u}_{i,j+1}$ are two fictitious values at nodes (x_i, y_j) and (x_i, y_{j+1}), respectively. In the MIB scheme, these two fictitious are determined by solving the system of equations

$$\begin{aligned} w_{0,1}^+\tilde{u}_{i,j} + w_{0,2}^+u_{i,j+1} + w_{0,3}^+u_{i,j+2} &= w_{0,1}^-u_{i,j-1} + w_{0,2}^-u_{i,j} + w_{0,3}^-\tilde{u}_{i,j+1} + \phi, \\ \alpha^+\left(w_{1,1}^+\tilde{u}_{i,j} + w_{1,2}^+u_{i,j+1} + w_{1,3}^+u_{i,j+2}\right) &= \alpha^-\left(w_{1,1}^-u_{i,j-1} + w_{1,2}^-u_{i,j} + w_{1,3}^-\tilde{u}_{i,j+1}\right) + \hat{\psi}, \end{aligned} \tag{14}$$

resulting from imposing the jump conditions (12). Here $w_{I,J}^-$ and $w_{I,J}^+$ for $I = 0, 1$ and $J = 1, 2, 3$ are one-sided finite difference weights for left and right subdomains and are pre-calculated by Fornberg's method [9]. Here the subscript I represents the approximation to the value $(I = 0)$ and the first derivative $(I = 1)$, and J is for grid index. Solving (14) allows $\tilde{u}_{i,j}$ and $\tilde{u}_{i,j+1}$ being expressed as linear combinations of $u_{i,j-1}$, $u_{i,j}$, $u_{i,j+1}$, $u_{i,j+2}$, ϕ and $\hat{\psi}$. For example, we have symbolically

$$\tilde{u}_{i,j} = \sum_{p=1}^{4} w_{i,j}^p u_{i,j+p-2} + w_{i,j}^5\phi + w_{i,j}^6\hat{\psi}, \tag{15}$$

for some coefficients $w_{i,j}^p$, $p = 1, \ldots 6$. The same philosophy is equally applicable to the corner points where the interface crosses one grid line twice successively. The only difference is that two interface points of a corner point are taken care of at the same time so that four fictitious values are solved from a system of four equations by considering the jump conditions at both interface points. We refer to the original work [28] for more details.

2.3. The matched Douglas ADI method

We note that for an interface point with fixed x and y values, $\bar{\psi}$ and $\hat{\psi}$ involved in 1D jump conditions (11) and (12) are functions of the time t only. Since u_τ^+ in (11) and (12) has to be approximated by known values, it can only be accurate up to the time instant t_k, while $\bar{\psi}$ and $\hat{\psi}$ need to be invoked at a later time instant in the ADI formulation. This means that a temporal approximation is inevitably involved in the spatial approximations in the matched ADI method [28]. This subtle issue considerably complicates our algorithm development and numerical analysis.

In the existing matched D-ADI method [28], the spatial discretization is formulated after the D-ADI temporal semi-discretization (6) is formed. For instance, to approximate $\delta_{yy}u_{i,j}^{k+1}$ in (6), one substitutes linear combinations of the fictitious values $\tilde{u}_{i,j}^{k+1}$ and $\tilde{u}_{i,j+1}^{k+1}$ into (13). Nevertheless, due to the aforementioned issue, $\hat{\psi}^{k+1}$ has to be replaced by $\hat{\psi}^k$ in this process. Moreover, $\hat{\psi}^k$ will be approximated as a linear combination of several $u_{i,j}^k$ values at nearby nodes, and non-homogeneous values ψ^k and ϕ_τ^k at the underlying interface point. Thus, the corrected finite difference operator takes a form [28]

$$\delta_{yy}u_{i,j}^{k+1} = D_{yy}u_{i,j}^{k+1} + \hat{B}u_{i,j}^k + \hat{\Phi}^k, \tag{16}$$

where the first term $D_{yy}u_{i,j}^{k+1}$ has a band-width 4 for a usual irregular node, and a band-width 5 for a corner node. The second term $\hat{B}u_{i,j}^k$ is due to the approximation of u_τ^+ by some $u_{i,j}^k$ values. The last term $\hat{\Phi}^k$ is a collective term including non-homogeneous contributions from ϕ^k, ψ^k, and ϕ_τ^k. We can unify the notations of (9) and (16) as one by defining that for a regular node (x_i, y_j) away from the interface, $\hat{B}u_{i,j}^k$ and $\hat{\Phi}^k$ vanish and $D_{yy}u_{i,j}^{k+1}$ has a band-width 3 with the standard central finite difference coefficients. Similarly, to approximate $\delta_{xx}u_{i,j}^*$ in (6), we have

$$\delta_{xx}u_{i,j}^* = D_{xx}u_{i,j}^* + \bar{B}u_{i,j}^k + \bar{\Phi}^k. \tag{17}$$

Therefore, the full discretization of the D-ADI scheme (6) reads

$$\begin{aligned} \frac{1}{\alpha}u_{i,j}^* - \Delta t(D_{xx}u_{i,j}^* + \bar{B}u_{i,j}^k + \bar{\Phi}^k) &= \frac{1}{\alpha}u_{i,j}^k + \Delta t(D_{yy}u_{i,j}^k + \hat{B}u_{i,j}^k + \hat{\Phi}^k) + \frac{\Delta t}{\alpha}f_{i,j}^{k+1}, \\ \frac{1}{\alpha}u_{i,j}^{k+1} - \Delta t(D_{yy}u_{i,j}^{k+1} + \hat{B}u_{i,j}^k + \hat{\Phi}^k) &= \frac{1}{\alpha}u_{i,j}^* - \Delta t(D_{yy}u_{i,j}^k + \hat{B}u_{i,j}^k + \hat{\Phi}^k). \end{aligned} \tag{18}$$

By introducing the MIB approximations (16) and (17) after splitting the x and y directions [28], there are two shortcomings associated with the existing matched ADI temporal formulation (18). First, the connection between the D-ADI scheme (18) and the implicit Euler scheme (4) is no longer straightforward. One may still eliminate u^* from (18) and express u^{k+1} in terms of u^k. But many higher order perturbation terms have to be dropped in this process, which could be concerned from the accuracy point of view. Second, the immediate value u^* is not associated with any physical time instant. Thus the temporal error underlying (17) is uncertain. Moreover, one cannot evaluate $\bar{\Phi}$ at a specific time instant, but just at t_k. In fact, this is why all nonhomogeneous terms $\bar{\Phi}$ and $\hat{\Phi}$ are evaluated at t_k in (18).

2.4. A new time discretization of the matched ADI method

To overcome the disadvantages of the existing matched ADI temporal formulation, we propose a new temporal formulation in the present study. To construct a new PR-ADI scheme, we will introduce the MIB spatial discretization in the Crank–Nicolson scheme (5), before it is split into 1D ADI forms (7). Such a procedure will introduce less perturbation terms and allow an accurate evaluation of nonhomogeneous terms at the right time instant $t_{k+\frac{1}{2}}$.

Using similar notations, we propose a general MIB finite difference approximation in this study

$$\frac{\partial^2}{\partial x^2}u(x_i, y_j, t_{k+\frac{1}{2}}) = D_{xx}u_{i,j}^{k+\frac{1}{2}} + \bar{B}u_{i,j}^k + \bar{\Phi}_1^{k+\frac{1}{2}} + \bar{\Phi}_2^{k-\frac{1}{2}} + O(h^2 + \Delta t) \tag{19}$$

$$\frac{\partial^2}{\partial y^2}u(x_i, y_j, t_{k+\frac{1}{2}}) = D_{yy}u_{i,j}^{k+\frac{1}{2}} + \hat{B}u_{i,j}^k + \hat{\Phi}_1^{k+\frac{1}{2}} + \hat{\Phi}_2^{k-\frac{1}{2}} + O(h^2 + \Delta t). \tag{20}$$

where the discrete operators D_{xx} and D_{yy} are the same as in the D-ADI method [28], while $\bar{B}$ and $\hat{B}$ are different due to an improved approximation to u_τ^+ with details being offered later. Moreover, the approximation of u_τ^+ may involve some jump values at $t_{k-\frac{1}{2}}$ in the present work. Thus, additional terms for $\bar{\Phi}$ and $\hat{\Phi}$ at $t_{k-\frac{1}{2}}$ are introduced in (19) and (20). We note that $\bar{\Phi}_1$ and $\bar{\Phi}_2$ are different vectors, even if they are evaluated at the same time instant. It is because the approximation of u_τ^+ by some previous values of u, the temporal order of the PR-ADI scheme is limited to be one in (19) and (20).

By applying (19) and (20) in the Crank–Nicolson scheme (5), and using $u^{k+\frac{1}{2}} = (u^k + u^{k+1})/2 + O(\Delta t^2)$ as usual, we have

$$\begin{aligned}\frac{1}{\alpha\Delta t}u_{i,j}^{k+1} - \frac{1}{2}D_{xx}u_{i,j}^{k+1} - \frac{1}{2}D_{yy}u_{i,j}^{k+1} &= \frac{1}{\alpha\Delta t}u_{i,j}^{k}\\ &+ \frac{1}{2}D_{xx}u_{i,j}^{k} + \bar{B}u_{i,j}^{k} + \bar{\Phi}_1^{k+\frac{1}{2}} + \bar{\Phi}_2^{k-\frac{1}{2}} + \frac{1}{2}D_{yy}u_{i,j}^{k} + \hat{B}u_{i,j}^{k} + \hat{\Phi}_1^{k+\frac{1}{2}} + \hat{\Phi}_2^{k-\frac{1}{2}} + \frac{1}{\alpha}f_{i,j}^{k+\frac{1}{2}},\end{aligned} \tag{21}$$

with an error of $O(h^2 + \Delta t)$. By letting $F_{i,j}^{k+\frac{1}{2}} = \bar{B}u_{i,j}^{k} + \bar{\Phi}_1^{k+\frac{1}{2}} + \bar{\Phi}_2^{k-\frac{1}{2}} + \hat{B}u_{i,j}^{k} + \hat{\Phi}_1^{k+\frac{1}{2}} + \hat{\Phi}_2^{k-\frac{1}{2}} + f_{i,j}^{k+\frac{1}{2}}/\alpha$, (21) can be rewritten as

$$\left(\frac{1}{\alpha} - \frac{\Delta t}{2}D_{xx} - \frac{\Delta t}{2}D_{yy}\right)u_{i,j}^{k+1} = \left(\frac{1}{\alpha} + \frac{\Delta t}{2}D_{xx} + \frac{\Delta t}{2}D_{yy}\right)u_{i,j}^{k} + \Delta t F_{i,j}^{k+\frac{1}{2}}. \tag{22}$$

With $F_{i,j}^{k+\frac{1}{2}}$ being calculated first, we propose the following PR-ADI scheme

$$\begin{aligned}\left(\frac{1}{\alpha} - \frac{\Delta t}{2}D_{xx}\right)u_{i,j}^{*} &= \left(\frac{1}{\alpha} + \frac{\Delta t}{2}D_{yy}\right)u_{i,j}^{k} + \frac{\Delta t}{2}F_{i,j}^{k+\frac{1}{2}},\\ \left(\frac{1}{\alpha} - \frac{\Delta t}{2}D_{yy}\right)u_{i,j}^{k+1} &= \left(\frac{1}{\alpha} + \frac{\Delta t}{2}D_{xx}\right)u_{i,j}^{*} + \frac{\Delta t}{2}F_{i,j}^{k+\frac{1}{2}}.\end{aligned} \tag{23}$$

To see the relationship between the PR-ADI scheme (23) and the Crank–Nicolson scheme (22), one can eliminate $u_{i,j}^{*}$ from (23),

$$\left(\frac{1}{\alpha} - \frac{\Delta t}{2}D_{xx}\right)\left(\frac{1}{\alpha} - \frac{\Delta t}{2}D_{yy}\right)u_{i,j}^{k+1} = \left(\frac{1}{\alpha} + \frac{\Delta t}{2}D_{xx}\right)\left(\frac{1}{\alpha} + \frac{\Delta t}{2}D_{yy}\right)u_{i,j}^{k} + \Delta t F_{i,j}^{k+\frac{1}{2}}. \tag{24}$$

Expanding both hand sides of (24) and re-arranging the terms, we arrive at

$$\begin{aligned}&t\left(\frac{1}{\alpha} - \frac{\Delta t}{2}D_{xx} - \frac{\Delta t}{2}D_{yy}\right)u_{i,j}^{k+1}\\ &= \left(\frac{1}{\alpha} + \frac{\Delta t}{2}D_{xx} + \frac{\Delta t}{2}D_{yy}\right)u_{i,j}^{k} + \Delta t F_{i,j}^{k+\frac{1}{2}} - \frac{\alpha\Delta t^2}{4}D_{xx}D_{yy}(u_{i,j}^{k+1} - u_{i,j}^{k}).\end{aligned} \tag{25}$$

Obvious, the difference between the PR-ADI scheme (23) and the Crank–Nicolson scheme (22) is just one term, i.e., $\frac{\alpha\Delta t^2}{4}D_{xx}D_{yy}(u_{i,j}^{k+1} - u_{i,j}^{k})$. Moreover, this term is actually on the order of $O(\Delta t^3)$, which is a higher order perturbation term to the Crank–Nicolson scheme. Thus, the PR-ADI scheme (23) is numerically equivalent to the Crank–Nicolson scheme (22), and both schemes have a temporal order one in the context of the matched ADI scheme. We note that if the PR-ADI scheme was constructed by applying (19) and (20) directly to the ADI scheme (7), its connection with the Crank–Nicolson scheme is not that obvious, because this will involve many perturbation terms.

2.5. Stability analysis

Above discussions allow us to study the stability of the proposed PR-ADI method (23) by examining the stability of the algorithm which couples the Crank-Nicolson method (21) with the MIB spatial discretization. This is because the difference between these two schemes is just one higher order perturbation term, so that their stability conditions are essentially the same.

Denote $\mathbf{U}^k = [u_{1,1}^k, u_{2,1}^k, \ldots, u_{N_x,1}^k, u_{1,2}^k, u_{2,2}^k, \ldots, u_{N_x,2}^k, \ldots]^T$ a vector of length $N_x \times N_y$ containing all u values in the time step t_k. The matrix form of the derivative approximations (19) and (20) can be expressed as

$$\frac{\partial^2}{\partial x^2}\mathbf{U}^{k+\frac{1}{2}} \approx \mathbf{D}_{xx}\mathbf{U}^{k+\frac{1}{2}} + \bar{\mathbf{B}}\mathbf{U}^k + \bar{\mathbf{\Phi}}_1^{k+\frac{1}{2}} + \bar{\mathbf{\Phi}}_2^{k-\frac{1}{2}}, \tag{26}$$

$$\frac{\partial^2}{\partial y^2}\mathbf{U}^{k+\frac{1}{2}} \approx \mathbf{D}_{yy}\mathbf{U}^{k+\frac{1}{2}} + \hat{\mathbf{B}}\mathbf{U}^k + \hat{\mathbf{\Phi}}_1^{k+\frac{1}{2}} + \hat{\mathbf{\Phi}}_2^{k-\frac{1}{2}}, \tag{27}$$

where $\mathbf{D}_{xx}$, $\bar{\mathbf{B}}$, $\mathbf{D}_{yy}$, and $\hat{\mathbf{B}}$ are matrices of the dimension $N_x \times N_y$ by $N_x \times N_y$, and $\bar{\mathbf{\Phi}}_1^{k+\frac{1}{2}}$, $\bar{\mathbf{\Phi}}_2^{k-\frac{1}{2}}$, $\hat{\mathbf{\Phi}}_1^{k+\frac{1}{2}}$, and $\hat{\mathbf{\Phi}}_2^{k-\frac{1}{2}}$ are vectors of the length $N_x \times N_y$. For a regular node (x_i, y_j), entries of $\bar{\mathbf{B}}$, $\hat{\mathbf{B}}$, $\bar{\mathbf{\Phi}}_1^{k+\frac{1}{2}}$, $\bar{\mathbf{\Phi}}_2^{k-\frac{1}{2}}$, $\hat{\mathbf{\Phi}}_1^{k+\frac{1}{2}}$, and $\hat{\mathbf{\Phi}}_2^{k-\frac{1}{2}}$ will vanish, while $\mathbf{D}_{xx}$ and $\mathbf{D}_{yy}$ have usual tri-diagonal elements. For an irregular node, the corresponding rows of $\mathbf{D}_{xx}$ and $\mathbf{D}_{yy}$ will have four and five nonzero entries, respectively, in the non-corner and corner cases, due to the MIB interface treatment, such as (13) and (14). Thus, the matrices $\mathbf{D}_{xx}$ and $\mathbf{D}_{yy}$ are minor variations of the standard matrix for the central difference approximation, and dominate the stability. The matrices $\bar{\mathbf{B}}$ and $\hat{\mathbf{B}}$ are due to the approximation of u_τ^+ by some $u_{i,j}^k$ values, and will affect stability too. Nevertheless, the sparse entries of $\bar{\mathbf{B}}$ and $\hat{\mathbf{B}}$ are distributed in a rather random fashion – their locations depend on the interface geometry and grid size. One can only analyze their stability impact numerically. The vectors $\bar{\mathbf{\Phi}}_1^{k+\frac{1}{2}}$, $\bar{\mathbf{\Phi}}_2^{k-\frac{1}{2}}$, $\hat{\mathbf{\Phi}}_1^{k+\frac{1}{2}}$, and $\hat{\mathbf{\Phi}}_2^{k-\frac{1}{2}}$ are linear combinations of the nonhomogeneous values ϕ and ψ, and will not influence the stability.

In the present study, the stability analysis of the Crank–Nicolson method (21) will be conducted via the following matrix form

$$\mathbf{D}\mathbf{U}^{k+1} = \mathbf{B}\mathbf{U}^k + \mathbf{F}, \tag{28}$$

where $\mathbf{D} = \frac{1}{\alpha\Delta t}\mathbf{I} - \frac{1}{2}\mathbf{D}_{xx} - \frac{1}{2}\mathbf{D}_{yy}$, $\mathbf{B} = \frac{1}{\alpha\Delta t}\mathbf{I} + \frac{1}{2}\mathbf{D}_{xx} + \frac{1}{2}\mathbf{D}_{yy} + \bar{\mathbf{B}} + \hat{\mathbf{B}}$, and $\mathbf{F} = \bar{\mathbf{\Phi}}_1^{k+\frac{1}{2}} + \bar{\mathbf{\Phi}}_2^{k-\frac{1}{2}} + \hat{\mathbf{\Phi}}_1^{k+\frac{1}{2}} + \hat{\mathbf{\Phi}}_2^{k-\frac{1}{2}} + \frac{1}{\alpha}\mathbf{F}^{k+\frac{1}{2}}$. The matrix $\mathbf{D}$ is a modified central difference matrix, and is always invertible in our numerical computations. Denote $\mathbf{M} = \mathbf{D}^{-1}\mathbf{B}$, Eq. (28) can be rewritten as

$$\mathbf{U}^{k+1} = \mathbf{M}\mathbf{U}^k + \mathbf{D}^{-1}\mathbf{F}. \tag{29}$$

Thus, the stability of the proposed scheme is determined by the spectral radius ρ of the magnifying matrix $\mathbf{M}$

$$\rho(\mathbf{M}) = \max_j |\lambda_j|, \tag{30}$$

where λ_j are eigenvalues of $\mathbf{M}$. We note that an analytical spectrum analysis of $\mathbf{M}$ is not feasible, because the finite difference formulas underlying $\mathbf{D}$ and $\mathbf{B}$ crucially depend on the relative locations of the interface intersection points with respect to the uniform grid. Hence, the leading eigenvalues of $\mathbf{M}$ can only be calculated numerically.

2.6. Improvements on u_τ^+ approximation

In the matched ADI method, the tangential derivative u_τ^+ plays a critical role, because it essentially determines the jump values of the 1D jump conditions (11) and (12). We first note that u_τ^+ values will be referred at a future time, i.e., t_{k+1} and $t_{k+\frac{1}{2}}$ for the D-ADI and PR-ADI scheme respectively. However, one can only calculate u_τ^+ values at time step t_k based on known $u_{i,j}^k$ values. Thus, a temporal approximation is inevitably involved in the calculations of u_τ^+, which limits the temporal orders of the D-ADI and PR-ADI method. On the other hand, the spatial approximations to u_τ^+ employed by the original matched ADI method [28] have a great room for improvements. In the present study, we will focus on the development of new spatial approximations for u_τ^+, so that the accuracy and stability of the proposed PR-ADI scheme can be improved. Estimating u_τ^+ in the x and y directions is of great similarity. Thus, we take estimating u_τ^+ in the x direction as an example to demonstrate the numerical procedure, while that in the y direction is omitted.

The following notations are adopted in this subsection for an in-depth description. Let $(u_\tau^+)_{x_i,y_\Gamma}^k$ denote the value of u_τ^+ in the time step t_k, at an intersection point IPX(x_i, y_Γ) where the interface Γ intersects the grid line $x = x_i$. Through the point IPX, a tangent line τ is drawn and crosses neighboring x grid lines at two auxiliary points AXL(x_{i-1}, y_τ) and AXR(x_{i+1}, y_τ), as well as neighboring y grid lines at four auxiliary points AYL2(x_τ, y_{j-1}), AYL1(x_τ, y_j), AYR1(x_τ, y_{j+1}) and AYR2(x_τ, y_{j+2}) (red filled circles in Fig. 1(b)), respectively. Here x_τ and y_τ are simplified notations which take different values on various x and y grid lines.

In the D-ADI method [28], $(u_\tau^+)_{x_i,y_\Gamma}^k$ is estimated by the central difference formula using two auxiliary function values, the left auxiliary value u_{x_{i-1},y_τ}^k at the AXL and the right auxiliary value u_{x_{i+1},y_τ}^k at the AXR, respectively, as shown in Fig. 1(a). These two auxiliary values, in turn, are interpolated by the supporting function values at the nodes on the same lines $x = x_{i-1}$ and $x = x_{i+1}$ and all selected from the region of Ω^+ (open red circles in Fig. 1(a)) [28]. Similar treatment is used to estimate $(u_\tau^+)_{x_\Gamma,y_j}^k$ in the y direction. This spatial treatment is named **TAU1** method in this paper, and is incorporated into the framework of the Douglas ADI method, resulting in the D-ADI scheme mentioned in previous sections. It has been numerically demonstrated that D-ADI scheme is of the first order temporal accuracy and second order spatial accuracy, yet well maintains unconditional stability on all tested two-dimensional examples with various temporal-and-spatial dependent jump conditions imposed on the interfaces [28].

On the other hand, the results obtained by implementing the same spatial treatment in the Peaceman–Rachford ADI scheme to solve the same two-dimensional examples are surprisingly different from those obtained by the D-ADI scheme. It was found that the resulting scheme is able to achieve more accurate results compared to those obtained by the D-ADI scheme on some examples, while fails to converge on the others. Following-up investigates on the divergent examples showed that, when substituting the numerically estimated $(u_\tau^+)_{x_i,y_\Gamma}^k$ and $(u_\tau^+)_{x_\Gamma,y_j}^k$ by their exact values calculated by the formulas, the resulting scheme becomes unconditionally stable and converges. The conducted numerical experiments reveal the significance of the $(u_\tau^+)_{x_i,y_\Gamma}^k$ and $(u_\tau^+)_{x_\Gamma,y_j}^k$ in the proposed MIB method, and motivate us to seek better approximations to them. The following upgrades are the first attempts in the direction of delivering better approximations to $(u_\tau^+)_{x_i,y_\Gamma}^k$ and $(u_\tau^+)_{x_\Gamma,y_j}^k$. The improved spatial treatment is named the **TAU2** method, which is incorporated in the new time discretization scheme proposed in Section 2.4, yielding the **PR-ADI** method.

Improvement I: estimating $(u_\tau^+)_{x_i,y_\Gamma}^k$ and $(u_\tau^+)_{x_\Gamma,y_j}^k$ by the auxiliary values in the alternative directions

In the TAU1 method, the second order spatial accuracy obtained by estimating $(u_\tau^+)_{x_i,y_\Gamma}^k$ using the pair of the auxiliary values $\{u_{x_{i-1},y_\tau}^k, u_{x_{i+1},y_\tau}^k\}$ diminishes when the tangent line τ is steep, as shown in Fig. 1(b), and it even fails when τ is vertical. The

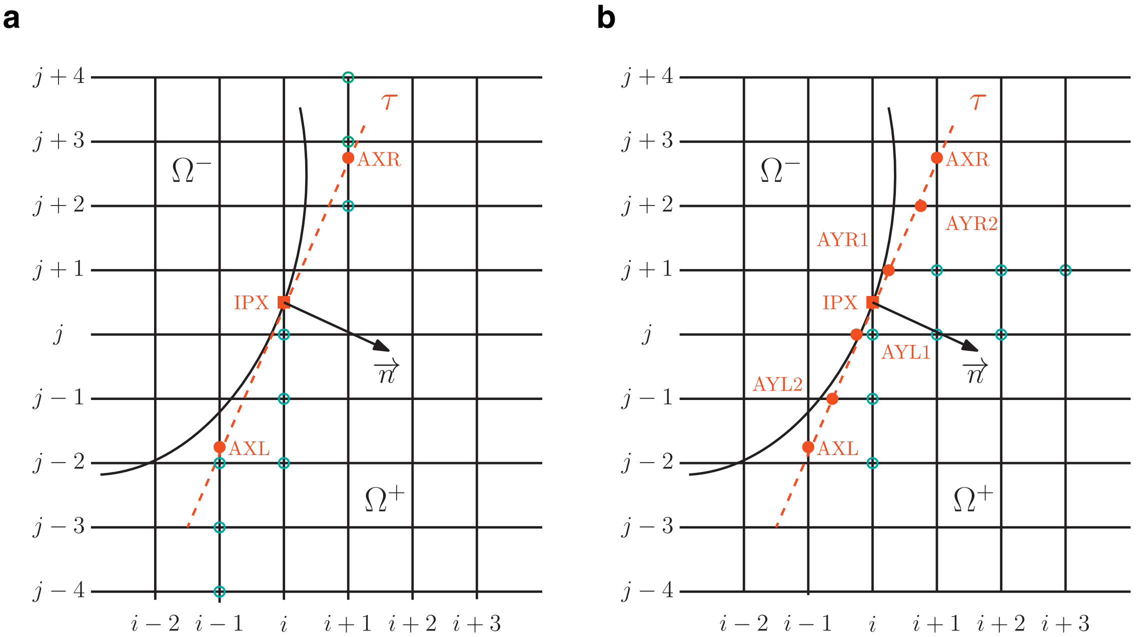

Fig. 1. Estimating u_τ^+ at an interface point IPX(x_i, y_Γ) (red filled square). (a) The TAU1 method [28]. The two selected auxiliary points are AXL(x_{i-1}, y_τ) and AXR(x_{i+1}, y_τ). The auxiliary values are interpolated/extrapolated by function values at supporting nodes (green open circle) all chosen from the Ω^+ side of the interface Γ. (b) Improvement I in the TAU2 method. Four additional auxiliary points, AYL2(x_τ, y_{j-1}), AYL1(x_τ, y_j), AYR1(x_τ, y_{j+1}) and AYR2(x_τ, y_{j+2}) are taken into account. The two selected auxiliary points are AYL1 and AYR1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

same issue occurs when estimating $(u_\tau^+)_{x_\Gamma,y_j}^k$ with the pair of the auxiliary values $\{u_{x_\tau,y_{j-1}}^k, u_{x_\tau,y_{j+1}}^k\}$ while the tangential line τ is flat. Under such deteriorated situations, we propose the following method as an alternative to approximate $(u_\tau^+)_{x_i,y_\Gamma}^k$. It shall be pointed out that the following procedure is performed only once before entering time evolution so that the additional computational cost is negligible.

On the grid line $x = x_i$, $i = 2, \ldots, N_x - 1$, the tangent line τ through the intersection IPX(x_i, y_Γ) is drawn. If τ is not vertical, the left auxiliary point AXL(x_{i-1}, y_τ) and the right auxiliary point AXR(x_{i+1}, y_τ) are marked and their coordinates are calculated. Otherwise, the two auxiliary points are marked as AXL($x_{i-1}, -\infty$) and AXR($x_{i+1}, +\infty$). Next, the distance between the points AXL and IPX is compared with a pre-described tolerance, *TOL* (5*h* for instance). If $||\text{IPX} - \text{AXL}|| \le TOL$, $(u_\tau^+)_{x_i,y_\Gamma}^k$ is marked to be approximated by the triplet of values $\{u_{x_{i-1},y_\tau}^k, u_{x_i,y_\Gamma}^k, u_{x_{i+1},y_\tau}^k\}$. Otherwise, four additional auxiliary points, AYL2(x_τ, y_{j-2}), AYL1(x_τ, y_{j-1}), AYR1(x_τ, y_j) and AYR2(x_τ, y_{j+1}) are taken into account. One of the two left auxiliary points (AYL1 and AYL2), is paired with the other from the right (AYR1 and AYR2) so that the interface point IPX is (nearly) centered in between. $(u_\tau^+)_{x_i,y_\Gamma}^k$ is then marked to be interpolated by the triplet of values $\{u_{x_\tau,y_p}^k, u_{x_i,y_\Gamma}^k, u_{x_\tau,y_q}^k\}$, where $y_p = y_{j-1}$ or y_j and $y_q = y_{j+1}$ or y_{j+2} are the grid lines where the pair of auxiliary points are selected from. After the two auxiliary points are marked, their associated weights for approximating $(u_\tau^+)_{x_i,y_\Gamma}^k$ at IPX are calculated. This procedure is graphically demonstrated in Fig. 1(b). The same procedure is carried out in y direction for estimating $(u_\tau^+)_{x_\Gamma,y_j}^k$ at the interface point IPY(x_Γ, y_j) on the grid line y_j, for $j = 2, \ldots, N_y - 1$. With the flexibility to select auxiliary points along the tangential line as intersection points of either x or y grid lines, the present procedure guarantees that the selected pair of auxiliary points are close enough to the point of interest IPX and are nearly central for accuracy.

Next, one can see that the function values at the selected pair of auxiliary points cannot be readily used since the two auxiliary points are in general off-grid. Thus, one needs to represent them as the linear combinations of some nearby on-grid function values, which will be called *supporting nodal values* in this work. Moreover, since $(u_\tau^+)_{x_i,y_\Gamma}^k$ and $(u_\tau^+)_{x_\Gamma,y_j}^k$ are the tangential derivatives on the Ω^+ side of the interface Γ, it naturally requires that the auxiliary values take the true function values when the auxiliary points are on the Ω^+ side of the interface Γ, while they take certain “fictitious” values extended from the Ω^+ side when the auxiliary points are on the other side, Ω^-, of the interface Γ.

In the TAU1 method, three supporting nodes are chosen on each of the three grid lines, $x = x_{i-1}, x_i$, and x_{i+1}. The supporting nodal values are then used to represent the function values at the two auxiliary values, as well as the function value at the interface point IPX. It requires that all these nine nodes are restricted to be in the region of Ω^+. The resulting nine-node representation of $(u_\tau^+)_{x_i,y_\Gamma}^k$ is demonstrated in Fig. 1(a). $(u_\tau^+)_{x_\Gamma,y_j}^k$ is represented in the same fashion and is omitted here.

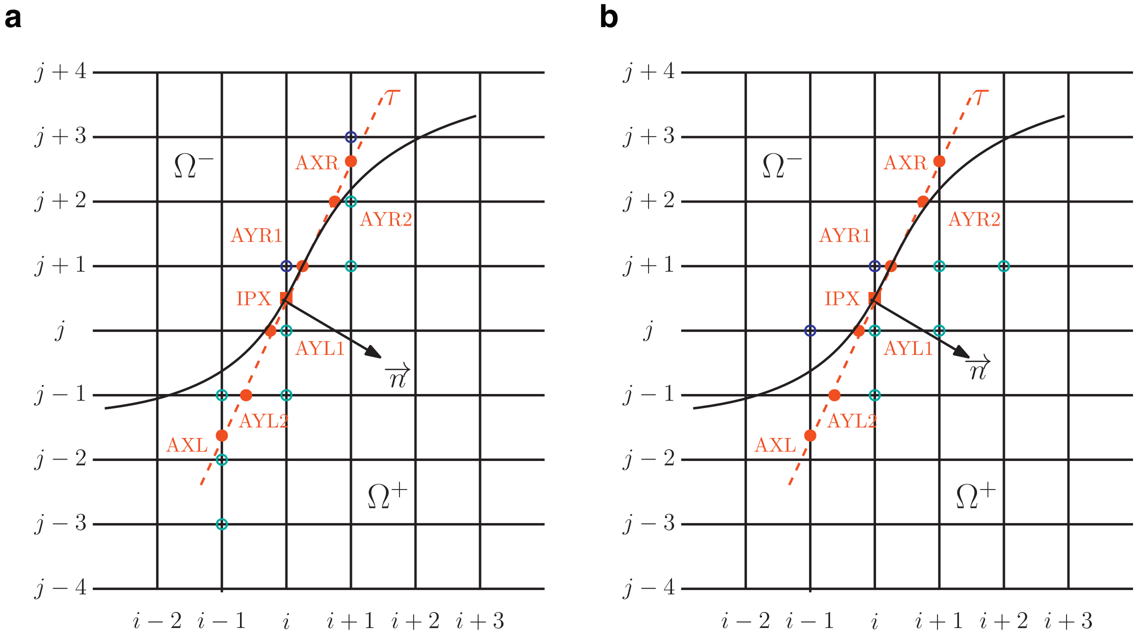

Fig. 2. Estimating u_τ^+ at an interface point IPX(x_i, y_Γ) (red filled square). (a) Improvement II in the TAU2 method. Fictitious values at supporting nodes (blue open circle) are utilized to interpolate the function values at the auxiliary points AXL and AXR. (b) The TAU2 method. Coupling the two proposed improvements in the TAU2 method results in estimating u_τ^+ by function values at auxiliary points AYL1 and AYR1. These two auxiliary values, in turn, are interpolated using both fictitious and true supporting nodal values. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The TAU1 method delivers reasonably accurate approximations to $(u_\tau^+)^k_{x_i,y_\Gamma}$ and $(u_\tau^+)^k_{x_\Gamma,y_j}$ when the geometry of the interface Γ is relatively simple, while its accuracy is jeopardized when the curvature of the interface changes dramatically near an interface point so that one (or both) auxiliary point ends in the region of Ω^-. An example is demonstrated in Fig. 2(a). The right auxiliary point AXR(x_{i+1}, y_τ) ends in Ω^- due to a large change in the curvature near IPX(x_i, y_Γ). When extrapolating $u^k_{x_{i+1},y_\tau}$ by three supporting nodal values on the grid line x_{i+1} from the Ω^+ side of the interface, the accuracy is limited due to the long spacing between the supporting nodes and AXR. It motivates us to propose the second improvement to estimate the auxiliary values $u^k_{x_p,y_\tau}$ and $u^k_{x_\tau,y_q}$ using nodes from both sides of Γ in the TAU2 method.

Improvement II: estimating an auxiliary value by using supporting nodes from both sides of Γ
To achieve a better accuracy, it is desired to interpolate an auxiliary value by using nearly central differences. However, when we restrict the supporting nodes being taken only from the Ω^+ side, such a goal is hardy realized. For example, we consider the interface point IPX(x_i, y_Γ) and two auxiliary points AXL(x_{i-1}, y_τ) and AXR(x_{i+1}, y_τ), shown in Fig. 2(a). The situation for AXL is obviously different from that of AXR and IPX. Because the point AXL is located in Ω^+ and far away from the interface Γ, its auxiliary value can be well approximated by three supporting nodal values, $\{u^k_{x_{i-1},y_{j-1}}, u^k_{x_{i-1},y_{j-2}}, u^k_{x_{i-1},y_{j-3}}\}$, that is,

$$u^k_{x_{i-1},y_\tau} = \sum_{q=j-3}^{j-1} w_{x_{i-1},y_q} u^k_{i-1,q}, \tag{31}$$

where w_{x_{i-1},y_q} are interpolating coefficients. Nevertheless, for AXR and IPX, extrapolations instead of interpolations have to be used when one just chooses supporting nodes within the Ω^+.

In the Improvement II, we propose to allow using supporting nodes from both sides of Γ so that nearly central finite differences are feasible in interpolating auxiliary values. In particular, out of three supporting nodal values, we allow the use of one node from the negative side of Γ. This is because in the MIB implementation of the jump conditions (11) and (12), one needs to generate fictitious values on two layers of fictitious nodes around the interface Γ, one immediately inside and one immediately outside Γ. Thus, we can utilize the fictitious value on one supporting node in the Ω^- to improve precision here.

We use a particular example to illustrate the new algorithm. Consider AXR(x_{i+1}, y_τ) again in Fig. 2(a). It is located in Ω^-, but an interpolation is still possible in our new method, by using three supporting values $\{\tilde{u}^k_{x_{i+1},y_{j+3}}, u^k_{x_{i+1},y_{j+2}}, u^k_{x_{i+1},y_{j+1}}\}$, that is

$$u^k_{x_{i+1},y_\tau} = \sum_{q=j+1}^{j+2} w_{x_{i+1},y_q} u^k_{i+1,q} + w_{x_{i+1},y_{j+3}} \tilde{u}^k_{i+1,j+3} \tag{32}$$

where w_{x_{i+1},y_q} are interpolating coefficients. Note that $\tilde{u}^k_{i+1,j+3}$ is a fictitious value at the supporting node (x_{i+1}, y_{j+2}), evaluated at the time step t_k. Following the general MIB fictitious value estimation (15), we have a particular form for $\tilde{u}^k_{i+1,j+3}$ in the present context

$$\tilde{u}^k_{i+1,j+3} = \sum_{p=1}^{4} w^p_{i+1,j+3} u^k_{i+1,j+p} + w^5_{i+1,j+3}\phi^{k-\frac{1}{2}}_{x_{i+1},y_\tau} + w^6_{i+1,j+3}\hat{\psi}^{k-\frac{1}{2}}_{x_{i+1},y_\tau}, \tag{33}$$

where $w^p_{i+1,j+3}$ are the fictitious value representation coefficients obtained by discretizing the jump condition (12). Nevertheless, the approximation to the jump value $\hat{\psi}^{k+\frac{1}{2}}_{x_{i+1},y_\tau}$ is not known yet at this stage. Thus, in (33), $\hat{\psi}_{x_{i+1},y_\tau}$ value from the previous time step, i.e., $\hat{\psi}^{k-\frac{1}{2}}_{x_{i+1},y_\tau}$, will be used. To be consistent, we also evaluate ϕ_{x_{i+1},y_τ} at $t_{k-\frac{1}{2}}$ in (33). We note that it is because of the use of jump values such as $\phi^{k-\frac{1}{2}}_{x_{i+1},y_\tau}$ and $\hat{\psi}^{k-\frac{1}{2}}_{x_{i+1},y_\tau}$ in estimating u^+_τ, the general finite difference derivative approximations in the matched PR-ADI method, i.e., (19) and (20), involve jump values in the time step $t_{k-\frac{1}{2}}$, i.e., $\bar{\Phi}_2^{k-\frac{1}{2}}$ and $\hat{\Phi}_2^{k-\frac{1}{2}}$.

By substituting (33) into (32), we can see that the auxiliary value $u^k_{x_{i+1},y_\tau}$ at the AXR depends on four u^k values and two jump values

$$u^k_{x_{i+1},y_\tau} = \sum_{p=1}^{4} w^p_{x_{i+1},y_\tau} u^k_{i+1,j+p} + w^5_{x_{i+1},y_\tau}\phi^{k-\frac{1}{2}}_{x_{i+1},y_\tau} + w^6_{x_{i+1},y_\tau}\hat{\psi}^{k-\frac{1}{2}}_{x_{i+1},y_\tau}. \tag{34}$$

The coefficients $w^p_{x_{i+1},y_\tau}$ can be computed in the preprocessing stage and saved in advance, while ϕ and $\hat{\psi}$ values will be updated during the time stepping. The interpolation of u_{x_i,y_Γ} at the IPX(x_i, y_Γ) will be conducted in a similar manner with supporting nodes from both sides of the interface Γ, see Fig. 2(a). Thus, the estimation of $(u^+_\tau)^k_{x_i,y_\Gamma}$ at the IPX depends on a total of eleven supporting nodal values in the time step t_k, and four jump values of ϕ and $\hat{\psi}$ in the time step $t_{k-\frac{1}{2}}$.

Moreover, the coupling of the two proposed improvements actually provides the TAU2 method additional flexibility and better accuracy for estimating $(u^+_\tau)^k_{x_i,y_\Gamma}$ and $(u^+_\tau)^k_{x_\Gamma,y_j}$ when the geometry of the interface Γ becomes complicated. A graphical demonstration is given in Fig. 2(b). When the curvature of the interface Γ changes rapidly near the interface point IPX, the upper-right intersection point, AXR, ends in Ω^-. Utilizing the AXR as an auxiliary point in the TAU1 method for estimating $(u^+_\tau)^k_{x_i,y_\Gamma}$ results in limited accuracy since the auxiliary value at AXR is eventually represented by supporting nodal values far on the other side (Ω^+) of the interface Γ. In contrast, improvement I allows the two auxiliary points AXL and AXR to be substituted by the auxiliary points AYL1(x_τ, y_j) and AYR1(x_τ, y_{j+1}), respectively, which locate much closer to the interface point IXP. By using improvement II, function values at the AYL1, IPX, and AYR1 are interpolated by $\{(x_{i-1}, y_j), (x_i, y_j), (x_{i+1}, y_j)\}$, $\{(x_i, y_{j-1}), (x_i, y_j), (x_i, y_{j+1})\}$, and $\{(x_i, y_{j+1}), (x_{i+1}, y_{j+1}), (x_{i+2}, y_{j+1})\}$, respectively. Note that one fictitious supporting nodal value is employed in each interpolation for avoiding extrapolations. Thus, the estimated $(u^+_\tau)^k_{x_i,y_\Gamma}$ is a linear combination of ten $u^k_{i,j}$ values in a compact neighborhood across the interface Γ, as well as six jump values in the time step $t_{k-\frac{1}{2}}$. In the most general case, the estimation of $(u^+_\tau)^k_{x_i,y_\Gamma}$ or $(u^+_\tau)^k_{x_\Gamma,y_j}$ at different locations may involve up to twelve $u^k_{i,j}$ values and six jump values nearby. A pseudo-code of performing both improvements of the TAU2 method is provided in Table 1.

With the obtained representations of $(u^+_\tau)^k_{x_i,y_\Gamma}$ and $(u^+_\tau)^k_{x_\Gamma,y_j}$, the 1D jump conditions (11) and (12) can now be updated. In particular, in order to be consistent with the proposed PR-ADI scheme (23), the analytical jump values ϕ, ψ, and ϕ_τ in (11) and (12) will be evaluated at the time instant $t_{k+\frac{1}{2}}$. One can then calculate $\hat{\psi}$ and $\bar{\psi}$ values as

$$\bar{\psi}^{k+\frac{1}{2}}_{x_i,y_\Gamma} = \cos\theta\,\psi^{k+\frac{1}{2}}_{x_i,y_\Gamma} - \sin\theta(\alpha^+ - \alpha^-)(u^+_\tau)^k_{x_i,y_\Gamma} - \sin\theta\,\alpha^-(\phi_\tau)^{k+\frac{1}{2}}_{x_i,y_\Gamma}, \tag{35}$$

$$\hat{\psi}^{k+\frac{1}{2}}_{x_\Gamma,y_j} = \sin\theta\,\psi^{k+\frac{1}{2}}_{x_\Gamma,y_j} + \cos\theta(\alpha^+ - \alpha^-)(u^+_\tau)^k_{x_\Gamma,y_j} + \cos\theta\,\alpha^-(\phi_\tau)^{k+\frac{1}{2}}_{x_\Gamma,y_j}. \tag{36}$$

Next, we supply $\bar{\psi}^{k+\frac{1}{2}}_{x_i,y_\Gamma}$ and $\hat{\psi}^{k+\frac{1}{2}}_{x_\Gamma,y_j}$ to the MIB jump conditions discretizations, such as (14), to generate fictitious values surrounding the interface Γ. For the proposed PR-ADI scheme, every term in (14) is evaluated at the time $t_{k+\frac{1}{2}}$. By using such fictitious values, one can correct the finite difference approximations, e.g., (13). This results in the proposed finite difference formulas (19) and (20) for the PR-ADI scheme. In particular, the discrete operators D_{xx} and D_{yy} are the same as in the D-ADI scheme [28]. The next three terms in both (19) and (20) are due to the approximations (35) and (36). Basically, $\bar{\Phi}_1^{k+\frac{1}{2}}$ and $\hat{\Phi}_1^{k+\frac{1}{2}}$ appear because of using ϕ, ψ and ϕ_τ values at the time step $t_{k+\frac{1}{2}}$. Terms $\bar{B}u^k_{i,j}$ and $\hat{B}u^k_{i,j}$ accounts for the contribution of $u^k_{i,j}$ values in evaluating $(u^+_\tau)^k$. Lastly, $\bar{\Phi}_2^{k-\frac{1}{2}}$ and $\hat{\Phi}_2^{k-\frac{1}{2}}$ are owing to the use of jump values at the time instant $t_{k-\frac{1}{2}}$ in calculating $(u^+_\tau)^k$. The finite difference formulas (19) and (20) yield perturbed tri-diagonal systems in

Table 1
Estimating $(u_\tau^+)^k_{x_i,y_\Gamma}$ and $(u_\tau^+)^k_{x_\Gamma,y_j}$ in the TAU2 method.

```
for $i = 2 : N_x - 1$ do
    Calculate the intersection point IPX($x_i$, $y_\Gamma$) where the interface $\Gamma$ intersects the grid line $x = x_i$.

    if the tangential line $\tau$ is not vertical then
        Calculate the left auxiliary point AXL($x_{i-1}, y_\tau$) and the right auxiliary point AXR($x_{i+1}, y_\tau$).
    else
        Set AXL($x_{i-1}, -\infty$) and AXR($x_{i+1}, +\infty$)
    end if

    if the distance $||$IPX $-$ AXL$|| \le TOL$ then
        $(u_\tau^+)^k_{x_i,y_\Gamma}$ is marked to be approximated by the triplet of values $\{u^k_{x_{i-1},y_\tau}, u^k_{x_i,y_\Gamma}, u^k_{x_{i+1},y_\tau}\}$.
    else
        Four additional auxiliary points AYL2($x_\tau, y_{j-2}$), AYL1($x_\tau, y_{j-1}$), AYR1($x_\tau$, $y_j$) and
        AYR2($x_\tau, y_{j+1}$) are calculated. One of the two left auxiliary points (AYL1 and AYL2), is
        paired with one (AYR1 and AYR2) from the right so that the interface point IPX is (nearly)
        centered in between. $(u_\tau^+)^k_{x_i,y_\Gamma}$ is marked to be interpolated by the triplet of values
        $\{u^k_{x_\tau,y_p}, u^k_{x_i,y_\Gamma}, u^k_{x_\tau,y_q}\}$, where $y_p = y_{j-1}$ or $y_j$ and $y_q = y_{j+1}$ or $y_{j+2}$ are the grid lines
        where the pair of auxiliary points are selected from.
    end if

    Determine the three supporting nodes to represent the left and right auxiliary values, respectively.
    if the supporting node is in $\Omega^-$ then
        the node is marked for using the fictitious value
    else
        the node is marked for using the true function value
    end if

    Calculate the corresponding weights at the supporting nodes to evaluate $(u_\tau^+)^k_{x_i,y_\Gamma}$. The
    obtained nodes and weights are saved.
end for

for $j = 2 : N_y - 1$ do
    The same procedure is taken to obtain the nodes and weights to evaluate $(u_\tau^+)^k_{x_\Gamma,y_j}$ at the
    intersection IPY($x_\Gamma$, $y_j$) where the interface $\Gamma$ crosses the grid line $y_j$.
end for
```

each step of the PR-ADI scheme (23). The perturbed system is first converted to tri-diagonal by row-eliminations, and then solved either by the Woodbury formula [20], or by the fast Thomas algorithm [23]. The detailed description of the algebraic computation is provided in [28].

3. Numerical experiments

In this section, we investigate the accuracy and stability of the proposed PR-ADI method. Piecewisely defined analytical solutions will be constructed in each example. The initial solution is chosen according the analytical solution at $t = 0$. The schemes will be carried out until a stopping time $t = T$. A square domain $[-0.99, 0.99] \times [-0.99, 0.99]$ with the Dirichlet boundary condition is considered in all examples. Here, the boundary data is simply given by the analytical solutions. Similarly, the jump conditions defining function and flux jumps across the interfaces are also calculated according to the given analytical solutions. For simplicity, the mesh sizes in both x and y directions are chosen to be the same, $N = N_x = N_y$ with $h = \Delta x = \Delta y$. Numerical errors in L_∞ and L_2 norms are reported in all examples.

In order to provide an detailed insight to the performance of the proposed PR-ADI method, numerical results obtained by the PR-ADI method and the previously developed D-ADI method [28] for solving 2D interface problems with various interfaces and jump conditions are reported and compared side by side. Moreover, we will consider two other reference schemes, i.e., **PR-EXACT** and **PR-TAU1**, in which the same spatial and temporal discretization as in the PR-ADI method are employed, except that $(u_\tau^+)^k$ is calculated differently. In the PR-EXACT scheme, $(u_\tau^+)^k$ is given analytically, so that this scheme provides a benchmark for us to access the performance of two proposed improvements or the TAU2 method for approximating $(u_\tau^+)^k$. In the PR-TAU1 scheme, $(u_\tau^+)^k$ is computed by the TAU1 method without new improvements. Consequently, the PR-TAU1 scheme will experience some instability.

The following examples are constructed to have the same exact solution

$$u(x, y, t) = \begin{cases} \sin(kx)\cos(ky)\cos(t), & \text{in } \Omega^-, \\ \cos(kx)\sin(ky)\cos(t), & \text{in } \Omega^+, \end{cases} \tag{37}$$

with the interface Γ provided by the formula

$$\Gamma: \quad r = \frac{1}{2} + b\sin(ms), \quad s \in [0, 2\pi], \tag{38}$$

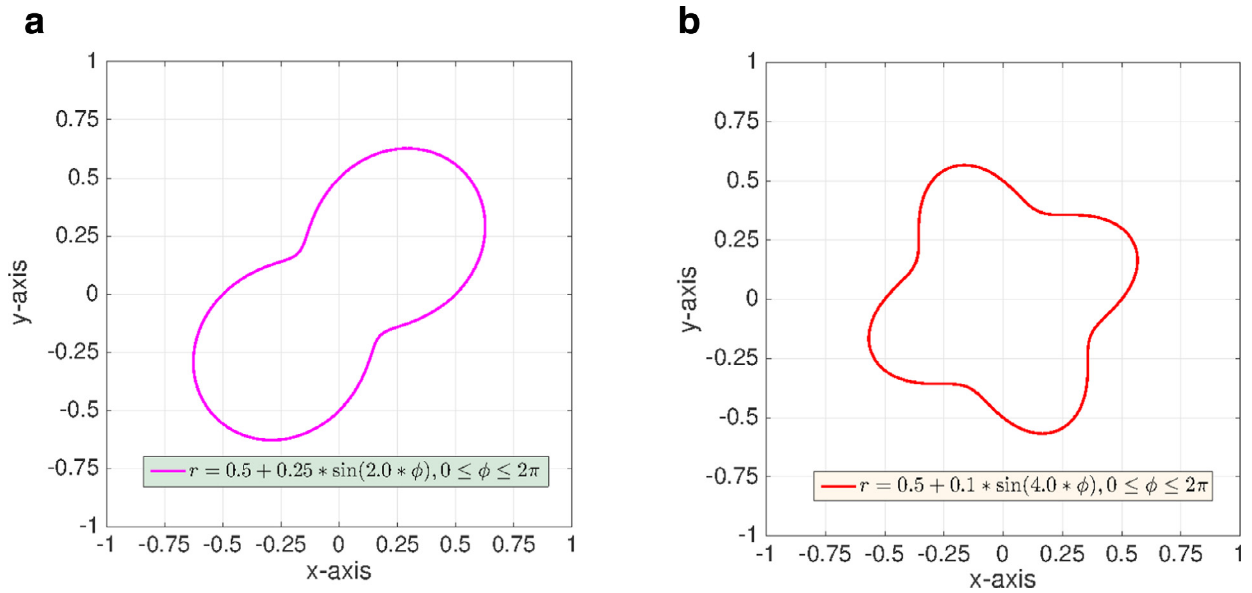

Fig. 3. The interfaces constructed by Eq. (38) with selected values of parameters. (a) Two leaves constructed by $(m, b) = (2, 1/4)$. (b) Four leaves constructed by $(m, b) = (4, 1/10)$.

where parameter m determines the number of “leaves” of the core region Ω^-, and parameter b controls the magnitude of the curvature. Two choices of values of parameters, $(m, b) = (2, 1/4)$ and $(m, b) = (4, 1/10)$, are used to construct the interfaces. The resulting interfaces are the same as those demonstrated in [28] and graphically shown in Fig. 3.

The solution (37) is obviously discontinuous across the interface. One can see that the interface jumps are temporal-and-spatial dependent over the interface Γ, and are calculated by

$$[u] = (\cos(kx)\sin(ky) - \sin(kx)\cos(ky))\cos(t), \tag{39}$$

$$[\alpha u_n] = (\sin(kx)\sin(ky)(\alpha^-\sin(\theta) - \alpha^+\cos(\theta)) + \cos(kx)\cos(ky) \\ (\alpha^+\sin(\theta) - \alpha^-\cos(\theta)))k\cos(t), \tag{40}$$

$$[u_\tau] = (\cos(kx)\cos(ky) + \sin(kx)\sin(ky))k(\cos(\theta) + \sin(\theta))\cos(t), \tag{41}$$

$$[\alpha u_x] = -(\alpha^+\sin(kx)\sin(ky) + \alpha^-\cos(kx)\cos(ky))k\cos(t), \tag{42}$$

$$[\alpha u_y] = (\alpha^+\cos(kx)\cos(ky) + \alpha^-\sin(kx)\sin(ky))k\cos(t). \tag{43}$$

Notice that (39)–(41) are those really being used by the proposed numerical methods, while (42) and (43) are provided and used only in the PR-EXACT method for numerical comparison.

Example 1. The interface in this example is constructed by Eq. (38) with the parameters $(m, b) = (2, 1/4)$ and is demonstrated in Fig. 3(a). Here α takes piecewisely constant values, $\alpha^+ = 10$ and $\alpha^- = 1$, so that the source term is explicitly given by

$$f(x, y, t) = \begin{cases} \left(2k^2\alpha^-\cos(t) - \sin(t)\right)\sin(kx)\cos(ky), & \text{in } \Omega^- \\ \left(2k^2\alpha^+\cos(t) - \sin(t)\right)\cos(kx)\sin(ky), & \text{in } \Omega^+. \end{cases} \tag{44}$$

We first investigate the spatial convergence. To this end, a time step $\Delta t = 10^{-6}$ is fixed and N_x is varied from 41 to 321 with increment 5 so that $\Delta t \ll h$ for all tested h's. The log–log plots of L_∞ and L_2 errors at the final time $T = 10$ are reported in Fig. 4. It can be seen that all methods converge for all tested cases, except that PR-TAU1 scheme diverges when $N_x = N_y = 201$ (indicated with a red dashed line) due to a nearly horizontal tangent line at one interface point. With a very small Δt, the stability of the PR-TAU1 method is still jeopardized by the rough estimation to the u_τ^+ at this interface point by the TAU1 method; in contrast, TAU2 successfully overcomes the difficulty occurred at this interface point and well maintains the stability of the PR-ADI method with an accurate estimation to the u_τ^+ at this interface point. Moreover, we calculate the overall convergence rates of four methods for all tested h values and reported these rates in the legends of Fig. 4. Although all numerical methods achieve the order of convergence close to two, the PR-EXACT method is obviously

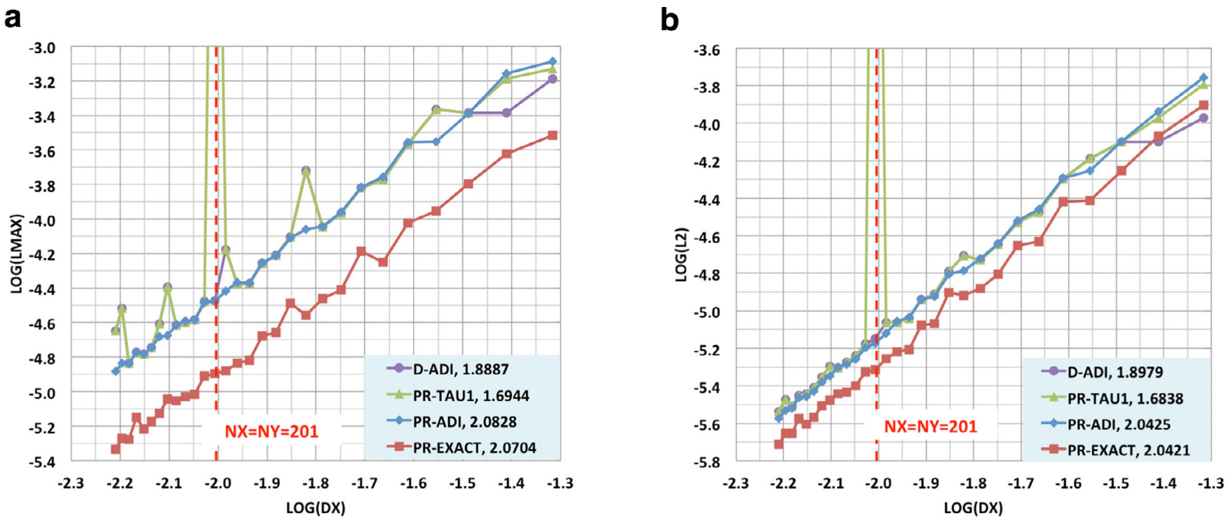

Fig. 4. Spatial convergence tests of Example 1 with fixed $\Delta t = 10^{-6}$ and various h. (a) L_∞ norm of the error; (b) L_2 norm of the error. The calculated overall convergence rates are demonstrated in the legends as well. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

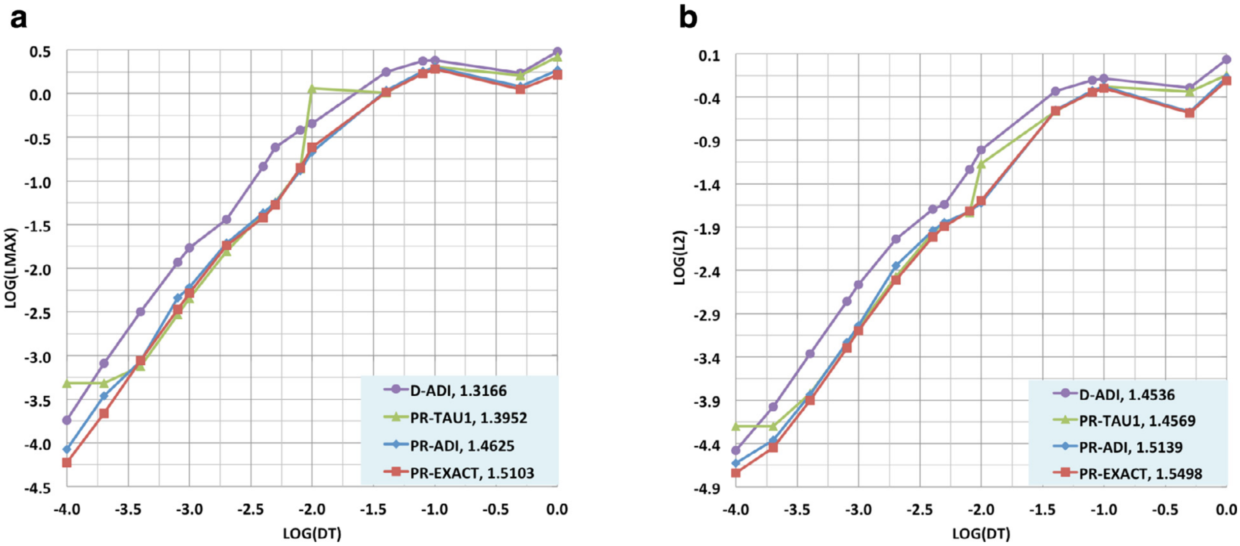

Fig. 5. Temporal convergence tests of Example 1 with fixed $N_x = 136$ and various Δt to the final time $T = 10$. (a) L_∞ norm of the error; (b) L_2 norm of the error. The convergence rates are calculated based on converged Δt values, i.e., $10^{-4} \le \Delta t \le 10^{-1}$, and are demonstrated in the legends.

the most accurate scheme, due to an analytical computation of u_τ^+. The order achieved by the PR-ADI method is higher than those of the PR-TAU1 and D-ADI methods. This indicates that the proposed two improvements are not only more stable but also more accurate than the TAU1 method in this example.

We next investigate the temporal convergence by fixing $N_x = N_y = 136$ and varying Δt from 10^{-4} to 1. The obtained L_∞ and L_2 errors at the final time $T = 10$ are reported in Fig. 5. It can be seen that all four methods remain stable with Δt up to $\Delta t = 1$. Nevertheless, the temporal convergence will start only when Δt is small enough. The same situation has also been observed in Ref. [28]. For this reason, the temporal convergence rates are calculated based on the converged Δt values, i.e., $10^{-4} \le \Delta t \le 10^{-1}$, in the present study. It can be observed from Fig. 5 that the PR-EXACT is still the most accurate scheme, and the PR-ADI results are actually very close to those of the PR-EXACT. The PR-ADI method clearly outperforms both the D-ADI and PR-TAU1 methods in this example.

Example 2. Similar spatial and temporal experiments are carried out on the example with the interface constructed by Eq. (38) with the parameter values $(m, b) = (4, 1/10)$. The interface, as shown in Fig. 3(b), is obviously subject to more significantly negative curvatures. The numerical results for temporal and spatial convergence tests are represented in Figs. 6 and 7, respectively. Our experiments show that the shape of the interface has very little effect on the temporal convergence,

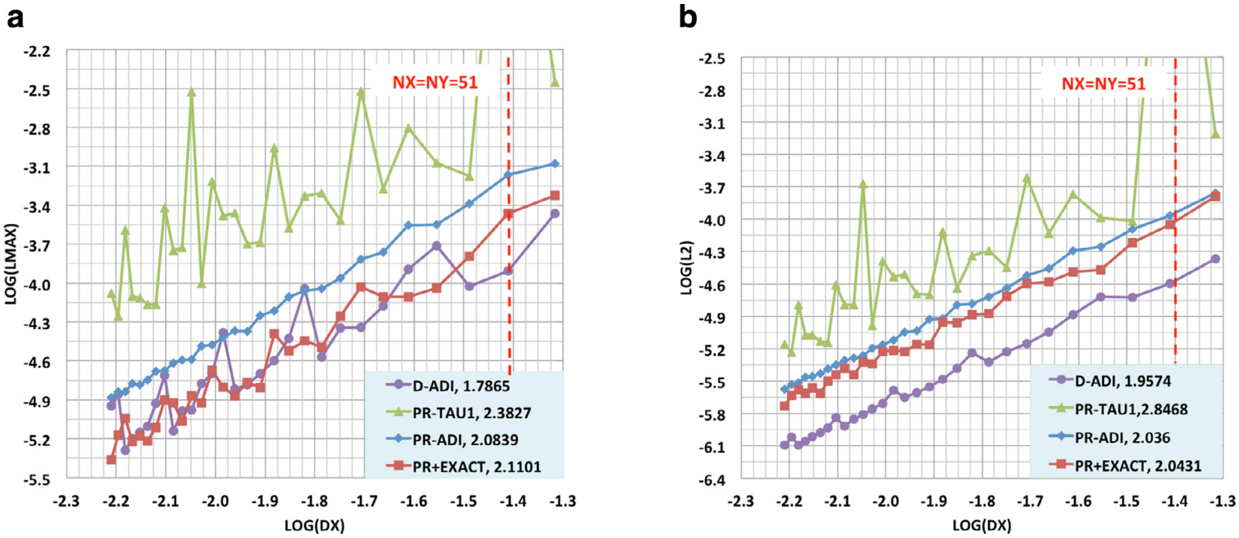

Fig. 6. Spatial convergence tests of Example 2 with fixed $\Delta t = 10^{-6}$ and various h. (a) L_∞ norm of the error; (b) L_2 norm of the error. The calculated overall convergence rates are demonstrated in the legends as well.

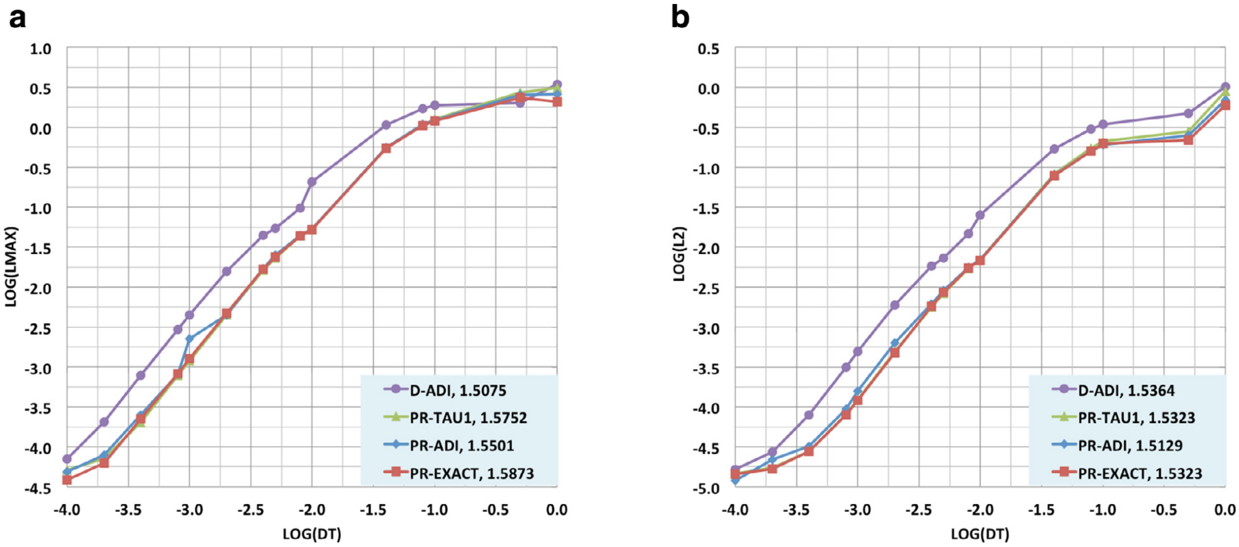

Fig. 7. Temporal convergence tests of Example 2 with fixed $N_x = 136$ and various Δt to the final time $T = 10$. (a) L_∞ norm of the error; (b) L_2 norm of the error. The convergence rates are calculated based on $10^{-4} \le \Delta t \le 10^{-1}$, and are demonstrated in the legends.

as shown in Fig. 7. The results are very similar to those obtained in previous example. In particular, all three Peaceman–Rachford ADI methods outperforms the Douglas ADI method. On the other hand, the shape of the interface does impact the spatial convergence significantly. It can be seen that the error curves obtained by the D-ADI and PR-TAU1 methods are subject to significant oscillations in Fig. 6, while those obtained by the PR-ADI method are of much insensitive to the change of the shape of the interface. Nevertheless, all methods still manage to converge with the order of accuracy close to two, except for one case that PR-TAU1 diverges when $N_x = N_y = 51$, as shown in Fig. 6. It is also interesting to note that the D-ADI method happens to be more accurate than the PR-EXACT method in L_2 norm in this example. Nevertheless, by using the same spatial discretization as the D-ADI method, the PR-TAU1 method is the most inaccurate scheme in Fig. 6.

The numerical experiments performed in Examples 1 and 2 have strongly suggested the PR-ADI method is more accurate in time and more stable comparing to the D-ADI and PR-TAU1 methods. Furthermore, we study the stability of the PR-ADI methods more rigorously by calculating the leading eigenvalues of the amplification matrix $\mathbf{M} = \mathbf{D}^{-1}\mathbf{B}$ in (29) for N_x close to the values where the PR-TAU1 method diverges. The inverse of $\mathbf{D}$ is calculated by a biconjugate gradient iterative solver in SLATEC package (http://www.netlib.org/slatec/) with the tolerance $= 10^{-7}$, and the eigenvalues are computed by the eigenvalue package ARPACK (http://www.caam.rice.edu/software/ARPACK/). Being aware of that multiple eigenvalues may be of

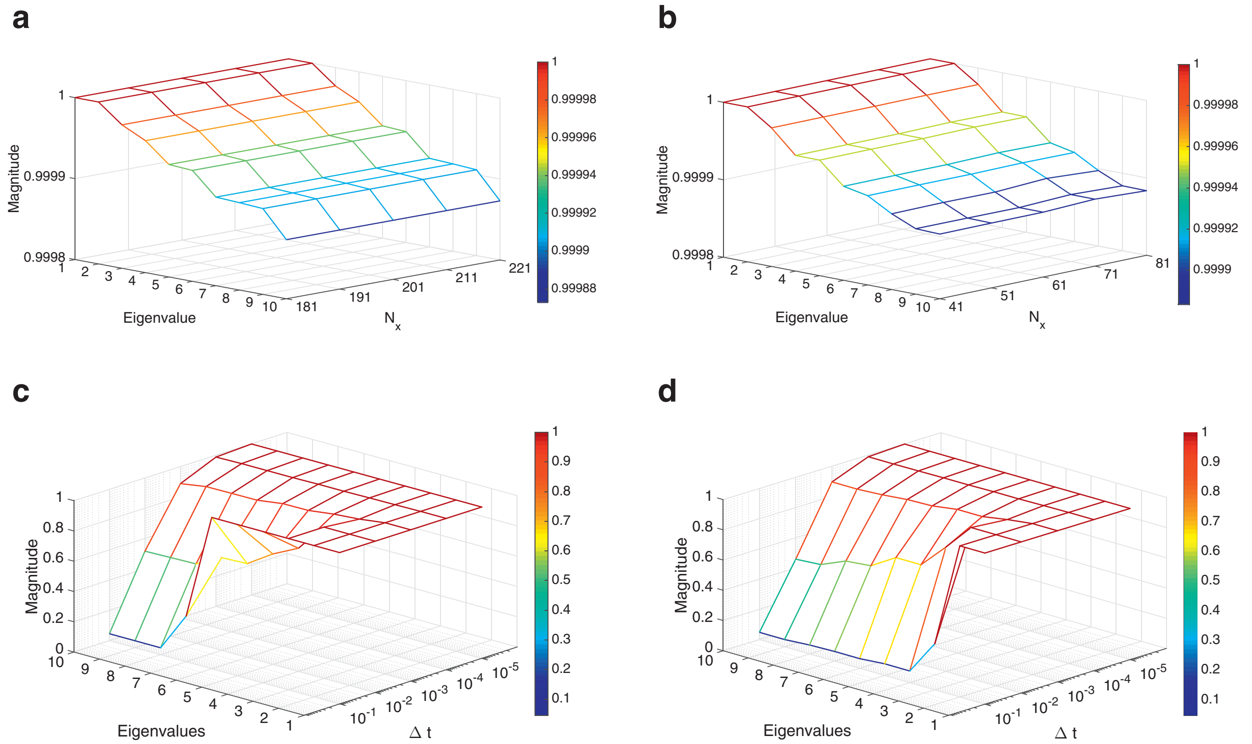

Fig. 8. Stability tests of Examples 1 and 2. In each chart, magnitudes of ten leading eigenvalues of the PR-ADI method are reported. (a) Example 1 with $N_x = 181, 191, 201, 211, 221$ and $\Delta t = 10^{-6}$; (b) Example 2 with $N_x = 41, 51, 61, 71, 81$ and $\Delta t = 10^{-6}$; (c) Example 1 with $N_x = 201$ and $\Delta t = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$; (d) Example 2 with $N_x = 51$ and $\Delta t = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$.

the same magnitude, we calculate ten leading eigenvalues with the largest magnitude. Moreover, with the knowledge that the PR-TAU1 diverges when $N_x = N_y = 201$ in Example 1 and $N_x = N_y = 51$ in Example 2, it will be interesting to calculate the leading eigenvalues for N_x taking values around 201 and 51 for above two examples, respectively. The results are shown in Fig. 8. By fixing $\Delta t = 10^{-6}$, the magnitudes of all calculated eigenvalues of PR-ADI stay very close to one. When Δt is larger, only first a few eigenvalues are close to one. The rest of eigenvalues quickly decay. We note that in all cases, the calculated eigenvalues are strictly less than one, indicating that the proposed PR-ADI scheme is unconditionally stable for solving 2D parabolic interface problems.

Example 3. We conclude the section of numerical experiments with one more study of the effect of the diffusion coefficient α on the proposed numerical methods. In this example, we consider a variable coefficient diffusion equation

$$u_t = \alpha(x, y)(u_{xx} + u_{yy}) + f \tag{45}$$

where the diffusion coefficient α is defined as a function of spatial variables

$$\alpha(x, y) = \begin{cases} e^{-(x^2+y^2)}, & \text{in } \Omega^-, \\ e^{x^2+y^2}, & \text{in } \Omega^+, \end{cases} \tag{46}$$

so that $\alpha > 0$ in the whole domain Ω. The same exact solution (37) , as well as the interface constructed by Eq. (38) with the parameter values $(m, b) = (2, 1/4)$, is reused for convenience. The source term is thereby given by

$$f(x, y, t) = \begin{cases} -\sin(kx)\cos(ky)\sin(t) + 2k^2\sin(kx)\cos(ky)e^{-x^2-y^2}\cos(t), & \text{in } \Omega^-, \\ -\cos(kx)\sin(ky)\sin(t) + 2k^2\cos(kx)\sin(ky)e^{x^2+y^2}\cos(t), & \text{in } \Omega^+. \end{cases} \tag{47}$$

Similar temporal and spatial experiments are conducted on this example. The obtained temporal convergence results are very similar to those obtained in Examples 1 and 2, and omitted here. Only the spatial convergence tests are shown in Fig. 9. For simplicity, we only consider the PR-ADI and D-ADI methods here. Both methods converge for all tested cases. However, the L_∞ errors obtained by the D-ADI method change very dramatically subject to the change of the variable diffusion coefficient α. On the other hand, the PR-ADI method does not experience such significant oscillations, especially for small h values. Thus, the spatial order of the PR-ADI is higher than that of the D-ADI in the L_∞ norm in this example.

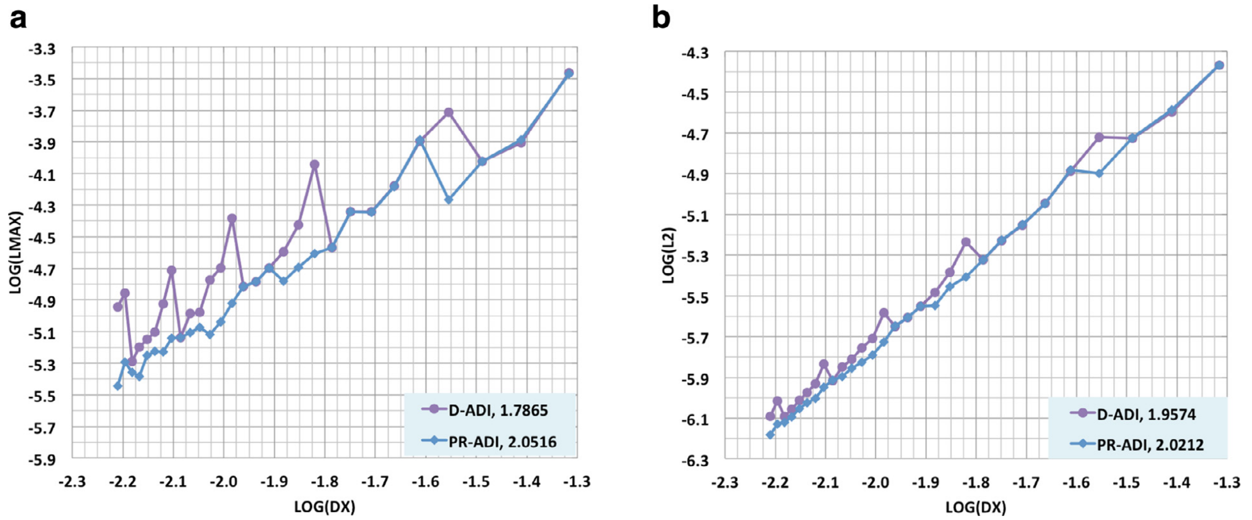

Fig. 9. Spatial convergence tests of Example 3 with fixed $\Delta t = 10^{-6}$ and various h. (a) L_∞ norm of the error; (b) L_2 norm of the error. The calculated overall convergence rates are demonstrated in the legends as well.

4. Conclusion

In this work, we developed a new matched Peaceman–Rachford alternating direction implicit (PR-ADI) scheme for solving two-dimensional (2D) parabolic interface problems. Several numerical experiments were conducted to demonstrate that the PR-ADI scheme can achieve a second order of accuracy in space for problems with curved interfaces and spatial and temporal dependent jump conditions. This is the second ADI scheme that is able to secure a spatially second order for solving material interface problems in the finite difference literature, in addition to the existing matched Douglas ADI (D-ADI) scheme [28]. In comparing with the D-ADI scheme for solving 2D heat equations, the PR-ADI scheme is always more accurate in time by using the same Δt. However, the temporal order of the PR-ADI scheme is just slightly higher than that of the D-ADI scheme. This is because in the sophisticated matched ADI procedure, a temporal approximation is inevitably introduced in the spatial approximation of the tangential derivative u_τ^+. Even though numerically the PR-ADI scheme does not significantly outperform the D-ADI scheme, there are two major theoretical breakthroughs in this work, that overcome difficulties associated with the original matched D-ADI approach [28].

First, two spatial improvements are developed in this work to provide a high fidelity approximation to the tangential derivative, i.e., approximating u_τ^+ by the auxiliary values in the alternative directions and estimating an auxiliary value by using supporting nodes from both sides of the interface. These improvements not only enhance the spatial accuracy, but also stabilize the ADI time stepping. Without such improvements, the D-ADI scheme is known to be unconditionally stable, but the PR-ADI scheme could be unstable by using the same spatial discretization [28]. This is because the stability region of the implicit Euler scheme is larger than that of the Crank–Nicolson scheme. With these two improvements, the proposed PR-ADI method becomes unconditionally stable for solving 2D parabolic interface problems, as demonstrated both numerically and through stability analysis.

Second, a new time discretization procedure is proposed. Unlike in the previous study [28], the corrections to the finite difference derivation approximations are considered in the Crank–Nicolson scheme, instead of in the split ADI scheme. This allows us to sort out various approximation terms that can be computed before the ADI splitting. By putting these terms as one general source term, the construction of the PR-ADI scheme becomes very clean. The difference between the PR-ADI and Crank–Nicolson schemes is just one higher order perturbation term. Such a time discretization procedure is certainly simpler than the previous one in Ref. [28], as well as being more accurate. Moreover, this procedure allows us to precisely trace the sources of the temporal and spatial error, and will play an important role in our subsequent numerical analysis and algorithm development.

Finally, we note that the proposed PR-ADI method is as efficient as the previous D-ADI method [28], because most of additional spatial treatments for approximating u_τ^+ are conducted at the preprocessing stage. With a little more overhead, the PR-ADI scheme scales like the D-ADI in the time stepping. In particular, by using the Thomas algorithm, the flop counts of one time step in the PR-ADI is about $O(N^2)$ for a mesh size $N = N_x = N_y$. Moreover, due to the excellent stability of the PR-ADI scheme, one can simply fix Δt to be on the order of h. Consequently, the complexity of entire time integration is on the order of $O(N^3)$ for solving a 2D heat equation with N^2 unknowns.

As can be seen in our numerical studies, the temporal order of the proposed PR-ADI scheme is about 1.5. In our future studies, it is of great interests to further improve the temporal accuracy of the matched ADI methods, while maintaining the second order in space, unconditional stability, and ultra high efficiency.

Acknowledgment

This work was supported in part by the National Science Foundation (NSF) Grant DMS-1318898 and the University of Alabama Research Stimulation Program (RSP) award.

References

[1] L. Adams, Z.L. Li, The immersed interface/multigrid methods for interface problems, SIAM J. Sci. Comput. 24 (2002) 463–479.
[2] C. Attanayake, D. Senaratne, Convergence of an immersed finite element method for semilinear parabolic interface problems, Appl. Math. Sci. 5 (2011) 135–147.
[3] P.W. Bates, Z. Chen, Y.H. Sun, G.W. Wei, S. Zhao, Geometric and potential driving formation and evolution of biomolecular surfaces, J. Math. Biol. 59 (2009) 193–231.
[4] F. Bouchon, G.H. Peichl, An immersed interface technique for the numerical solution of the heat equation on a moving domain, in: G. Kreiss, et al. (Eds.), Numerical Mathematics and Advanced Applications 2009, Springer, 2010, pp. 181–189.
[5] F. Bouchon, G.H. Peichl, The immersed interface technique for parabolic problems with mixed boundary conditions, SIAM J. Numer. Anal. 48 (2010) 2247–2266.
[6] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math. 79 (1998) 175–202.
[7] 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, J. Soc. Ind. Appl. Math. 3 (1955) 42–65.
[8] J. Douglas Jr., D. Peaceman, Numerical solution of two-dimensional heat flow problems, Am. Inst. Chem. Eng. J. 1 (1955) 505–512.
[9] B. Fornberg, Classroom note: calculation of weights in finite difference formulas, SIAM Rev. 40 (1998) 685–691.
[10] W.H. Geng, S. Zhao, Fully implicit ADI schemes for solving the nonlinear Poisson-Boltzmann equation, Mol. Based Math. Biol. 1 (2013) 109–123.
[11] G. Horton, S. Vandewalle, A space-time multigrid method for parabolic partial differential equations, SIAM J. Sci. Comput. 16 (1995) 848–864.
[12] J.D. Kandilarov, L.G. Vulkov, The immersed interface method for a nonlinear chemical diffusion equation with local sites of reactions, Numer. Algorithms 36 (2004) 285–307.
[13] J.D. Kandilarov, L.G. Vulkov, The immersed interface method for two-dimensional heat-diffusion equations with singular own sources, Appl. Numer. Math. 57 (2007) 486–497.
[14] 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.
[15] Z.L. Li, A. Mayo, ADI methods for heat equations with discontinuities along an arbitrary interface, in: W. Gautschi (Ed.), Proceedings of Symposia in Applied Mathematics, Vol. 48, AMS, 1993, pp. 311–315.
[16] Z.L. Li, Y.-Q. Shen, A numerical method for solving heat equations involving interfaces, Electron. J. Differ. Equ. Conf. 3 (1999) 100–108.
[17] J. Liu, Z. Zheng, IIM-based ADI finite difference scheme for nonlinear convection-diffusion equations with interfaces, Appl. Math. Model. 37 (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] D. Peaceman, H. Rachford, The numerical solution of parabolic and elliptic equations, J. Soc. Ind. Appl. Math. 3 (1955) 28–41.
[20] H. Press, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, Cambridge, 2007.
[21] R.K. Sinha, B. Deka, Optimal error estimates for linear parabolic problems with discontinuous coefficients, SIAM J. Numer. Anal. 43 (2005) 733–749.
[22] R.K. Sinha, B. Deka, Finite element methods for semilinear elliptic and parabolic interface problems, Appl. Numer. Math. 59 (2009) 1870–1883.
[23] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, second ed., SIAM, 2004.
[24] W. Tian, S. Zhao, A fast ADI algorithm for geometric flow equations in biomolecular surface generation, Int. J. Numer. Methods Biomed. Eng. 30 (2014) 490–516.
[25] S.Q. Wang, R. Samulyak, T.F. 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 (2010) 499–521.
[26] S. Zhao, G.W. Wei, High order FDTD methods via derivative matching for Maxwell's equations with material interfaces, J. Comput. Phys. 200 (2004) 60–103.
[27] S. Zhao, Operator splitting ADI schemes for pseudo-time coupled nonlinear solvation simulations, J. Comput. Phys. 257 (2014) 1000–1021.
[28] S. Zhao, A matched alternating direction implicit (ADI) method for solving the heat equation with interfaces, J. Sci. Comput. 63 (2015) 118–137.
[29] 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 sources, J. Comput. Phys. 213 (2006) 1–30.