

Accepted Manuscript

A dimension by dimension splitting immersed interface method for heat conduction equation with interfaces

Jiankang Liu, Zhoushun Zheng

PII: S0377-0427(13)00615-8

DOI: <http://dx.doi.org/10.1016/j.cam.2013.10.051>

Reference: CAM 9445

To appear in: *Journal of Computational and Applied Mathematics*

Received date: 14 December 2010

Revised date: 3 October 2013



Please cite this article as: J. Liu, Z. Zheng, A dimension by dimension splitting immersed interface method for heat conduction equation with interfaces, *Journal of Computational and Applied Mathematics* (2013), <http://dx.doi.org/10.1016/j.cam.2013.10.051>

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

A Dimension by Dimension Splitting Immersed Interface Method for Heat Conduction Equation with Interfaces[☆]

Jiankang Liu^{a,b,*}, Zhoushun Zheng^{b,c}

^a*School of Mathematical Sciences, Shanxi University, Taiyuan 030006, China*

^b*School of Mathematics and Statistics, Central South University, Changsha 410083, China*

^c*State Key Laboratory of Porous Metal Materials, Xi'an 710016, China*

Abstract

The numerical method proposed in this paper is an improvement of ADI method by Li [Z. Li and A. Mayo. ADI methods for heat equations with discontinuities along an arbitrary interface. Volume 48, pages 311-315. American Mathematical Society, 1994]. The proposed method is unconditionally stable for both two and three-dimensional heat conduction interface problems, while Li's ADI method is only stable for two-dimensional problem. The method is a modification of Locally One-Dimensional (LOD) difference scheme, with correction term added to the right-hand side of the standard LOD difference scheme at irregular points. The correction term is determined so that the local truncation error is of order $O(h)$ at irregular points. Then the method is two-order convergent in both time and space directions. Numerical examples show good agreement with exact solutions and confirm the order of convergence and stability.

Keywords: Splitting method, LOD scheme, Immersed interface method, Level set function

2010 MSC: 65M06

1. Introduction

Interface problems are those problems, in which discontinuous coefficients or singular sources are presented across some interfaces. The solutions of these problems are often non-smooth or even discontinuous across the interface. Hence, for interface problems, the accuracy of traditional difference scheme may be degraded because the Taylor expansions that they are based on are not valid across the interface.

In 1994, Leveque and Li [1, 2] proposed an efficient numerical method, Immersed Interface Method (IIM), for elliptic interface problems. It is a method with two-order accuracy in maximum norm. The difference scheme of IIM is the same as the standard difference scheme away from interface. When the interface passes through the difference stencil, the difference scheme should be modified by incorporating the jump condition of variables across the interface. However, in two-dimensional problems another grid point should be added to the standard five-point difference stencil, leading to a non-symmetric coefficient matrix. This problem was improved by introducing the maximum principle preserving immersed interface method [3]. It produces a diagonally dominant coefficient matrix and has been successfully implemented with a specially designed multi-grid method [4]. Explicit jump immersed interface method (EJIIM) was proposed in [5] for two-dimensional elliptic interface problems, which produces a symmetric linear system by adding proper correction terms on the right-hand side (RHS) of difference equation. The decomposed IIM [6] was proposed to handle variable coefficient elliptic interface problems. It decomposes the jump condition into each direction and produces a symmetric problem. Li proposed a fast iterative immersed interface method (FIIIM) in [7] for solving elliptic interface problems with discontinuous coefficients. The convergence analysis [8, 9] shows that the local truncation error is allowed to be of lower order along some lower dimensional manifolds without influencing the global convergence order of the solution.

For parabolic interface problems, IIM has been extended to deal with linear parabolic interface problems [4], in which the Crank-Nicolson scheme is used to treat the diffusion term. Li considered the nonlinear parabolic

[☆]This work was supported partially by National Natural Science Foundation of China under Project 51174236, the National Basic Research Program of China under Project 2011CB606306 and the Opening Project of State Key Laboratory of Porous Metal Materials under Project PMM-SKL-4-2012.

*Corresponding author

Email address: liujk189@gmail.com (Jiankang Liu)

interface problems in [10], mainly focusing on how to treat the moving interface. After that, Wiegmann and Bube [11] investigated the same nonlinear parabolic interface problems, handling the nonlinear convection term in a convenient way under the IIM strategy. The immersed interface based θ -difference scheme was proposed for two dimensional heat-diffusion equations with singular own sources [12]. A stable and fast prediction-correction implicit scheme [13] was developed to handle the thermo-elastic system with discontinuities.

For time-dependent problems, implicit schemes are popular because it can eliminate the time step restriction. One disadvantage is that a large linear system of equations need solving at each time step. The approach to overcome this drawback may be to use Alternative Direction Implicit (ADI) or splitting method, in which only a sequence of tridiagonal systems need solving. The P-R ADI scheme [14] and the LOD scheme [15, 16] are of these methods. In [17, 18], the P-R ADI scheme was applied to heat equations and Stefan problems, respectively. However, Li's modified P-R ADI scheme is only absolutely stable for two-dimensional heat conduction problems, while the LOD scheme can be extended directly from 2D to 3D without losing the merit of absolute stability [19, 20]. The intention of this paper is to apply LOD schemes to heat interface problem in 2D and 3D. The main objective is to add correction term to the RHS of standard difference scheme at irregular points, thus it will not affect the stability of standard LOD schemes. The level set function [21] is used to represent the interface due to its simplicity and strength in describing complex shapes.

The rest of this paper is organized as follows: In Section 2 we describe the problems to be solved and provide some preparations for this paper. Then, the construction and analysis of LOD-IIM in 2D are presented in Section 3. Similar extension of LOD-IIM to 3D problem is shown in Section 4. We display some numerical examples in Section 5 before a concluding remark is made in Section 6.

2. Preliminaries

2.1. Problem description

We consider the following heat conduction equation

$$\frac{\partial u}{\partial t} = \Delta u - f(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \Omega \times (0, T], \quad (2.1)$$

with initial condition,

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad (2.2)$$

and boundary condition,

$$u(\mathbf{x}, t) = g(\mathbf{x}, t), \quad (\mathbf{x}, t) \in \partial\Omega \times (0, T], \quad (2.3)$$

where Δ is the Laplace operator in 2D or 3D, with $\mathbf{x} = (x, y)$ in 2D and $\mathbf{x} = (x, y, z)$ in 3D. The source $f(\mathbf{x}, t)$ and initial value $u_0(\mathbf{x})$ may be discontinuous or singular across interface Γ . The interface Γ is immersed in the domain Ω and divides Ω into two parts, Ω^+ and Ω^- . Due to the bad initial value and source term, the solution $u(\mathbf{x}, t)$ will be discontinuous or non-smooth across the interface. Along the interface, the jump conditions $[u]$ and $[u_{\mathbf{n}}]$ can often be derived from the problem itself,

$$[u] = \lim_{\substack{\mathbf{x} \rightarrow \Gamma \\ \mathbf{x} \in \Omega^+}} u(\mathbf{x}, t) - \lim_{\substack{\mathbf{x} \rightarrow \Gamma \\ \mathbf{x} \in \Omega^-}} u(\mathbf{x}, t) = w(\mathbf{x}, t), \quad (2.4)$$

$$[u_{\mathbf{n}}] = \lim_{\substack{\mathbf{x} \rightarrow \Gamma \\ \mathbf{x} \in \Omega^+}} u_{\mathbf{n}}(\mathbf{x}, t) - \lim_{\substack{\mathbf{x} \rightarrow \Gamma \\ \mathbf{x} \in \Omega^-}} u_{\mathbf{n}}(\mathbf{x}, t) = v(\mathbf{x}, t), \quad (2.5)$$

where $\mathbf{n}$ is the unit outward normal vector, see Figure 1.

2.2. Level set function

Since the interface can have a fairly complex shape, a smooth level set function $\phi(\mathbf{x})$ is introduced as

$$\phi(\mathbf{x}) = \pm dis,$$

where dis is the shortest distance from $\mathbf{x}$ to the interface Γ . Thus the set of points with $\phi(\mathbf{x}) = 0$ represents the interface Γ ,

$$\Gamma = \{\mathbf{x} \in \mathbb{R}^{dim} \mid \phi(\mathbf{x}) = 0\}, \quad dim = 2, 3. \quad (2.6)$$

The set of all points with $\phi(\mathbf{x}) < 0$ and the set of all points with $\phi(\mathbf{x}) > 0$ represent two disjoint domains Ω^- and Ω^+ , respectively. The unit outward normal vector $\mathbf{n}$ can be approximated by $\mathbf{n} = \nabla\phi/|\nabla\phi|$ at any point.

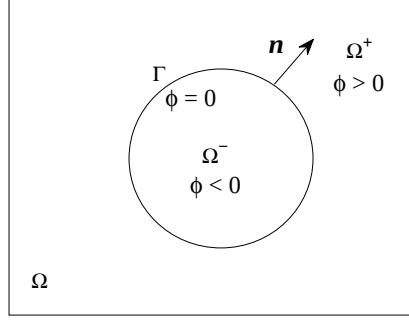


Figure 1: The diagram of domain, interface and unit outward normal vector $\mathbf{n}$

In favor of level set function $\phi(x, y)$, we can determine the locations of intersection between interface and grid lines. Suppose that the interface lies between (x_i, y_j) and (x_{i+1}, y_j) , if we define

$$\theta_x = \frac{\phi_{i,j}}{\phi_{i,j} - \phi_{i+1,j}},$$

then $0 \leq \theta_x \leq 1$ and the interface is located at

$$(x^*, y^*) = (x_i + \theta_x h_x, y_j), \quad h_x = x_{i+1} - x_i. \quad (2.7)$$

Similarly, when the interface lies between (x_i, y_j) and (x_i, y_{j+1}) , we can define

$$\theta_y = \frac{\phi_{i,j}}{\phi_{i,j} - \phi_{i,j+1}},$$

thus $0 \leq \theta_y \leq 1$ and the interface is located at

$$(x^*, y^*) = (x_i, y_j + \theta_y h_y), \quad h_y = y_{j+1} - y_j. \quad (2.8)$$

2.3. Corrected central difference schemes

In this subsection, we assume that the domain Ω is a rectangular, $[a, b] \times [c, d]$, and the interface Γ is a circle immersed in Ω . The domain is discretized uniformly by a $M \times N$ grid with grid coordinates

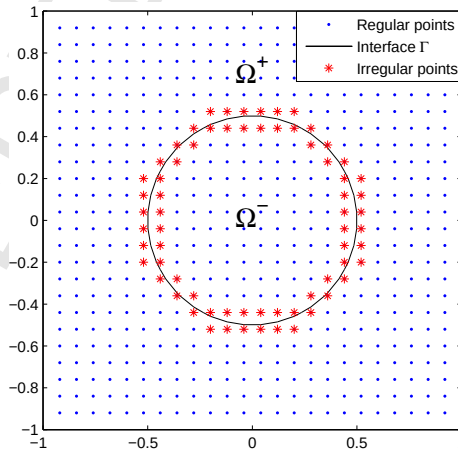


Figure 2: Mesh grid of 25×25 subdivision with reference to five-point stencil

$$x_i = a + (i - 1)h_x, \quad y_j = c + (j - 1)h_y, \quad i = 1, \dots, M + 1, \quad j = 1, \dots, N + 1,$$

where $h_x = (b - a)/M$, $h_y = (d - c)/N$ are space steps in x - and y -axis directions, respectively. Thus the interface is not always align with the mesh grids. A grid point (x_i, y_j) is defined as regular if all the points in the difference stencil are on the same side of the interface. On the contrary, if at least one point is on the other side of the interface it is irregular. In Figure 2, the mesh grid of 25×25 subdivision with reference to five-point stencil is displayed, with the solid circle being the interface Γ , the “*” representing irregular points and the “•” the regular points. Assume that $u_{i,j}^n$ represents the value of $u(x_i, y_j, t_n)$ and $U_{i,j}^n$ is the numerical approximation of $u(x_i, y_j, t_n)$, where t_n is the n -th time level.

The second-derivative u_{xx} and u_{yy} are approximated by central difference schemes. For smooth solutions, the central difference scheme is of second-order accuracy. However, when the solution is non-smooth or discontinuous across the interface, the accuracy at irregular points will be degraded. Thus we should add correction terms to the standard difference schemes at irregular points to improve the accuracy. The following lemma, analogous to Lemma 3 in [5], is the basis for determining the correction term in Section 3 and Section 4.

Lemma 1. *Suppose that the interface lies between (x_i, y_j) and (x_{i+1}, y_j) , located at (x^*, y^*) , and $u \in C^3(\Omega^- \cup \Omega^+)$, with derivatives extending continuously up to the boundary Γ . Then the following approximations hold to $O(h)$:*

$$(u_{xx})_{i,j} \approx \delta_{xx} u_{i,j} - (C_{x2})_{i,j}, \quad (u_{xx})_{i+1,j} \approx \delta_{xx} u_{i+1,j} - (C_{x2})_{i+1,j},$$

if $(x_i, y_j) \in \Omega^-$ and $(x_{i+1}, y_j) \in \Omega^+$, where

$$(C_{x2})_{i,j} = \frac{1}{h^2} \left([u] + [u_x](x_{i+1} - x^*) + \frac{1}{2}[u_{xx}](x_{i+1} - x^*)^2 \right),$$

$$(C_{x2})_{i+1,j} = -\frac{1}{h^2} \left([u] + [u_x](x_i - x^*) + \frac{1}{2}[u_{xx}](x_i - x^*)^2 \right),$$

and (x^*, y^*) can be obtained by (2.7).

According to different relative position between the interface and irregular points, the correction term C_{x2} is different.

Remark 1. *If the interface lies between (x_i, y_j) and (x_{i+1}, y_j) , $(x_i, y_j) \in \Omega^+$ and $(x_{i+1}, y_j) \in \Omega^-$, then the correction term will be*

$$(C_{x2})_{i,j} = -\frac{1}{h^2} \left([u] + [u_x](x_{i+1} - x^*) + \frac{1}{2}[u_{xx}](x_{i+1} - x^*)^2 \right),$$

$$(C_{x2})_{i+1,j} = \frac{1}{h^2} \left([u] + [u_x](x_i - x^*) + \frac{1}{2}[u_{xx}](x_i - x^*)^2 \right).$$

If the interface lies on the grid point, then we should decide from which side the limit value of primary variables at this point will approach. Without losing the generality, we assume that the limit value approaches from “ Ω^+ ” direction.

Remark 2. *If the interface lies on the grid point (x_i, y_j) , $(x_{i-1}, y_j) \in \Omega^+$ and $(x_{i+1}, y_j) \in \Omega^-$, then the correction term will be*

$$(C_{x2})_{i,j} = -\frac{1}{h^2}[u] - \frac{1}{h}[u_x] - \frac{1}{2}[u_{xx}], \quad (C_{x2})_{i+1,j} = \frac{1}{h^2}[u] - [u_{xx}].$$

If the interface lies on the grid point (x_i, y_j) , $(x_{i-1}, y_j) \in \Omega^-$ and $(x_{i+1}, y_j) \in \Omega^+$, then the correction term will be

$$(C_{x2})_{i,j} = -\frac{1}{h^2}[u] + \frac{1}{h}[u_x] - \frac{1}{2}[u_{xx}], \quad (C_{x2})_{i-1,j} = \frac{1}{h^2}[u] - [u_{xx}].$$

Remark 3. *The correction terms C_{y2s} in y -axis direction can be determined analogously to Lemma 1, Remark 1 and Remark 2.*

With the help of these correction terms, we can approximate the second derivatives to be of local truncation error $O(h)$ at irregular points.

3. Numerical methods

In this section, we will construct the LOD-IIM scheme for heat interface problem (2.1)-(2.3) in 2D.

3.1. The idea of LOD-IIM

For heat equation (2.1) in 2D, it can be discretized by LOD scheme as,

$$(I - \frac{\Delta t}{2}\delta_{xx})U^{n+\frac{1}{2}} = (I + \frac{\Delta t}{2}\delta_{xx})U^n - \frac{\Delta t}{2}(I + \frac{\Delta t}{2}\delta_{xx})f^{n+\frac{1}{2}}, \quad (3.9)$$

$$(I - \frac{\Delta t}{2}\delta_{yy})U^{n+1} = (I + \frac{\Delta t}{2}\delta_{yy})U^{n+\frac{1}{2}} - \frac{\Delta t}{2}(I - \frac{\Delta t}{2}\delta_{yy})f^{n+\frac{1}{2}}, \quad (3.10)$$

where Δt is the time step, I the identity operator. The truncation error of magnitude $O(\Delta t^2 + h^2)$ can be achieved for smooth problems, while for interface problems it will be degraded. In order to improve the accuracy of LOD scheme for interface problems, we need to add some correction terms to the RHS of LOD difference schemes. We add one correction term on the RHS of equation (3.9), then (3.9) becomes,

$$(I - \frac{\Delta t}{2}\delta_{xx})U^{n+\frac{1}{2}} = (I + \frac{\Delta t}{2}\delta_{xx})U^n - \frac{\Delta t}{2}(I + \frac{\Delta t}{2}\delta_{xx})f^{n+\frac{1}{2}} - \Delta t C. \quad (3.11)$$

We call (3.11) and (3.10) the LOD-IIM schemes for 2D interface problem (2.1)-(2.3). Here and from now on, we suppress the subscript of space in our notation and assume that all terms are computed at (x_i, y_j) if it can not cause any confusions.

3.2. Local truncation error analysis

In this subsection, we will discuss how to determine the correction terms C by local truncation error analysis. In order to achieve two-order convergence in maximum norm, we should make the local truncation error to be of order $O(h)$ at irregular points and of $O(h^2)$ at regular points. At regular points it is natural to satisfy local truncation error of order $O(h^2)$ due to the central difference schemes, while at irregular points we should correct it so that the local truncation error is of order $O(h)$.

Applying the operators $(I - \frac{\Delta t}{2}\delta_{xx})$ and $(I + \frac{\Delta t}{2}\delta_{yy})$ to equations (3.10) and (3.11), respectively, after some manipulations we can get the actual finite difference scheme from t_n to t_{n+1} ,

$$\frac{U^{n+1} - U^n}{\Delta t} = M - f^{n+\frac{1}{2}} - \frac{\Delta t^2}{4}\delta_{xx}\delta_{yy}f^{n+\frac{1}{2}} - (I + \frac{\Delta t}{2}\delta_{yy})C, \quad (3.12)$$

where

$$M = \frac{1}{2}\delta_{xx}(U^{n+1} + U^n) + \frac{1}{2}\delta_{yy}(U^{n+1} + U^n) - \frac{\Delta t}{4}\delta_{xx}\delta_{yy}(U^{n+1} - U^n).$$

Since the interface is fixed and all the quantities are continuous with time, we can check the local truncation error of difference scheme (3.12) by examining each term. The left hand side in (3.12), with U substituted by the exact solution u , yields

$$\frac{u^{n+1} - u^n}{\Delta t} = u_t^{n+\frac{1}{2}} + O(\Delta t^2).$$

Keeping in mind the fact that

$$u_{xx} = \delta_{xx}u - C_{x2} + O(h), \quad u_{yy} = \delta_{yy}u - C_{y2} + O(h),$$

the term M on the RHS of (3.12) can be corrected as, with U substituted by u ,

$$\begin{aligned}
 & \frac{1}{2}\delta_{xx}(u^{n+1} + u^n) + \frac{1}{2}\delta_{yy}(u^{n+1} + u^n) - \frac{\Delta t}{4}\delta_{xx}\delta_{yy}(u^{n+1} - u^n) \\
 = & \frac{1}{2}(u_{xx}^{n+1} + u_{xx}^n) + \frac{1}{2}(u_{yy}^{n+1} + u_{yy}^n) + \frac{1}{2}(C_{x2}^{n+1} + C_{x2}^n) + \frac{1}{2}(C_{y2}^{n+1} + C_{y2}^n) \\
 & - \frac{\Delta t}{4}\delta_{xx}(u_{yy}^{n+1} - u_{yy}^n + C_{y2}^{n+1} - C_{y2}^n + O(h)) + O(h) \\
 = & \frac{1}{2}(u_{xx}^{n+1} + u_{xx}^n) + \frac{1}{2}(u_{yy}^{n+1} + u_{yy}^n) + \frac{1}{2}(C_{x2}^{n+1} + C_{x2}^n) + \frac{1}{2}(C_{y2}^{n+1} + C_{y2}^n) \\
 & - \frac{\Delta t}{4}(C_{y2x0}^{n+1} - C_{y2x0}^n + O(h^{-1})) - \frac{\Delta t}{4}\delta_{xx}(C_{y2}^{n+1} - C_{y2}^n) + O(h) \tag{3.13}
 \end{aligned}$$

$$\begin{aligned}
 = & u_{xx}^{n+\frac{1}{2}} + u_{yy}^{n+\frac{1}{2}} + \frac{1}{2}(C_{x2}^{n+1} + C_{x2}^n) + \frac{1}{2}(C_{y2}^{n+1} + C_{y2}^n) \\
 & - \frac{\Delta t}{4}(C_{y2x0}^{n+1} - C_{y2x0}^n) - \frac{\Delta t}{4}\delta_{xx}(C_{y2}^{n+1} - C_{y2}^n) + O(h + h^{-1}\Delta t^2) \tag{3.14}
 \end{aligned}$$

$$= u_{xx}^{n+\frac{1}{2}} + u_{yy}^{n+\frac{1}{2}} + F + O(h + h^{-1}\Delta t^2), \tag{3.15}$$

where

$$F = \frac{1}{2}(C_{x2}^{n+1} + C_{x2}^n) + \frac{1}{2}(C_{y2}^{n+1} + C_{y2}^n) - \frac{\Delta t}{4}(C_{y2x0}^{n+1} - C_{y2x0}^n) - \frac{\Delta t}{4}\delta_{xx}(C_{y2}^{n+1} - C_{y2}^n),$$

in which, $C_{y2x0}^n = \pm \frac{1}{h^2}[u_{yy}](x^*, y_j, t_n)$, satisfying $\delta_{xx}u_{yy}^n = C_{y2x0}^n + O(h^{-1})$. Since all quantities are continuous with time, the quantity $O(h^{-1})$ in (3.13) will become $O(h^{-1}\Delta t)$ when we expand all quantities at time level $t_{n+\frac{1}{2}}$, thus (3.14) holds. Combining (3.12) and (3.15), we get

$$(I + \frac{\Delta t}{2}\delta_{yy})C = F - \frac{\Delta t^2}{4}\delta_{xx}\delta_{yy}f^{n+\frac{1}{2}}, \tag{3.16}$$

which is the tri-diagonal system for correction term C at each time step. The tri-diagonal systems arising from the operator splitting and the computation of correction term can be solved by Thomas method [20] efficiently.

3.3. Computation of jump conditions

The formula of correction term F in (3.16) is the combination of jump conditions of u and its derivatives. Since the jump conditions are usually defined in normal or tangential direction by the problem itself, we need to define a local coordinate system at (x^*, y^*) on the interface

$$\begin{aligned}
 \xi &= (x - x^*)\cos\theta + (y - y^*)\sin\theta, \\
 \eta &= -(x - x^*)\sin\theta + (y - y^*)\cos\theta,
 \end{aligned} \tag{3.17}$$

where θ is the angle between the x - and ξ -axis. The ξ -axis is normal to the interface while the η -axis is tangential, see Figure 3. We can prove that 2D equation (2.1) under the local coordinate (3.17) is unchanged.

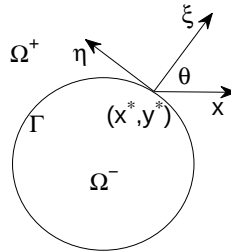


Figure 3: The diagram of local coordinate system

In the neighborhood of point (x^*, y^*) , we parameterize Γ locally by $\xi = \chi(\eta)$, $\eta = \eta$. Then, we have $\chi(0) = 0$, $\chi'(0) = 0$ provided the interface is smooth at (x^*, y^*) , with $\chi''(0)$ being the curvature at (x^*, y^*) .

Suppose that the jump conditions (2.4) and (2.5) are only defined on the interface, then we can obtain required jump conditions under the local coordinate system,

$$[u] = w(\eta, t), \quad [u_\xi] = [u_n] = v(\eta, t), \quad [u_\eta] = [u]_\eta = w_\eta(\eta, t),$$

$$[u_{\eta\eta}] = -\chi''[u_\xi] + w_{\eta\eta}, \quad [u_{\xi\eta}] = \chi''[u_\eta] + v_\eta, \quad [u_{\xi\xi}] = \chi''[u_\xi] - w_{\eta\eta} + [f] + w_t,$$

which are similar to that in [17, 22]. Thus the jump conditions in the x - and y -directions are given by following formulas,

$$[u_x] = [u_\xi] \cos \theta - [u_\eta] \sin \theta, \quad [u_{xx}] = [u_{\xi\xi}] \cos^2 \theta - 2[u_{\xi\eta}] \cos \theta \sin \theta + [u_{\eta\eta}] \sin^2 \theta,$$

$$[u_y] = [u_\xi] \sin \theta + [u_\eta] \cos \theta, \quad [u_{yy}] = [u_{\xi\xi}] \sin^2 \theta + 2[u_{\xi\eta}] \cos \theta \sin \theta + [u_{\eta\eta}] \cos^2 \theta.$$

4. Extend LOD-IIM to 3D

In this section, we extend the LOD-IIM scheme to heat interface problem (2.1)-(2.3) in 3D.

4.1. The LOD-IIM in 3D

For 3D heat conduction equation (2.1), the LOD scheme reads

$$(I - \frac{\Delta t}{2} \delta_{xx}) U^{n+\frac{1}{3}} = (I + \frac{\Delta t}{2} \delta_{xx}) U^n - \Delta t f^{n+\frac{1}{2}}, \quad (4.18)$$

$$(I - \frac{\Delta t}{2} \delta_{yy}) U^{n+\frac{2}{3}} = (I + \frac{\Delta t}{2} \delta_{yy}) U^{n+\frac{1}{3}} + \Delta t f^{n+\frac{1}{2}}, \quad (4.19)$$

$$(I - \frac{\Delta t}{2} \delta_{zz}) U^{n+1} = (I + \frac{\Delta t}{2} \delta_{zz}) U^{n+\frac{2}{3}} - \Delta t f^{n+\frac{1}{2}}, \quad (4.20)$$

which is unconditionally stable and the local truncation error is of $O(\Delta t^2 + h^2)$ for smooth problems. For interface problems, we also should correct the LOD scheme. We add one correction term C to the RHS of (4.18), thus the equation (4.18) becoms

$$(I - \frac{\Delta t}{2} \delta_{xx}) U^{n+\frac{1}{3}} = (I + \frac{\Delta t}{2} \delta_{xx}) U^n - \Delta t f^{n+\frac{1}{2}} - \Delta t C. \quad (4.21)$$

Equations (4.21), (4.19) and (4.20) are the LOD-IIM scheme for 3D heat interface problems. Here we suppress the subscript of space and assume that all terms are computed at (x_i, y_j, z_k) .

4.2. Local truncation error analysis

Eliminating intermediate terms $U^{n+\frac{1}{3}}$ and $U^{n+\frac{2}{3}}$ in (4.19)-(4.21) and after some manipulations, we get the actual difference scheme for U^{n+1} ,

$$\frac{U^{n+1} - U^n}{\Delta t} = M_1 + M_2 + M_3 + M_4 - f^{n+\frac{1}{2}} - (I + \frac{\Delta t}{2} \delta_{yy})(I + \frac{\Delta t}{2} \delta_{zz})C, \quad (4.22)$$

where

$$M_1 = \frac{1}{2} \delta_{xx} (U^n + U^{n+1}) + \frac{1}{2} \delta_{yy} (U^n + U^{n+1}) + \frac{1}{2} \delta_{zz} (U^n + U^{n+1}),$$

$$M_2 = -\frac{\Delta t}{4} \delta_{xx} \delta_{yy} (U^{n+1} - U^n) - \frac{\Delta t}{4} \delta_{xx} \delta_{zz} (U^{n+1} - U^n) - \frac{\Delta t}{4} \delta_{yy} \delta_{zz} (U^{n+1} - U^n),$$

$$M_3 = \frac{\Delta t^2}{8} \delta_{xx} \delta_{yy} \delta_{zz} (U^n + U^{n+1}), \quad M_4 = -\frac{\Delta t^2}{4} \delta_{xx} \delta_{yy} f^{n+\frac{1}{2}} - \frac{\Delta t^2}{4} \delta_{xx} \delta_{zz} f^{n+\frac{1}{2}} - \frac{\Delta t^2}{4} \delta_{yy} \delta_{zz} f^{n+\frac{1}{2}}.$$

The terms M_1 , M_2 and M_3 need correcting, while M_4 should be cancelled out by C .

Since the primary variables are continuous with time, we check the local truncation error of (4.22) by each term, with U substituted by u . For the first part M_1 , we get

$$\begin{aligned} & \frac{1}{2}\delta_{xx}(u^n + u^{n+1}) + \frac{1}{2}\delta_{yy}(u^n + u^{n+1}) + \frac{1}{2}\delta_{zz}(u^n + u^{n+1}) \\ &= \frac{1}{2}(u_{xx}^n + u_{xx}^{n+1}) + \frac{1}{2}(u_{yy}^n + u_{yy}^{n+1}) + \frac{1}{2}(u_{zz}^n + u_{zz}^{n+1}) + C_1 + O(h) \\ &= u_{xx}^{n+\frac{1}{2}} + u_{xx}^{n+\frac{1}{2}} + u_{xx}^{n+\frac{1}{2}} + C_1 + O(h), \end{aligned} \quad (4.23)$$

where

$$C_1 = \frac{1}{2}(C_{x2}^n + C_{x2}^{n+1}) + \frac{1}{2}(C_{y2}^n + C_{y2}^{n+1}) + \frac{1}{2}(C_{z2}^n + C_{z2}^{n+1}).$$

Similar to (3.15), M_2 can be corrected by

$$-\frac{\Delta t}{4}\delta_{xx}\delta_{yy}(u^{n+1} - u^n) - \frac{\Delta t}{4}\delta_{xx}\delta_{zz}(u^{n+1} - u^n) - \frac{\Delta t}{4}\delta_{yy}\delta_{zz}(u^{n+1} - u^n) = C_2 + O(h + h^{-1}\Delta t^2), \quad (4.24)$$

where

$$\begin{aligned} C_2 &= -\frac{\Delta t}{4}(C_{y2x0}^{n+1} - C_{y2x0}^n + \delta_{xx}(C_{y2}^{n+1} - C_{y2}^n) + C_{z2x0}^{n+1} - C_{z2x0}^n \\ &\quad + \delta_{xx}(C_{z2}^{n+1} - C_{z2}^n) + C_{z2y0}^{n+1} - C_{z2y0}^n + \delta_{yy}(C_{z2}^{n+1} - C_{z2}^n)). \end{aligned}$$

In term M_3 , if u is discontinuous across interface Γ , it will be a quantity of $O(h^{-6}\Delta t^2)$. In order to acquire a local truncation error of order $O(h)$, we should correct $\delta_{xx}\delta_{yy}\delta_{zz}(u^n + u^{n+1})$ to be of order $O(h^{-1})$. For convenience, we consider the following formulas,

$$\begin{aligned} \delta_{xx}\delta_{yy}\delta_{zz}u &= \delta_{xx}\delta_{yy}(u_{zz} + \frac{h^2}{12}u_{zzzz} + C_{z4} + O(h^3)) \\ &= \delta_{xx}\delta_{yy}u_{zz} + \frac{h^2}{12}\delta_{xx}\delta_{yy}u_{zzzz} + \delta_{xx}\delta_{yy}C_{z4} + O(h^{-1}) \\ &= \delta_{xx}(u_{zzyy} + C_{z2y2} + O(h)) + \frac{h^2}{12}\delta_{xx}(C_{z4y0} + O(h^{-1})) + \delta_{xx}\delta_{yy}C_{z4} + O(h^{-1}) \\ &= C_{z2y2x0} + \delta_{xx}C_{z2y2} + \frac{h^2}{12}\delta_{xx}C_{z4y0} + \delta_{xx}\delta_{yy}C_{z4} + O(h^{-1}), \end{aligned}$$

where

$$\begin{aligned} C_{z4} &= \pm \frac{1}{h^2}([u] + [u_z](z_{i\pm 1} - z^*) + \frac{1}{2}[u_{zz}](z_{i\pm 1} - z^*)^2 + \frac{1}{6}[u_{zzz}](z_{i\pm 1} - z^*)^3 + \frac{1}{24}[u_{zzzz}](z_{i\pm 1} - z^*)^4), \\ C_{z2y2} &= \pm \frac{1}{h^2}([u_{zz}] + [u_{zzy}](y_{i\pm 1} - y^*) + \frac{1}{2}[u_{zzyy}](y_{i\pm 1} - y^*)^2), \\ C_{z4y0} &= \pm \frac{1}{h^2}[u_{zzzz}](x_i, y^*, z_k), \quad C_{z2y2x0} = \pm \frac{1}{h^2}[u_{zzyy}](x^*, y_j, z_k). \end{aligned}$$

The “ $\pm$ ” sign is determined by the the relative position between interface and irregular points. Thus the correction term C_3 for M_3 should be

$$\frac{\Delta t^2}{8}\delta_{xx}\delta_{yy}\delta_{zz}(u^n + u^{n+1}) = C_3 + O(h^{-1}\Delta t^2), \quad (4.25)$$

where

$$\begin{aligned} C_3 &= \frac{\Delta t^2}{8}(C_{z2y2x0}^n + \delta_{xx}C_{z2y2}^n + \frac{h^2}{12}\delta_{xx}C_{z4y0}^n + \delta_{xx}\delta_{yy}C_{z4}^n \\ &\quad + C_{z2y2x0}^{n+1} + \delta_{xx}C_{z2y2}^{n+1} + \frac{h^2}{12}\delta_{xx}C_{z4y0}^{n+1} + \delta_{xx}\delta_{yy}C_{z4}^{n+1}). \end{aligned}$$

Combining (4.22), (4.23), (4.24) and (4.25), we can acquire the linear system for correction term C in (4.22) at each time level,

$$(I + \frac{\Delta t}{2}\delta_{yy})(I + \frac{\Delta t}{2}\delta_{zz})C = C_1 + C_2 + C_3 + M_4.$$

4.3. Computation of jump conditions

The jump conditions in 3D can be acquired similarly to that in [23]. At a point (x^*, y^*, z^*) on the interface, we need to define the following local coordinate transformation to simplify the derivation of jump relations,

$$\begin{aligned}\xi &= \cos \beta \cos \alpha (x - x^*) - \cos \beta \sin \alpha (y - y^*) - \sin \beta (z - z^*), \\ \eta &= \sin \alpha (x - x^*) + \cos \alpha (y - y^*), \\ \zeta &= \sin \beta \cos \alpha (x - x^*) - \sin \beta \sin \alpha (y - y^*) + \cos \beta (z - z^*),\end{aligned}\tag{4.26}$$

where α is the angle between x -axis and the projection of outward normal direction on xOy plane, β is the angle between the outward normal direction and xOy plane. The local coordinate system (4.26) transforms (x, y, z) to (ξ, η, ζ) , with ξ paralleling to the normal direction of the interface pointing toward the + side. The η - and ζ - axes are in the tangent plane passing through (x^*, y^*, z^*) . Similar to that in [2], the equation (2.1) in 3D is unchanged in the local coordinate. In the neighborhood of this point, the interface can be expressed as

$$\xi = \chi(\eta, \zeta), \quad \text{with} \quad \chi(0, 0) = 0, \quad \chi_\eta(0, 0) = 0, \quad \chi_\zeta(0, 0) = 0.$$

The jump conditions (2.4) and (2.5) are only defined on the interface, thus the jump relations in the first derivative under the local coordinate read,

$$[u_\xi] = [u_n] = v(\eta, \zeta, t), \quad [u_\eta] = w_\eta(\eta, \zeta, t), \quad [u_\zeta] = w_\zeta(\eta, \zeta, t).$$

The second derivative jump conditions give,

$$\begin{aligned}[u_{\eta\eta}] &= w_{\eta\eta} - [u_\xi]\chi_{\eta\eta}, \quad [u_{\eta\zeta}] = w_{\eta\zeta} - [u_\xi]\chi_{\eta\zeta}, \quad [u_{\zeta\zeta}] = w_{\zeta\zeta} - [u_\xi]\chi_{\zeta\zeta}, \\ [u_{\xi\eta}] &= v_\eta + [u_\eta]\chi_{\eta\eta} + [u_\zeta]\chi_{\zeta\eta}, \quad [u_{\xi\zeta}] = v_\zeta + [u_\eta]\chi_{\eta\zeta} + [u_\zeta]\chi_{\zeta\zeta}, \quad [u_{\xi\xi}] = w_t - [u_{\eta\eta}] - [u_{\zeta\zeta}] + [f].\end{aligned}$$

The third derivative jump conditions read,

$$\begin{aligned}[u_{\eta\eta\eta}] &= w_{\eta\eta\eta} - 3\chi_{\eta\eta}[u_{\xi\eta}] - \chi_{\eta\eta\eta}[u_\xi], \quad [u_{\eta\eta\zeta}] = w_{\eta\eta\zeta} - 2\chi_{\eta\zeta}[u_{\xi\eta}] - \chi_{\eta\eta}[u_{\xi\zeta}] - \chi_{\eta\eta\zeta}[u_\xi], \\ [u_{\zeta\zeta\zeta}] &= w_{\zeta\zeta\zeta} - 3\chi_{\zeta\zeta}[u_{\xi\zeta}] - \chi_{\zeta\zeta\zeta}[u_\xi], \quad [u_{\eta\zeta\zeta}] = w_{\eta\zeta\zeta} - 2\chi_{\eta\zeta}[u_{\xi\zeta}] - \chi_{\zeta\zeta}[u_{\xi\eta}] - \chi_{\eta\zeta\zeta}[u_\xi], \\ [u_{\xi\eta\eta}] &= v_{\eta\eta} + (\chi_{\eta\eta}^2 + \chi_{\eta\zeta}^2)v - \chi_{\eta\eta}[u_{\xi\xi}] + 2\chi_{\eta\zeta}[u_{\eta\zeta}] + 2\chi_{\eta\eta}[u_{\eta\eta}] + \chi_{\eta\eta\eta}[u_\eta] + \chi_{\eta\eta\zeta}[u_\zeta], \\ [u_{\xi\zeta\zeta}] &= v_{\zeta\zeta} + (\chi_{\eta\zeta}^2 + \chi_{\zeta\zeta}^2)v - \chi_{\zeta\zeta}[u_{\xi\xi}] + 2\chi_{\eta\zeta}[u_{\eta\zeta}] + 2\chi_{\zeta\zeta}[u_{\zeta\zeta}] + \chi_{\eta\zeta\zeta}[u_\eta] + \chi_{\zeta\zeta\zeta}[u_\zeta], \\ [u_{\xi\eta\zeta}] &= v_{\eta\zeta} + (\chi_{\eta\eta} + \chi_{\zeta\zeta})\chi_{\eta\zeta}v - \chi_{\eta\zeta}[u_{\xi\xi}] + (\chi_{\eta\eta} + \chi_{\zeta\zeta})[u_{\eta\zeta}] + \chi_{\eta\zeta}([u_{\eta\eta}] + [u_{\zeta\zeta}]) + \chi_{\eta\eta\zeta}[u_\eta] + \chi_{\eta\zeta\zeta}[u_\zeta], \\ [u_{\xi\xi\eta}] &= w_{t\eta} - [u_{\eta\eta\eta}] - [u_{\eta\zeta\zeta}] + [f_\eta], \quad [u_{\xi\xi\zeta}] = w_{t\zeta} - [u_{\eta\eta\zeta}] - [u_{\zeta\zeta\zeta}] + [f_\zeta], \\ [u_{\xi\xi\xi}] &= [u_{t\xi}] - [u_{\xi\eta\eta}] - [u_{\xi\zeta\zeta}] + [f_\xi].\end{aligned}$$

The fourth derivative jump conditions read,

$$\begin{aligned}[u_{\eta\eta\eta\eta}] &= w_{\eta\eta\eta\eta} - 6\chi_{\eta\eta}[u_{\xi\eta\eta}] - 3\chi_{\eta\eta}^2[u_{\xi\xi}] - 4\chi_{\eta\eta\eta}[u_{\xi\eta}] - \chi_{\eta\eta\eta\eta}[u_\xi], \\ [u_{\zeta\zeta\zeta\zeta}] &= w_{\zeta\zeta\zeta\zeta} - 6\chi_{\zeta\zeta}[u_{\xi\zeta\zeta}] - 3\chi_{\zeta\zeta}^2[u_{\xi\xi}] - 4\chi_{\zeta\zeta\zeta}[u_{\xi\zeta}] - \chi_{\zeta\zeta\zeta\zeta}[u_\xi], \\ [u_{\eta\eta\eta\zeta}] &= w_{\eta\eta\eta\zeta} - 3\chi_{\eta\zeta}[u_{\xi\eta\eta}] - 3\chi_{\eta\eta}[u_{\xi\eta\zeta}] - 3\chi_{\eta\eta\eta}[u_{\xi\xi}] - 3\chi_{\eta\eta\zeta}[u_{\xi\eta}] - \chi_{\eta\eta\eta\eta}[u_{\xi\zeta}] - \chi_{\eta\eta\eta\zeta}[u_\xi], \\ [u_{\eta\zeta\zeta\zeta}] &= w_{\eta\zeta\zeta\zeta} - 3\chi_{\eta\zeta}[u_{\xi\zeta\zeta}] - 3\chi_{\zeta\zeta}[u_{\xi\eta\zeta}] - 3\chi_{\zeta\zeta}\chi_{\eta\zeta}[u_{\xi\xi}] - 3\chi_{\eta\zeta\zeta}[u_{\xi\zeta}] - \chi_{\zeta\zeta\zeta}[u_{\xi\eta}] - \chi_{\eta\zeta\zeta\zeta}[u_\xi], \\ [u_{\eta\eta\zeta\zeta}] &= w_{\eta\eta\zeta\zeta} - \chi_{\zeta\zeta}[u_{\xi\eta\eta}] - 4\chi_{\eta\zeta}[u_{\xi\eta\zeta}] - \chi_{\eta\eta}[u_{\xi\zeta\zeta}] - (2\chi_{\eta\zeta}^2 + \chi_{\eta\eta}\chi_{\zeta\zeta})[u_{\xi\xi}] \\ &\quad - 2\chi_{\eta\zeta\zeta}[u_{\xi\eta}] - 2\chi_{\eta\eta\zeta}[u_{\xi\zeta}] - \chi_{\eta\eta\zeta\zeta}[u_\xi], \\ [u_{\xi\eta\eta\eta}] &= v_{\eta\eta\eta} + 3(\chi_{\eta\eta}^2 + \chi_{\eta\zeta}^2)v_\eta + 3(\chi_{\eta\zeta}\chi_{\eta\eta\zeta} + \chi_{\eta\eta}\chi_{\eta\eta\eta})v - 3\chi_{\eta\eta}[u_{\xi\xi\eta}] + 3\chi_{\eta\eta}[u_{\eta\eta\eta}] + 3\chi_{\eta\zeta}[u_{\eta\eta\zeta}] \\ &\quad - \chi_{\eta\eta\eta\eta}[u_{\xi\xi}] + 3\chi_{\eta\eta}^2[u_{\xi\eta}] + 3\chi_{\eta\zeta}\chi_{\eta\eta}[u_{\xi\zeta}] + 3\chi_{\eta\eta\eta}[u_{\eta\eta}] + 3\chi_{\eta\eta\zeta}[u_{\eta\zeta}] + \chi_{\eta\eta\eta\eta}[u_\eta] + \chi_{\eta\eta\eta\zeta}[u_\zeta], \\ [u_{\xi\zeta\zeta\zeta}] &= v_{\zeta\zeta\zeta} + 3(\chi_{\zeta\zeta}^2 + \chi_{\eta\zeta}^2)v_\zeta + 3(\chi_{\eta\zeta}\chi_{\eta\zeta\zeta} + \chi_{\zeta\zeta}\chi_{\zeta\zeta\zeta})v - 3\chi_{\zeta\zeta}[u_{\xi\xi\zeta}] + 3\chi_{\zeta\zeta}[u_{\zeta\zeta\zeta}] + 3\chi_{\eta\zeta}[u_{\eta\zeta\zeta}] \\ &\quad - \chi_{\zeta\zeta\zeta}[u_{\xi\xi}] + 3\chi_{\zeta\zeta}^2[u_{\xi\zeta}] + 3\chi_{\eta\zeta}\chi_{\zeta\zeta}[u_{\xi\eta}] + 3\chi_{\eta\zeta\zeta}[u_{\eta\zeta}] + 3\chi_{\zeta\zeta\zeta}[u_{\zeta\zeta}] + \chi_{\eta\zeta\zeta\zeta}[u_\eta] + \chi_{\zeta\zeta\zeta\zeta}[u_\zeta],\end{aligned}$$

$$\begin{aligned}
[u_{\xi\eta\eta\zeta}] &= v_{\eta\eta\zeta} + 2\chi_{\eta\zeta}(\chi_{\zeta\zeta} + \chi_{\eta\eta})v_{\eta} + (\chi_{\eta\zeta}^2 + \chi_{\eta\eta}^2)v_{\zeta} + (\chi_{\eta\zeta}(2\chi_{\eta\zeta\zeta} + \chi_{\eta\eta\eta}) + \chi_{\eta\eta\zeta}(2\chi_{\eta\eta} + \chi_{\zeta\zeta}))v \\
&\quad - 2\chi_{\eta\zeta}[u_{\xi\xi\eta}] - \chi_{\eta\eta}[u_{\xi\xi\zeta}] + \chi_{\eta\zeta}[u_{\eta\eta\eta}] + (\chi_{\zeta\zeta} + 2\chi_{\eta\eta})[u_{\eta\eta\zeta}] + 2\chi_{\eta\zeta}[u_{\eta\zeta\zeta}] - \chi_{\eta\eta\zeta}[u_{\xi\xi}] \\
&\quad + 3\chi_{\eta\zeta}\chi_{\eta\eta}[u_{\xi\eta}] + (\chi_{\eta\eta}\chi_{\zeta\zeta} + 2\chi_{\eta\zeta}^2)[u_{\xi\zeta}] + 2\chi_{\eta\eta\zeta}[u_{\eta\eta}] + (2\chi_{\eta\zeta\zeta} + \chi_{\eta\eta\eta})[u_{\eta\zeta}] \\
&\quad + \chi_{\eta\eta\zeta}[u_{\zeta\zeta}] + \chi_{\eta\eta\eta\zeta}[u_{\eta}] + \chi_{\eta\eta\zeta\zeta}[u_{\zeta}], \\
[u_{\xi\eta\zeta\zeta}] &= v_{\eta\zeta\zeta} + 2\chi_{\eta\zeta}(\chi_{\zeta\zeta} + \chi_{\eta\eta})v_{\zeta} + (\chi_{\eta\zeta}^2 + \chi_{\zeta\zeta}^2)v_{\eta} + (\chi_{\eta\zeta}(2\chi_{\eta\eta\zeta} + \chi_{\zeta\zeta\zeta}) + \chi_{\eta\zeta\zeta}(2\chi_{\zeta\zeta} + \chi_{\eta\eta}))v \\
&\quad - 2\chi_{\eta\zeta}[u_{\xi\xi\zeta}] - \chi_{\zeta\zeta}[u_{\xi\xi\eta}] + \chi_{\eta\zeta}[u_{\zeta\zeta\zeta}] + (\chi_{\eta\eta} + 2\chi_{\zeta\zeta})[u_{\eta\zeta\zeta}] + 2\chi_{\eta\zeta}[u_{\eta\eta\zeta}] - \chi_{\eta\zeta\zeta}[u_{\xi\xi}] \\
&\quad + 3\chi_{\eta\zeta}\chi_{\zeta\zeta}[u_{\xi\zeta}] + (\chi_{\eta\eta}\chi_{\zeta\zeta} + 2\chi_{\eta\zeta}^2)[u_{\xi\eta}] + 2\chi_{\eta\zeta\zeta}[u_{\zeta\zeta}] + (2\chi_{\eta\eta\zeta} + \chi_{\zeta\zeta\zeta})[u_{\eta\zeta}] \\
&\quad + \chi_{\eta\zeta\zeta}[u_{\eta\eta}] + \chi_{\eta\zeta\zeta\zeta}[u_{\zeta}] + \chi_{\eta\eta\zeta\zeta}[u_{\eta}], \\
[u_{\xi\xi\eta\eta}] &= [u_{t\eta\eta}] - [u_{\eta\eta\eta\eta}] - [u_{\eta\eta\zeta\zeta}] + [f_{\eta\eta}] + \chi_{\eta\eta}([u_{t\xi}] - [u_{\xi\xi\xi}] - [u_{\xi\xi\eta\eta}] - [u_{\xi\zeta\zeta}] + [f_{\xi}]), \\
[u_{\xi\xi\zeta\zeta}] &= [u_{t\zeta\zeta}] - [u_{\zeta\zeta\zeta\zeta}] - [u_{\eta\eta\zeta\zeta}] + [f_{\zeta\zeta}] + \chi_{\zeta\zeta}([u_{t\xi}] - [u_{\xi\xi\xi}] - [u_{\xi\xi\eta\eta}] - [u_{\xi\zeta\zeta}] + [f_{\xi}]), \\
[u_{\xi\xi\eta\zeta}] &= [u_{t\eta\zeta}] - [u_{\eta\eta\eta\zeta}] - [u_{\eta\zeta\zeta\zeta}] + [f_{\eta\zeta}] + \chi_{\eta\zeta}([u_{t\xi}] - [u_{\xi\xi\xi}] - [u_{\xi\xi\eta\eta}] - [u_{\xi\zeta\zeta}] + [f_{\xi}]), \\
[u_{\xi\xi\eta\eta}] &= [u_{t\xi\eta}] - [u_{\xi\eta\eta\eta}] - [u_{\xi\eta\zeta\zeta}] + [f_{\xi\eta}], \quad [u_{\xi\xi\zeta\zeta}] = [u_{t\xi\zeta}] - [u_{\xi\zeta\eta\eta}] - [u_{\xi\zeta\zeta\zeta}] + [f_{\xi\zeta}], \\
[u_{\xi\xi\zeta\zeta}] &= [u_{t\xi\zeta}] - [u_{\xi\xi\eta\eta}] - [u_{\xi\xi\zeta\zeta}] + [f_{\xi\xi}].
\end{aligned}$$

Then we can obtain the jump conditions required in the (x, y, z) coordinate system by the local coordinate transformation.

5. Numerical examples

We have performed a number of numerical experiments to check the accuracy of LOD-IIM schemes for heat conduction interface problems (2.1)-(2.3). Example 1 and Example 2 are constructed in 2D and Example 3 is in 3D. The convergence order of the scheme is given as

$$\text{Order} = \frac{\log(\|E_{N_1}\|/\|E_{N_2}\|)}{\log(N_2/N_1)}.$$

5.1. Example 1

In this example the interface is a circle $x^2 + y^2 = 0.25$, the solution domain $\Omega = [-1, 1] \times [-1, 1]$. The source term $f(x, y, t)$ is defined as

$$f(x, y, t) = \begin{cases} \exp(-t)(x^2 - y^2), & r \leq \frac{1}{2}, \\ 0, & r > \frac{1}{2}, \end{cases}$$

where $r = \sqrt{x^2 + y^2}$. The exact solution u of this problem has the same formula as f . The boundary condition and initial condition are specified by u .

In this example, both the function u and its normal derivative u_n have jumps, and these jumps vary along the interface. The jumps in u and u_n are chosen from the exact solution u . The figure of numerical solution and error plot on a 41×41 mesh are shown in Figure 4. The grid refinement analysis is presented in Table 1, in which the final time is $T = 2$.

Table 1: Grid refinement analysis in Example 1 with $T = 2$ ($\Delta t = h$)

N	$\ E_N\ _{\infty}$	Order
21	1.423648325615812e-007	*
41	3.824634103519187e-008	1.9645
61	1.737130295512146e-008	1.9865
81	9.777828435808544e-009	2.0266

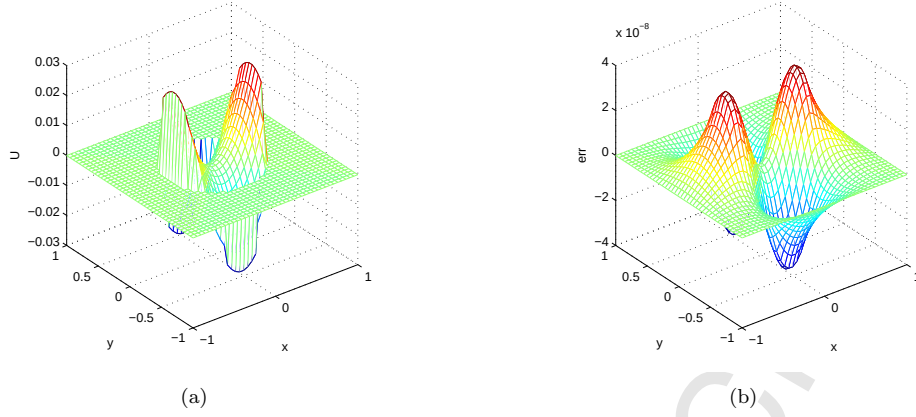


Figure 4: The numerical solution (a) and the error plot (b) of Example 1 on a 41×41 mesh with $T = 2$

5.2. Example 2

This example is constructed so that the exact solution is

$$u(x, y, t) = \begin{cases} \exp(-\frac{5}{4}t) \sin(x)(y^2 - 1)^2, & r \leq \frac{1}{2}, \\ 0, & r > \frac{1}{2}, \end{cases} \quad (5.27)$$

where $r = \sqrt{x^2 + y^2}$ and the source term $f(x, y, t)$ is derived from (2.1) and (5.27). The solution domain is $[-1, 1] \times [-1, 1]$ and the interface is a circle $\Gamma : x^2 + y^2 = 0.25$.

The figure of numerical solution and error plot on a 41×41 mesh are shown in Figure 5. The grid refinement analysis of this example is presented in Table 2 with the final time being $T = 2$.

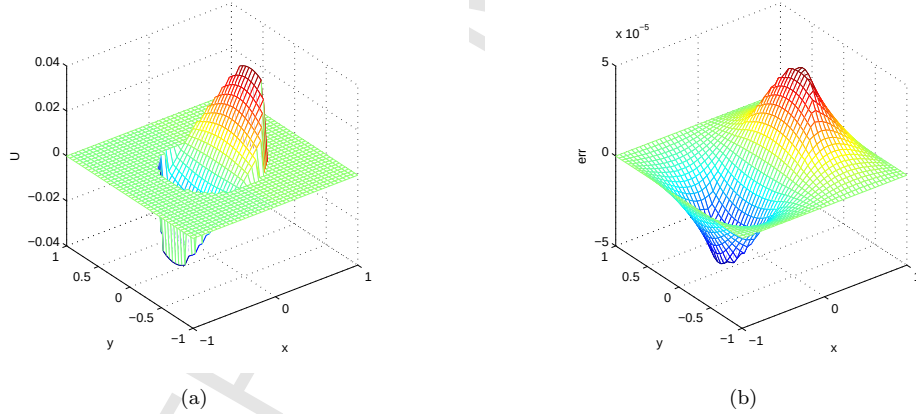


Figure 5: The numerical solution (a) and the error plot (b) of Example 2 on a 41×41 mesh with $T = 2$

Table 2: Grid refinement analysis in Example 2 with $T = 2$ ($\Delta t = h$)

N	$\ E_N\ _\infty$	Order
21	1.642819273395233e-004	*
41	4.497643828924426e-005	1.9362
61	2.058874627652226e-005	1.9668
81	1.169251833249396e-005	1.9952

5.3. Example 3

In this example we will verify the convergence order and stability of proposed LOD-IIM scheme on 3D heat conduction interface problems. Li's ADI-IIM is not stable for 3D problems unless the time step $\Delta t \leq 1.5h^2$. The example is constructed so that the solution domain is $\Omega = [-1, 1] \times [-1, 1] \times [-1, 1]$ and the interface is a sphere $x^2 + y^2 + z^2 = 0.25$. The exact solution $u(x, y, z, t)$ read

$$u(x, y, z, t) = \begin{cases} \exp(-\frac{5}{4}t) \sin(\pi x) \cos(\frac{1}{2}\pi y) \sin(\pi z), & r \leq \frac{1}{2}, \\ 0, & r > \frac{1}{2}, \end{cases} \quad (5.28)$$

where $r = \sqrt{x^2 + y^2 + z^2}$. The source term f is determined by (2.1) and (5.28). The grid refinement analysis of this example is presented in Table 3 with the final time being $T = 0.5$. From Table 3 we can find that if the time step is chosen as $\Delta t = h$, the proposed LOD-IIM scheme is stable and has two-order convergence, while the ADI-IIM will diverge as h goes to zero.

Table 3: Grid refinement analysis of Example 3 with $T = 0.5$ ($\Delta t = h$)

N	$\ E_N\ _\infty$ (LOD-IIM)	Order	$\ E_N\ _\infty$ (ADI-IIM)
21	0.178307188424012	*	0.010432810826413
41	0.046849107433850	1.9977	0.274416875175221
61	0.021125369398121	2.0047	3.979258572813444e+010
81	0.011964182356746	2.0050	

6. Concluding remark

In this paper a dimension by dimension splitting immersed interface method is proposed for both 2D and 3D heat conduction equations with interfaces. The method reduces a multi-dimensional interface problem into several one-dimensional interface problems by several steps, and in each step we only need to solve a tri-diagonal system. One correction term is added to the RHS of standard LOD difference scheme to ensure the local truncation error of order $O(h)$ at irregular points. In 2D problems, the correction term C is obtained by solving N tri-diagonal linear systems of order N at each time step, while in 3D problems N sparse linear systems of order N^2 need solving at each time step to get the correction term C . Although more time is taken to compute the correction term C in 3D problems, the whole time will be less than directly solving a large sparse linear system of order N^3 at each time step. Good agreement with analytical solution is shown by several numerical experiments and the order of convergence is found to be of second order in both time and space.

Acknowledgement

The authors thank the supports from the Institute of Applied Software of Central South University and from the High Performance Computing Platform of CSU, China. The authors wish to thank the anonymous reviewers for valuable comments and suggestions, which helped to improve the paper.

- [1] R. J. Leveque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, *SIAM Journal on Numerical Analysis* 31 (4) (1994) 1019–1044.
- [2] Z. Li, The Immersed Interface Method: A Numerical Approach for Partial Differential Equations with Interfaces, Ph.D. thesis, University of Washington (1994).
- [3] Z. Li, K. Ito, Maximum Principle Preserving Schemes for Interface Problems, *SIAM J. Sci. Comput* 23 (2001) 339–361.
- [4] L. Adams, Z. Li, The immersed interface/multigrid methods for interface problems, *SIAM Journal on Scientific Computing* 24 (2) (2003) 463–479.
- [5] A. Wiegmann, K. P. Bube, The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions, *SIAM Journal on Numerical Analysis* 37 (3) (2000) 827–862.

- [6] P. A. Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, *Journal of Computational Physics* 197 (1) (2004) 364–386.
- [7] Z. Li, A fast iterative algorithm for elliptic interface problems, *SIAM Journal on Numerical Analysis* 35 (1) (1998) 230–254.
- [8] H. Huang, Z. Li, Convergence analysis of the immersed interface method, *IMA Journal of Numerical Analysis* 19 (4) (1999) 583–608.
- [9] J. T. Beale, A. T. Layton, On the accuracy of finite difference methods for elliptic problems with interfaces, *Commun. Appl. Math. Comput. Sci.* 1 (1) (2006) 91–119.
- [10] Z. Li, Immersed interface methods for moving interface problems, *Numerical Algorithms* 14 (4) (1997) 269–293.
- [11] A. Wiegmann, K. P. Bube, The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources, *SIAM Journal on Numerical Analysis* 35 (1) (1998) 177–200.
- [12] J. D. Kandilarov, L. G. Vulkov, The immersed interface method for two-dimensional heat-diffusion equations with singular own sources, *Applied Numerical Mathematics* 57 (5-7) (2007) 486–497.
- [13] Z. Li, D. Wang, J. Zou, Theoretical and numerical analysis on a thermo-elastic system with discontinuities, *Journal of Computational and Applied Mathematics* 92 (1) (1998) 37–58.
- [14] D. W. Peaceman, H. H. Rachford Jr, The numerical solution of parabolic and elliptic differential equations, *Journal of the Society for Industrial and Applied Mathematics* 3 (1) (1955) 28–41.
- [15] E. G. D'yakonov, Difference schemes with split operators for multidimensional unsteady problems (English translation), *USSR Comp. Math.* 3 (1963) 581–607.
- [16] N. Yanenko, Convergence of the method of splitting for the heat conduction equations with variable coefficients (English translation), *USSR Comp. Math.* 3 (1963) 1094–1100.
- [17] Z. Li, A. Mayo, ADI methods for heat equations with discontinuities along an arbitrary interface, Vol. 48, *American Mathematical Society*, 1994, pp. 311–315.
- [18] Z. Li, B. Soni, Fast and accurate numerical approaches for Stefan problems and crystal growth, *Numerical Heat Transfer, Part B: Fundamentals* 35 (4) (1999) 461–484.
- [19] N. N. Yanenko, *The method of fractional steps*, Springer-Verlag., 1971.
- [20] K. W. Morton, D. F. Mayers, *Numerical solution of partial differential equations: an introduction*, Cambridge University Press, 2005.
- [21] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, *Journal of computational physics* 79 (1) (1988) 12–49.
- [22] Z. Li, K. Ito, *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*, Society for Industrial and Applied Mathematics, 2006.
- [23] Z. Li, A note on immersed interface method for three-dimensional elliptic equations, *Computers & Mathematics with Applications* 31 (3) (1996) 9–17.