INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 21, Number 3, Pages 353–374

doi: 10.4208/ijnam2024-1014

A DIRECT METHOD FOR SOLVING THREE-DIMENSIONAL ELLIPTIC INTERFACE PROBLEMS

KUMUDU GAMAGE[1,*], YAN PENG[2], AND ZHILIN LI[3]

Abstract. This paper presents a direct method for efficiently solving three-dimensional elliptic interface problems featuring piecewise constant coefficients with a finite jump across the interface. A key advantage of our approach lies in its avoidance of augmented variables, distinguishing it from traditional methods. The computational framework relies on a finite difference scheme implemented on a uniform Cartesian grid system. By utilizing a seven-point Laplacian for grid points away from the interface, our method only requires coefficient modifications for grid points located near or on the interface. Numerical experiments validate our method's effectiveness. Generally, it achieves second-order accuracy for both the solution and its gradient, measured in the maximum norm, particularly effective in scenarios with moderate coefficient jumps. Extending and building upon the recent work of [1] on 1D and 2D elliptic interfaces, our approach successfully introduces a simpler method for extension into three dimensions. Notably, our proposed method not only offers efficiency and accuracy but also enhances the simplicity of implementation, making it accessible to non-experts in the field.

Key words. Piecewise constant coefficients with a finite jump, elliptic interface problems, finite difference scheme.

1. Intrdoction

Interface problems have gained significant attention due to their extensive practical applications. Real-world examples of interface problems span diverse domains, including fluid dynamics where phenomena such as bubble formation have been studied [2], electromigration of voids [3], glacier prediction [4], growth of internal blood clots [5], and thermodynamics encompassing heat propagation in distinct materials. Additionally, these problems extend to areas related to Stefan problems, crystal growth [6], and various other applications.

In this paper, we consider the elliptic interface problem of the form,

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

$$[u](\mathbf{X}) = w(\mathbf{X}), \quad \mathbf{X} \in \Gamma, \tag{2}$$

$$[\beta u_{\boldsymbol{n}}](\mathbf{X}) = v(\mathbf{X}), \quad \mathbf{X} \in \Gamma, \tag{3}$$

with given boundary conditions on $\partial\Omega$, where, Γ is a smooth interface in the domain Ω and interface Γ divides the domain Ω into two subdomains Ω^+ and Ω^- and therefore, $\Omega = \Omega^+ \cup \Omega^- \cup \Gamma$. See Figure 1 for an illustration. $\mathbf{X}$ is a point on the interface Γ, $\mathbf{x}$ is a point in Ω and $\boldsymbol{n}$ is the unit outward normal vector to the interface at the point $\mathbf{X}$. The superscript $+$ or $-$ denotes the limiting value of a function from one side or the other of the interface. Here, $[u] = [u](\mathbf{X}) = u^+(\mathbf{X}) - u^-(\mathbf{X})$ is the jump in the solution at $\mathbf{X}$ and $u_{\boldsymbol{n}} = \boldsymbol{n}.\nabla u = \frac{\partial u}{\partial \boldsymbol{n}}$ is the normal derivative of the solution u. In many applications, the coefficient β often takes the form of a piecewise constant value, while the source term f may have discontinuities across the interface Γ. The jumps in the solution (2) and the flux (3), along with

Received by the editors on August 28, 2023 and, accepted on February 22, 2024.
2000 *Mathematics Subject Classification.* 35R35, 49J40, 60G40.
*Corresponding author.

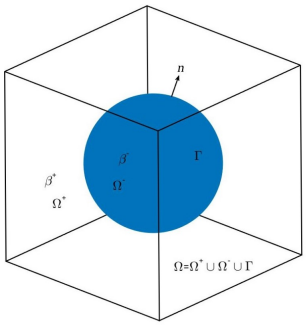

FIGURE 1. A diagram of a cubic domain Ω with a smooth interface Γ, where $\mathbf{n}$ represents the unit outward normal vector to the interface Γ.

the boundary conditions on $\partial\Omega$, are typically guided by the underlying physical principles.

Over time, substantial progress has been made in the development of numerical methods for solving interface problems [7, 8, 9, 10, 11, 12, 13, 14]. Strategies encompass the use of body-fitted grids [15, 16, 17], or the more favored Cartesian grids [12, 18, 19]. The latter choice gains prominence due to its simplified grid generation process, which is especially crucial when interfaces undergo frequent shape changes, often encountered in various physical phenomena. The preference for Cartesian grids is also amplified by the availability of versatile software tools such as fast Poisson solvers [20], Clawpack [21], Amrclawpack [22], the level set method [23, 24, 25], structured multigrid solvers [26, 27], and the immersed boundary method [10], along with others [28].

In Cartesian grid systems, interfaces are often embedded within a rectangular domain (in 2D) or a cube (in 3D). The immersed interface method (IIM), pioneered by LeVeque and Li [29], has emerged as a popular approach among the numerical community. Being the first second-order method for solving interface problems [30], IIM has been successfully applied to diverse linear and nonlinear problems, including hyperbolic elliptic systems [31], elasticity systems [32], [33], Hele-Shaw flow [34], traffic flow [35], glacier prediction [4], simulations of porosity evolution in chemical vapor infiltration [36], and shape identification in inverse problems [37].

However, while the IIM capably captures the solution and its gradient in the L^∞ norm for elliptic interface problems featuring variable coefficient β in various applications, it faces challenges when dealing with numerical examples characterized by significant jump discontinuities in the coefficient β. In such cases, the resulting linear system often becomes ill-conditioned, leading to potential non-convergence or inaccurate outcomes [38].

To address the aforementioned convergence issues, a fast immersed interface method (FIIM), also known as an augmented method, was introduced in [19]. This method involves a preconditioning step for the elliptic equation before applying the original IIM. Furthermore, it introduces an intermediate function to account for the jump in the normal derivative across the interface, enhancing the utilization of fast Poisson solvers. Although FIIM enhances accuracy, its implementation comes with complexities, notably involving establishing a Schur complement system, which adds computational overhead. Consequently, a novel direct IIM [1]

emerged, addressing one and two-dimensional elliptic interface problems with variable coefficients while eliminating the need for an augmented variable. This method reformulates the PDE at irregular grid points, integrating gradient and second-order derivative approximations at the interface into the finite difference scheme at these points. Consequently, this method achieves second-order accuracy for both the solution and its gradient.

In this paper, we extend the Direct IIM from [1] to three dimensions to address elliptic interface problems characterized by piecewise constant coefficients without introducing an augmented variable. This results in a method that is simpler to implement compared to the augmented IIM approach. Similar to [1], we circumvent the need for an augmented variable by incorporating gradient and second derivative approximations at the interface into the finite difference scheme. The resulting finite difference scheme involves a standard seven-point stencil for regular grid points and a twenty-seven compact scheme for irregular grid points. Convergence analysis conducted through grid refinement experiments on various numerical examples validates the effectiveness of our method. Generally, it achieves second-order accuracy for both the solution and its gradient, measured in the maximum norm, particularly effective in scenarios with moderate coefficient jumps. We have also noted that, for large jump ratios, we can still expect second-order accuracy when the meshes are fine enough. Furthermore, we establish the method's stability by combining eigenvalue analysis and condition number assessments. Our findings demonstrate that the coefficient matrix of the linear system for the finite difference scheme has exclusively negative real parts for its eigenvalues, and the condition number for an underdetermined linear system for finding the correction terms to the finite difference scheme does not change drastically with alterations in the coefficient β, further reinforcing the stability of our method. Moreover, we have conducted comprehensive CPU performance analyses using an AMD Ryzen Threadripper PRO 5965WX and verified the feasibility of these computations on a standard personal MacBook.

The paper is structured as follows: The next section delves into the algorithmic description for solving three-dimensional interface problems. Section 3 presents evidence of convergence and stability through numerical experiments. This section also elaborates on potential improvements for error analysis and the method's general applicability to diverse surface geometries. The final section concludes with a summary of key findings and contributions of our research while outlining future areas for improvement and exploration.

2. Algorithm description

In the context of three-dimensional elliptic interface problems with piecewise constant coefficients, we rephrase equation (1) as follows:

$$\beta(u_{xx} + u_{yy} + u_{zz}) = f(x, y, z), \quad (x, y, z) \in \Omega \backslash \Gamma, \tag{4}$$

$$[u](\mathbf{X}) = w(\mathbf{X}), \quad [\beta u_{\boldsymbol{n}}](\mathbf{X}) = v(\mathbf{X}). \tag{5}$$

For simplicity, we consider the domain Ω as a solid cube, denoted by $[a, b] \times [a, b] \times [a, b]$ (refer to Figure 1). To discretize the domain, we utilize a uniform grid system defined by:

$$\begin{aligned} x_i &= a + ih, \quad i = 0, 1, ..., N, \\ y_j &= a + jh, \quad j = 0, 1, ..., N, \\ z_l &= a + lh, \quad l = 0, 1, ..., N, \end{aligned} \tag{6}$$

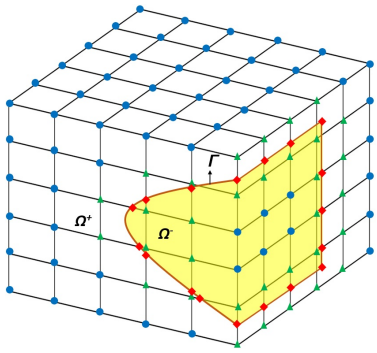

FIGURE 2. A cross-sectional illustration of domain Ω featuring a smooth interface Γ, where solid dots represent regular grid points, solid triangles represent irregular grid points, and solid diamonds represent the control points.

where $h = (b-a)/N$. Our objective is to develop a second-order accurate finite difference scheme of the form:

$$\sum_{m}^{n_s} \tau_m U_{i+i_m,j+j_m,l+l_m} = F_{ijl}, \tag{7}$$

at any grid point (x_i, y_j, z_l), where the summation is taken over n_s, the number of grid points centered at (x_i, y_j, z_l), and i_m, j_m, and l_m take the values from $-1, 0, 1$. The goal is to determine suitable coefficients τ_m that maintain second-order accuracy in both the solution and the gradient of the solution. We omit the explicit dependency of m on i, j, and l for simplicity. Furthermore, we aim to employ the standard seven-point stencil at regular grid points. A grid point (x_i, y_j, z_l) is classified as regular if the interface Γ does not intersect the grid lines connecting points within the standard seven-point stencil. Conversely, it is deemed irregular if such an intersection occurs (see Figure 2).

2.1. Finite difference scheme. For regular grid points (x_i, y_j, z_l), we begin by dividing equation (4) by β and apply a central difference scheme with a seven-point stencil. This results in the finite difference scheme:

$$\begin{aligned} &\frac{U_{i-1,j,l} - 2U_{i,j,l} + U_{i+1,j,l}}{h^2} \\ &+\frac{U_{i,j-1,l} - 2U_{i,j,l} + U_{i,j+1,l}}{h^2} \\ &+\frac{U_{i,j,l-1} - 2U_{i,j,l} + U_{i,j,l+1}}{h^2} = \frac{f_{i,j,l}}{\beta_{i,j,l}}. \end{aligned} \tag{8}$$

Here, $f_{i,j,l} = f(x_i, y_j, z_l)$, $\beta_{i,j,l} = \beta(x_i, y_j, z_l)$, and $U_{i,j,l}$ represents the numerical solution at $u(x_i, y_j, z_l)$.

For irregular grid points, a Taylor expansion at a control point leads to the finite difference scheme:

$$\begin{aligned}&\frac{U_{i-1,j,l}-2U_{i,j,l}+U_{i+1,j,l}}{h^2}+C^x_{i,j,l}\\ &+\frac{U_{i,j-1,l}-2U_{i,j,l}+U_{i,j+1,l}}{h^2}+C^y_{i,j,l}\\ &+\frac{U_{i,j,l-1}-2U_{i,j,l}+U_{i,j,l+1}}{h^2}+C^z_{i,j,l}=\frac{f_{i,j,l}}{\beta_{i,j,l}}.\end{aligned} \tag{9}$$

Assuming the seven-point stencil of an irregular grid point (x_i, y_j, z_k) intersects the right arm at the control point $\mathbf{x}^* = (x^*, y^*, z^*)$, the correction term $C^x_{i,j,l}$ is expressed as:

$$C^x_{i,j,l} = \pm\frac{[u]}{h^2} \pm [u_x]\frac{(x_{i+1}-x^*)}{h^2} \pm [u_{xx}]\frac{(x_{i+1}-x^*)^2}{2h^2}. \tag{10}$$

Here, plus or minus is chosen depending on which side the irregular grid point (x_i, y_j, z_l) lies. The correction terms in the y and z directions will also be in the same format.

The difference scheme for irregular points involves calculating jumps such as $[u]$, $[u_x]$, $[u_y]$, $[u_z]$, $[u_{xx}]$, $[u_{yy}]$, and $[u_{zz}]$. These jumps can be determined by differentiating the known jumps $[u] = w$ and $[\beta u_{\boldsymbol{n}}]$ at the interface. This is facilitated by performing a local coordinate transformation aligned with the normal and tangential directions to the interface Γ at the control point $\mathbf{x}^* = (x^*, y^*, z^*)$.

2.2. Local coordinate transformation. At a given point $(x^*, y^*, z^*) \in \Gamma$, let ξ represent the normal direction of Γ, while η and ζ are two orthogonal directions tangential to Γ. The local coordinates in these directions are given by:

$$\begin{aligned}\xi &= (x-x^*)\alpha_{x\xi} + (y-y^*)\alpha_{y\xi} + (z-z^*)\alpha_{z\xi},\\ \eta &= (x-x^*)\alpha_{x\eta} + (y-y^*)\alpha_{y\eta} + (z-z^*)\alpha_{z\eta},\\ \zeta &= (x-x^*)\alpha_{x\zeta} + (y-y^*)\alpha_{y\zeta} + (z-z^*)\alpha_{z\zeta},\end{aligned} \tag{11}$$

where $\alpha_{x\xi}$ is the directional cosine between the x-axis and ξ, and similarly for the other coefficients (See Figure 3). This transformation can be expressed in matrix form:

$$\begin{bmatrix}\xi\\ \eta\\ \zeta\end{bmatrix} = A\begin{bmatrix}x-x^*\\ y-y^*\\ z-z^*\end{bmatrix}, \tag{12}$$

with the matrix A given by:

$$A = \begin{bmatrix}\alpha_{x\xi} & \alpha_{y\xi} & \alpha_{z\xi}\\ \alpha_{x\eta} & \alpha_{y\eta} & \alpha_{z\eta}\\ \alpha_{x\zeta} & \alpha_{y\zeta} & \alpha_{z\zeta}\end{bmatrix}. \tag{13}$$

The first and second derivatives of any differentiable function $q(x, y, z)$ can be expressed in local coordinates using this transformation. The local coordinate transformation for the first derivative is given by:

$$\begin{bmatrix}\overline{q}_\xi\\ \overline{q}_\eta\\ \overline{q}_\zeta\end{bmatrix} = A\begin{bmatrix}q_x\\ q_y\\ q_z\end{bmatrix}. \tag{14}$$

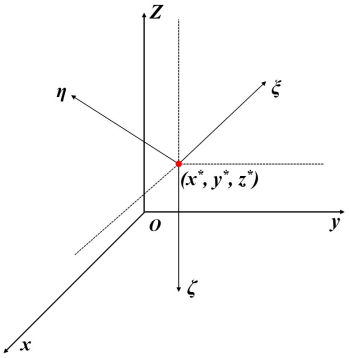

FIGURE 3. A sketch of a three-dimensional local coordinate transformation.

Similarly, the local coordinate transformation for the second derivative is:

$$\begin{bmatrix} \overline{q}_{\xi\xi} & \overline{q}_{\xi\eta} & \overline{q}_{\xi\zeta} \\ \overline{q}_{\eta\xi} & \overline{q}_{\eta\eta} & \overline{q}_{\eta\zeta} \\ \overline{q}_{\zeta\xi} & \overline{q}_{\zeta\eta} & \overline{q}_{\zeta\zeta} \end{bmatrix} = A \begin{bmatrix} q_{xx} & q_{xy} & q_{xz} \\ q_{yx} & q_{yy} & q_{yz} \\ q_{zx} & q_{zy} & q_{zz} \end{bmatrix} A^T, \tag{15}$$

where A^T is the transpose of the matrix A. It is important to note that during the local coordinate transformation, the partial differential equation (PDE), equation (4) retains the same. Therefore, we drop the bars for simplicity.

2.3. Local coordinate transformation in terms of level set function. At a given point (x^*, y^*, z^*) on the interface, let ξ be aligned with the outward normal direction of the interface, given by:

$$\xi = \frac{\bigtriangledown \phi}{|\bigtriangledown \phi|} = \frac{(\phi_x, \quad \phi_y, \quad \phi_z)^T}{\sqrt{\phi_x^2 + \phi_y^2 + \phi_z^2}}, \tag{16}$$

where $\nabla\phi = (\phi_x, \phi_y, \phi_z)^T$ is the gradient of the level set function ϕ.
For the first tangential direction η, we have:

$$\begin{aligned} &\text{if} \quad \phi_x^2 + \phi_y^2 \neq 0; \qquad \eta = \frac{(\phi_y, \quad -\phi_x, \quad 0)^T}{\sqrt{\phi_x^2 + \phi_y^2}}, \\ &\text{otherwise} \qquad \eta = \frac{(\phi_z, \quad 0, \quad -\phi_x)^T}{\sqrt{\phi_x^2 + \phi_z^2}}. \end{aligned} \tag{17}$$

For the second tangential direction ζ, we have:

(18)

$$\begin{aligned} &\text{if} \quad \phi_x^2 + \phi_y^2 \neq 0; \qquad \zeta = \frac{\mathbf{s}}{|\mathbf{s}|}, \qquad \text{where} \qquad \mathbf{s} = (\phi_x\phi_z, \quad \phi_y\phi_z, \quad -\phi_x^2 - \phi_y^2)^T, \\ &\text{otherwise} \qquad \zeta = \frac{\mathbf{t}}{|\mathbf{t}|} \qquad \text{where} \qquad \mathbf{t} = (-\phi_x\phi_y, \quad \phi_x^2 + \phi_z^2, \quad -\phi_y\phi_z)^T. \end{aligned}$$

2.4. Computing principal curvatures using the level set function. In the vicinity of a control point $\mathbf{x}^* = (x^*, y^*, z^*)$, the interface can be parametrized as:

$$\xi = \chi(\eta, \zeta), \quad \text{with} \quad \chi(0,0) = 0, \quad \chi_\eta(0,0) = 0, \quad \chi_\zeta(0,0) = 0. \tag{19}$$

To calculate the derivatives of jumps at the interface, we require the second-order tangential derivatives $\chi_{\eta\eta}$, $\chi_{\zeta\zeta}$, and $\chi_{\eta\zeta}$ of χ at $\mathbf{x}^*$.

On the interface, we have the relation $\phi(\chi(\eta,\zeta),\eta,\zeta)=0$. Starting with implicit differentiation, consider the derivatives of ϕ with respect to η and ζ:

$$\phi_\eta + \phi_\xi \chi_\eta = 0, \tag{20}$$

$$\phi_\zeta + \phi_\xi \chi_\zeta = 0. \tag{21}$$

Next, differentiate equations (20) and (21) with respect to η and ζ, resulting in:

$$\begin{aligned}
&\phi_{\eta\eta} + \phi_{\eta\xi}\chi_\eta + (\phi_{\xi\eta} + \phi_{\xi\xi}\chi_\eta)\chi_\eta + \phi_\xi \chi_{\eta\eta} = 0,\\
&\phi_{\eta\zeta} + \phi_{\eta\xi}\chi_\zeta + (\phi_{\xi\zeta} + \phi_{\xi\xi}\chi_\zeta)\chi_\eta + \phi_\xi \chi_{\eta\zeta} = 0,\\
&\phi_{\zeta\zeta} + \phi_{\zeta\xi}\chi_\zeta + (\phi_{\xi\zeta} + \phi_{\xi\xi}\chi_\zeta)\chi_\zeta + \phi_\xi \chi_{\zeta\zeta} = 0.
\end{aligned} \tag{22}$$

Since, $\chi_\eta(0,0)=0$ and $\chi_\zeta(0,0)=0$ on the interface, the following relationships emerge:

$$\begin{aligned}
\chi_{\eta\eta} &= -\phi_{\eta\eta}/\phi_\xi,\\
\chi_{\zeta\zeta} &= -\phi_{\zeta\zeta}/\phi_\xi,\\
\chi_{\eta\zeta} &= -\phi_{\eta\zeta}/\phi_\xi,
\end{aligned} \tag{23}$$

where,

$$\begin{bmatrix} \phi_\xi \\ \phi_\eta \\ \phi_\zeta \end{bmatrix} = A \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix}. \tag{24}$$

2.5. Local coordinate transformation of jump conditions. In order to determine the jump conditions $[u]$, $[u_x]$, $[u_y]$, $[u_z]$, $[u_{xx}]$, $[u_{yy}]$, and $[u_{zz}]$ at a control point $\mathbf{x}^* = (x^*, y^*, z^*)$, we find it convenient to differentiate the jump conditions $[u] = w$ and $[\beta u_{\boldsymbol{n}}] = v$ along the interface and then proceed with coordinate transformation.

Let's begin by differentiating $[u] = w$ with respect to η and ζ:

$$[u_\xi]\chi_\eta + [u_\eta] = w_\eta, \tag{25}$$

$$[u_\xi]\chi_\zeta + [u_\zeta] = w_\zeta. \tag{26}$$

Differentiating equation (25) with respect to η gives:

$$\chi_\eta \frac{\partial [u_\xi]}{\partial \eta} + \chi_{\eta\eta}[u_\xi] + \chi_\eta [u_{\eta\xi}] + [u_{\eta\eta}] = w_{\eta\eta}. \tag{27}$$

Differentiating equation (25) with respect to ζ yields:

$$\chi_\eta \frac{\partial [u_\xi]}{\partial \zeta} + \chi_{\eta\zeta}[u_\xi] + \chi_\zeta [u_{\eta\xi}] + [u_{\eta\zeta}] = w_{\eta\zeta}. \tag{28}$$

Similarly, differentiating equation (26) with respect to ζ results in:

$$\chi_\zeta \frac{\partial [u_\xi]}{\partial \zeta} + \chi_{\zeta\zeta}[u_\xi] + \chi_\zeta [u_{\zeta\xi}] + [u_{\zeta\zeta}] = w_{\zeta\zeta}. \tag{29}$$

Next, we express the unit normal vector of the interface in terms of χ:

$$\boldsymbol{n} = \frac{(1, -\chi_\eta, -\chi_\zeta)}{\sqrt{1 + \chi_\eta^2 + \chi_\zeta^2}}, \tag{30}$$

and rewrite the flux jump condition $[\beta u_{\boldsymbol{n}}] = v$ using local coordinate transformations:

$$[\beta(u_\xi - u_\eta\chi_\eta - u_\zeta\chi_\zeta)] = v(\eta,\zeta)\sqrt{1+\chi_\eta^2+\chi_\zeta^2}. \tag{31}$$

Differentiating equation (31) with respect to η, we get,

$$\begin{aligned}[\beta(u_{\xi\xi}\chi_\eta + u_{\xi\eta} - \chi_\eta\frac{\partial u_\eta}{\partial\eta} - \chi_\zeta\frac{\partial u_\zeta}{\partial\eta} - u_\eta\chi_{\eta\eta} - u_\zeta\chi_{\eta\zeta})] \\ = v_\eta\sqrt{1+\chi_\eta^2+\chi_\zeta^2} + v\frac{\chi_\eta\chi_{\eta\eta}}{\sqrt{1+\chi_\eta^2+\chi_\zeta^2}}.\end{aligned} \tag{32}$$

Likewise, differentiating the equation (31) with respect to ζ, we get,

$$\begin{aligned}[\beta(u_{\xi\xi}\chi_\zeta + u_{\xi\zeta} - \chi_\eta\frac{\partial u_\eta}{\partial\zeta} - \chi_\zeta\frac{\partial u_\zeta}{\partial\zeta} - u_\eta\chi_{\eta\zeta} - u_\zeta\chi_{\zeta\zeta})] \\ = v_\zeta\sqrt{1+\chi_\eta^2+\chi_\zeta^2} + v\frac{\chi_\zeta\chi_{\zeta\zeta}}{\sqrt{1+\chi_\eta^2+\chi_\zeta^2}}.\end{aligned} \tag{33}$$

With $\chi_\eta(0,0) = \chi_\zeta(0,0) = 0$ on the interface, the jump relations (25)-(33) simplify to:

$$[u] = w, \tag{34}$$

$$[u_\eta] = w_\eta, \tag{35}$$

$$[u_\zeta] = w_\zeta, \tag{36}$$

$$[\beta u_\xi] = v, \tag{37}$$

$$[u_{\eta\eta}] = -\chi_{\eta\eta}[u_\xi] + w_{\eta\eta} = D_1, \tag{38}$$

$$[u_{\zeta\zeta}] = -\chi_{\zeta\zeta}[u_\xi] + w_{\zeta\zeta} = D_2, \tag{39}$$

$$[u_{\eta\zeta}] = -\chi_{\eta\zeta}[u_\xi] + w_{\eta\zeta} = D_3, \tag{40}$$

$$[\beta u_{\xi\eta}] = \chi_{\eta\eta}[\beta u_\eta] + \chi_{\eta\zeta}[\beta u_\zeta] + v_\eta = D_4, \tag{41}$$

$$[\beta u_{\xi\zeta}] = \chi_{\eta\zeta}[\beta u_\eta] + \chi_{\zeta\zeta}[\beta u_\zeta] + v_\zeta = D_5. \tag{42}$$

In addition to these relations, the PDE (4) provides two more jump conditions:

$$u_{xx}^+ + u_{yy}^+ + u_{zz}^+ = \frac{f^+}{\beta^+} = D_6, \tag{43}$$

$$u_{xx}^- + u_{yy}^- + u_{zz}^- = \frac{f^-}{\beta^-} = D_7. \tag{44}$$

The subsequent step involves transforming these jump conditions (34)-(44) into Cartesian coordinates and expressing all the limiting values from outside the interface. The expressions for u^+, u_x^+, u_y^+, u_z^+, u_{xx}^+, u_{xy}^+, u_{xz}^+, u_{yy}^+, u_{yz}^+, and u_{zz}^+ can be found in Appendix A [39]. Notably, D_1 through D_5 can also be expressed in terms of u_x^-, u_y^-, and u_z^-.

2.6. The approximation of the correction terms. In this section, we will discuss how to interpolate the correction terms $C_{i,j,l}^x$, $C_{i,j,l}^y$, and $C_{i,j,l}^z$ using the Cartesian coordinate transformations of the jump relations given by equations (34) to (42), as well as the other two jump conditions equations (43) and (44) derived from the PDE itself.

Given an irregular grid point (x_i, y_j, z_l), we first select a point $\mathbf{x}^* = (x_i^*, y_j^*, z_l^*)$ on the interface Γ near (x_i, y_j, z_l). Here, we consider this point the control point closest to (x_i, y_j, z_l).

To ensure that the resulting difference scheme at the irregular point follows the form of equation (7), we expand each $U_{i+i_m,j+j_m,l+l_m}$ around the control point $\mathbf{x}^* = (x_i^*, y_j^*, z_l^*)$. Here, $m = 1, 2.., n_s$, and n_s is the number of grid points in the difference scheme. The values of i_m, j_m, and l_m will be selected from $-1, 0, 1$. We will elaborate later on how to choose n_s. Additionally, it is important to note that m depends on i, j, and l, although we have omitted this dependence for simplicity.

Without loss of generality, let's assume that (x_i, y_j, z_l) is an irregular point, and the seven-point stencil around (x_i, y_j, z_l) only cuts through the right arm, while $(x_i, y_j, z_l) \in \Omega^-$. We will proceed by considering the Taylor expansion of $u(x_{i+i_m}, y_{j+j_m}, z_{l+l_m})$ around the control point $\mathbf{x}^* = (x_i^*, y_j^*, z_l^*)$.

$$\begin{aligned} u(x_{i+i_m}, y_{j+j_m}, z_{l+l_m}) &= u^{\pm} + (x_{i+i_m} - x_i^*)u_x^{\pm} + (y_{j+j_m} - y_j^*)u_y^{\pm} \\ &+ (z_{l+l_m} - z_l^*)u_z^{\pm} + \frac{1}{2}(x_{i+i_m} - x_i^*)^2 u_{xx}^{\pm} \\ &+ \frac{1}{2}(y_{j+j_m} - y_j^*)^2 u_{yy}^{\pm} + \frac{1}{2}(z_{l+l_m} - z_l^*)^2 u_{zz}^{\pm} \\ &+ (x_{i+i_m} - x_i^*)(y_{j+j_m} - y_j^*)u_{xy}^{\pm} \\ &+ (x_{i+i_m} - x_i^*)(z_{l+l_m} - z_l^*)u_{xz}^{\pm} \\ &+ (y_{j+j_m} - y_j^*)(z_{l+l_m} - z_l^*)u_{yz}^{\pm} + O(h^3). \end{aligned} \tag{45}$$

In the above expression, the plus or minus sign is chosen based on whether $(x_{i+i_m}, y_{j+j_m}, z_{l+l_m})$ lies in Ω^+ or Ω^-. By utilizing the coordinate transformation of the nine interface relations (34) to (42), along with the other two jump relations given by equations (43) and (44), we can eliminate the limiting values from the plus side using the limiting values from the minus side, as explained in Appendix A [39]. Consequently, the Taylor series expansion of equation (45) will include terms dependent on u^-, u_x^-, u_y^-, u_z^-, u_{xx}^-, u_{xy}^-, u_{xz}^-, u_{yy}^-, and u_{yz}^-.

Now, the equation (45) can be expressed in the form:

$$\begin{aligned} u(x_{i+i_m}, y_{j+j_m}, z_{l+l_m}) &= c_m^1 u^- + c_m^2 u_x^- + c_m^3 u_y^- + c_m^4 u_z^- + c_m^5 u_{xx}^- \\ &+ c_m^6 u_{xy}^- + c_m^7 u_{xz}^- + c_m^8 u_{yy}^- + c_m^9 u_{yz}^- + c_m^{10}. \end{aligned} \tag{46}$$

Here, coefficients c_m^1 through c_m^{10} are known quantities. Consequently, the correction term $C_{i,j,l}^x$ can be reformulated as:

$$\begin{aligned} C_{i,j,l}^x =& a_1 u^- + a_2 u_x^- + a_3 u_y^- + a_4 u_z^- + a_5 u_{xx}^- \\ &+ a_6 u_{xy}^- + a_7 u_{xz}^- + a_8 u_{yy}^- + a_9 u_{yz}^- + a_{10}. \end{aligned} \tag{47}$$

In the above expression, coefficients a_1 through a_{10} are known. The interpolation for $C_{i,j,l}^y$ and $C_{i,j,l}^z$ will follow the same form as for $C_{i,j,l}^x$, with only the coefficients being different.

Now, we make an assumption that the correction terms $C_{i,j,l}^x$ can be approximated as follows:

$$C_{i,j,l}^x = \sum_m^{n_s} \gamma_m U_{i+i_m,j+j_m,l+l_m} + \gamma_c. \tag{48}$$

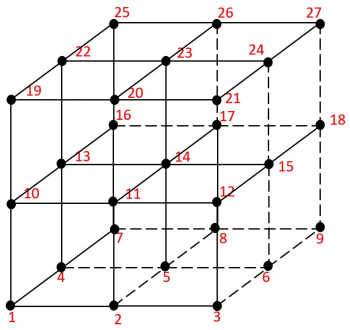

FIGURE 4. The 27-point stencil for three-dimensional compact scheme.

By comparing equations (47) and (48), we derive the following linear system:

$$\begin{bmatrix} c_1^1 & c_2^1 & \cdots & c_3^1 & c_{m-1}^1 & c_m^1 \\ c_1^2 & c_2^2 & \cdots & c_3^2 & c_{m-1}^2 & c_m^2 \\ & & \cdots & & & \\ c_1^8 & c_2^8 & \cdots & c_3^8 & c_{m-1}^8 & c_m^8 \\ c_1^9 & c_2^9 & \cdots & c_3^9 & c_{m-1}^9 & c_m^9 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \cdots \\ \gamma_{m-1} \\ \gamma_m \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \end{bmatrix}. \tag{49}$$

The value of γ_c in equation (48) can be determined using the following expression:

$$\gamma_c = a_{10} - \sum_{m}^{n_s} c_m^{10} \cdot \gamma_m. \tag{50}$$

The linear system presented in equation (49) consists of nine linear equations for determining the coefficients γ_m. To ensure a solvable system, we choose $n_s = 27$, which corresponds to all 27 grid points within the cube centered at the irregular grid point (x_i, y_j, z_l), as illustrated in Figure 4. Although we encounter an underdetermined linear system due to the greater number of unknowns than equations, we can solve it using singular value decomposition (SVD) to obtain the minimum norm solution. At this stage, the finite difference scheme at the irregular grid point (x_i, y_j, z_l) is entirely determined. The linear system for the finite difference equations can be expressed as:

$$A_h \mathbf{U} = \boldsymbol{F}. \tag{51}$$

Let us delve further into the coefficient matrix (51). As previously described, the coefficients τ_m in the finite difference scheme (7) are obtained by incorporating the γ_m values, which are derived by solving the underdetermined linear system equation (49), into the standard seven-point stencil. This stencil retains the same coefficients as those of the finite difference scheme used for the discrete Laplacian operator in a Poisson equation in the absence of an interface. Notably, these γ_m values are contingent on the interface's location and the jump β in the PDE (4).

It is important to note that the coefficients γ_m in the correction terms are generally of order $O(1/h^2)$. However, our numerical experiments indicate that their magnitudes are significantly smaller than $1/h^2$. Consequently, the dominant weights in the FD scheme (7) are still determined by the grid values of the seven-point

stencil. As a result, the eigenvalues of the coefficient matrix will remain within the stability region. In section 3, we will provide a specific example that illustrates the magnitudes of eigenvalues and coefficients γ_m for a problem with a moderate jump in β.

Now, to recover the gradient of the solution at the interface, the following formulas can be employed:

$$U_\xi^- = \alpha_{x\xi} U_x^- + \alpha_{y\xi} U_y^- + \alpha_{z\xi} U_z^-, \tag{52}$$

$$U_\xi^+ = \rho\alpha_{x\xi} U_x^- + \rho\alpha_{y\xi} U_y^- + \rho\alpha_{z\xi} U_z^- + v/\beta^+, \tag{53}$$

$$U_\eta^- = \alpha_{x\eta} U_x^- + \alpha_{y\eta} U_y^- + \alpha_{z\eta} U_z^-, \tag{54}$$

$$U_\eta^+ = \alpha_{x\eta} U_x^- + \alpha_{y\eta} U_y^- + \alpha_{z\eta} U_z^- + w_\eta, \tag{55}$$

$$U_\zeta^- = \alpha_{x\zeta} U_x^- + \alpha_{y\zeta} U_y^- + \alpha_{z\zeta} U_z^-, \tag{56}$$

$$U_\zeta^+ = \alpha_{y\zeta} U_x^- + \alpha_{y\zeta} U_y^- + \alpha_{z\zeta} U_z^- + w_\zeta. \tag{57}$$

Here, ξ represents the unit normal direction, while η and ζ denote the tangential directions to the interface at the control point $\mathbf{x}^* = (x_i^*, y_j^*, z_l^*)$. Additionally, $\rho = \beta^-/\beta^+$ stands for the jump ratio. Equations (52) and (53) are obtained through the coordinate transformation of equation (37). Equations (54) and (55) are derived from the coordinate transformation of equation (35). Similarly, equations (56) and (57) are acquired by the coordinate transformation of equation (36). To interpolate these gradients from the solution, we again assume that the interpolation scheme takes the form of an equation (48). Here, the point (x_i, y_j, z_l) is chosen as the closest irregular point to the considered control point. Consequently, the resulting linear system resembles equation (49), but with the right-hand vector replaced by $[0, \alpha_{x\xi}, \alpha_{y\xi}, \alpha_{z\xi}, 0, 0, 0, 0, 0, 0]^T$ for calculating U_ξ^- and $[0, \rho\alpha_{x\xi}, \rho\alpha_{y\xi}, \rho\alpha_{z\xi}, 0, 0, 0, 0, 0, v/\beta^+]^T$ for U_ξ^+.

2.7. Algorithm Overview. This section presents an outline of the algorithm for solving three-dimensional elliptic interface problems.

Step 1: Embed the irregular domain (interface) within a cubic domain $\Omega = [a, b] \times [a, b] \times [a, b]$ and represent the interface using a zero level set function.

Step 2: Determine the regular and irregular grid points, along with the locations of control points, which correspond to the intersection points of the interface and the grid lines. This is achieved using the level set grid function ϕ.

Step 3: Apply the standard 7-point central difference scheme at the regular grid points.

Step 4: Solve the underdetermined linear system provided by equation (49) to compute the correction terms $C^x_{i_k,j_k,l_k}$, $C^y_{i,j,l}$, and $C^z_{i,j,l}$ at irregular grid points. Subsequently, establish the 27-point compact scheme at irregular points.

Step 5: Solve the system of linear equations detailed in equation (51).

Step 6: Recover the gradients of the solution.

3. Numerical Examples

This section presents numerical experiments for 3D elliptic interface problems with Dirichlet boundary conditions using the proposed Direct IIM. Nevertheless, other linear boundary conditions, such as the Neumann or Robin type of BCs, can be easily adapted as long as the problem has a unique solution. The boundary conditions are implemented by modifying both the coefficients in the matrix and the right-hand side of the linear system (51). These experiments were conducted on an AMD Ryzen Threadripper PRO 5965WX to demonstrate the method's efficiency,

with detailed CPU times provided for various numerical examples. Furthermore, we validated the method's practical applicability on commonly used computing devices by successfully conducting computations on a standard personal MacBook, accommodating grid sizes up to 80 x 80 x 80. The linear system, as outlined in equation (51), was solved using the backslash operator in MATLAB.

Error measurements throughout our experiments are quantified in the L^∞ norm. The convergence order is calculated using the following formula:

$$\text{order} = \frac{1}{\log(2)} \log \frac{\|E_{2h}\|_\infty}{\|E_h\|_\infty}. \tag{58}$$

For all problems, $\beta(x, y, z)$ is defined as:

$$\beta(x, y, z) = \begin{cases} \beta^- & \text{if } (x, y, z) \in \Omega^-, \\ \beta^+ & \text{if } (x, y, z) \in \Omega^+. \end{cases} \tag{59}$$

In the convergence analysis tables, N represents the number of grid points in each direction. $E(u)$ denotes the maximum norm error of the numerical solution. Additionally, $E(u_n)$, $E(u_\eta)$, and $E(u_\zeta)$ respectively represent the maximum norm errors in the normal and tangential components of the solution.

Example 3.1.
In this example, the solution $u(x, y, z)$ and its flux demonstrate jump discontinuities. The governing differential equation is:

$$(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z = f. \tag{60}$$

The domain Ω is defined as $[-1, 1] \times [-1, 1] \times [-1, 1]$, with the interface Γ being a sphere described by:

$$\phi(x, y, z) = x^2 + y^2 + z^2 - 0.25. \tag{61}$$

The source term f is defined as follows:

$$f(x, y, z) = \begin{cases} 6\beta^- & \text{if } (x, y, z) \in \Omega^-, \\ 6\beta^+ & \text{if } (x, y, z) \in \Omega^+. \end{cases} \tag{62}$$

The jump in the solution $u(x, y, z)$ and its flux is given by:

$$[u] = 10, \tag{63}$$

$$[\beta u_n] = \beta^+ - \beta^-. \tag{64}$$

The Dirichlet boundary conditions are derived from the exact solution, which is defined as follows:

$$u(x, y, z) = \begin{cases} x^2 + y^2 + z^2 & \text{if } (x, y, z) \in \Omega^-, \\ x^2 + y^2 + z^2 + 10 & \text{if } (x, y, z) \in \Omega^+. \end{cases} \tag{65}$$

Figure 5 displays slices of the computed solution and the corresponding error distributions. We have conducted several cases for different jumps in β. For all these cases, we have observed that the maximum norm errors exhibit magnitudes ranging from 10^{-12} to 10^{-15} in the infinity norm. Detailed grid refinement analyses are presented in Tables 1 and 2. These errors primarily originate from round-off errors, which can be attributed to the quadratic nature of the solution.

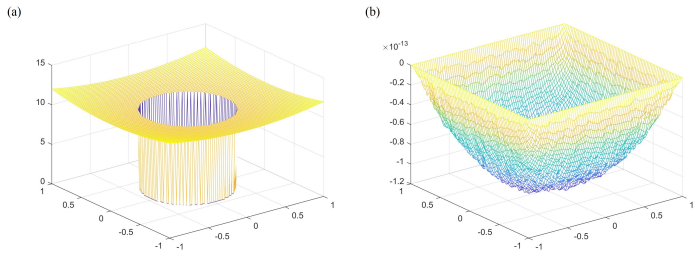

FIGURE 5. (a) Plot of the numerical solution for Example 3.1. (b) Plot of the error in the numerical solution with $N = 80$, $\beta^- = 1$, and $\beta^+ = 10$.

TABLE 1. A grid refinement analysis for Example 3.1 with $\beta^- = 1$ and $\beta^+ = 1$.

N	$E(u)$	$E(u_n)$	$E(u_\eta)$	$E(u_\varsigma)$
10	8.81E-13	3.73E-13	3.32E-13	3.82E-13
20	4.30E-13	1.10E-13	1.18E-13	1.53E-13
40	2.10E-14	1.93E-14	4.92E-14	3.34E-14
80	1.02E-14	0.54E-15	1.92E-14	1.68E-14
160	9.97E-15	9.40E-15	8.26E-15	6.59E-15

TABLE 2. A grid refinement analysis for Example 3.1 with $\beta^- = 1$ and $\beta^+ = 10^4$.

N	$E(u)$	$E(u_n)$	$E(u_\eta)$	$E(u_\varsigma)$
10	8.88E-12	3.73E-12	3.32E-12	3.82E-12
20	4.30E-13	1.10E-13	8.18E-13	8.53E-13
40	1.60E-14	1.93E-14	1.92E-14	2.34E-14
80	6.82E-15	3.64E-15	4.83E-15	5.68E-15
160	2.77E-15	1.40E-15	1.36E-15	1.59E-15

Example 3.2. **Stability analysis and applicability for general interfaces.** In this example, our initial focus is on the stability analysis of our method. This entails two primary components:

1. Eigenvalue analysis: An examination of the eigenvalues of the coefficient matrix derived from our finite difference scheme, as detailed in equation (51).

2. Condition number analysis: This involves investigating the condition number under varying values of β for the underdetermined linear system, as outlined in equation (49).

After a detailed analysis focused on the stability of our method, we will later extend our exploration to assess its applicability to general surface interfaces within this section.

To commence with the stability analysis, we consider an elliptic interface problem characterized by jump discontinuities in both the solution $u(x, y, z)$ and its flux. In

this analysis, the governing differential equation is defined as follows:

$$(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z = f. \tag{66}$$

The domain Ω is specified as $[-1,1] \times [-1,1] \times [-1,1]$. The interface, represented as a sphere, is given by the equation:

$$\phi(x,y,z) = x^2 + y^2 + z^2 - 0.25. \tag{67}$$

The source term f varies according to:

$$f(x,y,z) = \begin{cases} 2\beta^- e^{r^2}(3+2r^2) & \text{if } (x,y,z) \in \Omega^-, \\ 2\beta^+ e^{r^2}(3+2r^2) & \text{if } (x,y,z) \in \Omega^+, \end{cases} \tag{68}$$

where $r = \sqrt{x^2+y^2+z^2}$. The jumps in the solution and flux and boundary conditions are derived using the exact solution given by:

$$u(x,y,z) = \begin{cases} e^{r^2} & \text{if } (x,y,z) \in \Omega^-, \\ e^{r^2} + 1 & \text{if } (x,y,z) \in \Omega^+. \end{cases} \tag{69}$$

Detailed grid refinement analyses are provided in Tables 3 through 6 by varying the jump in β. The results demonstrate near second-order accuracy for both the solution and the gradient of the solution.

TABLE 3. A grid refinement analysis for Example 3.2 for interface as a sphere with $\beta^- = 1$ and $\beta^+ = 1$.

N	$E(u)$	order	$E(u_{\boldsymbol{n}})$	order	$E(u_\eta)$	order	$E(u_\zeta)$	order
10	6.40E-02		0.134324		0.141663		0.14311	
20	1.83E-02	1.81	3.47E-02	1.95	4.08E-02	1.80	4.15E-02	1.79
40	4.85E-03	1.91	8.98E-03	1.95	1.12E-02	1.86	1.09E-02	1.93
80	1.25E-03	1.96	2.41E-03	1.90	2.79E-03	2.01	2.78E-03	1.97
160	3.16E-04	1.98	6.01E-04	2.00	7.05E-04	1.99	7.59E-04	1.87

TABLE 4. A grid refinement analysis for Example 3.2 for interface as a sphere with $\beta^- = 1$ and $\beta^+ = 10^2$.

N	$E(u)$	order	$E(u_{\boldsymbol{n}})$	order	$E(u_\eta)$	order	$E(u_\tau)$	order
10	8.54E-02		1.23E-01		3.64E-01		3.73E-01	
20	2.40E-02	1.83	2.95E-02	2.06	8.97E-02	2.02	8.70E-02	2.10
40	6.32E-03	1.92	9.94E-03	1.57	2.50E-02	1.84	2.49E-02	1.81
80	1.61E-03	1.97	3.18E-03	1.64	9.32E-03	1.42	9.38E-03	1.41
160	4.08E-04	1.98	1.16E-03	1.45	2.26E-03	2.05	2.29E-03	2.04

Eigenvalue analysis: To assess our method's stability, we analyzed the eigenvalues of the coefficient matrix obtained from the finite difference (FD) scheme (51). Figure 6 illustrates the eigenvalues of the coefficient matrix of the FD scheme for Example 3.2 with the interface as a sphere. The analysis is conducted for a grid size of $N = 22$ points in each direction of the computational domain. The eigenvalues are presented for two cases: (a) $\beta^- = 1$ and $\beta^+ = 100$ and (b) $\beta^- = 100$ and $\beta^+ = 1$. Notably, for both cases, the coefficient matrix of the FD scheme exclusively has negative real parts for its eigenvalues, indicating the stability of the original method in this particular example. Figure 7 provides insight into the values of

TABLE 5. A grid refinement analysis for Example 3.2 for interface as a sphere with $\beta^- = 1$ and $\beta^+ = 10^4$.

N	$E(u)$	order	$E(u_{\boldsymbol{n}})$	order	$E(u_\eta)$	order	$E(u_\tau)$	order
10	8.58E-02		1.24E-01		3.64E-01		3.80E-01	
20	2.42E-02	1.83	2.98E-02	2.06	9.29E-02	1.97	8.97E-02	2.08
40	6.35E-03	1.93	8.29E-03	1.84	2.79E-02	1.73	2.80E-02	1.68
80	1.62E-03	1.97	3.87E-03	1.10	9.90E-03	1.50	9.96E-03	1.49
160	4.09E-04	1.98	1.40E-03	1.46	2.34E-03	2.08	2.38E-03	2.07

TABLE 6. A grid refinement analysis for Example 3.2 for interface as a sphere with $\beta^- = 10^2$ and $\beta^+ = 1$.

N	$E(u)$	order	$E(u_{\boldsymbol{n}})$	order	$E(u_\eta)$	order	$E(u_\varsigma)$	order
10	4.42E-01		4.65E-01		5.45E-01		5.45E-01	
20	2.97E-01	1.31	1.23E-01	1.91	1.60E-01	1.77	1.66E-01	1.72
40	1.19E-01	1.82	2.13E-02	2.54	2.87E-02	2.47	3.10E-02	2.42
80	3.39E-02	1.95	6.04E-03	1.81	5.40E-03	2.41	6.13E-03	2.34
160	8.74E-03	1.95	1.74E-03	1.80	1.61E-03	1.75	1.60E-03	1.94

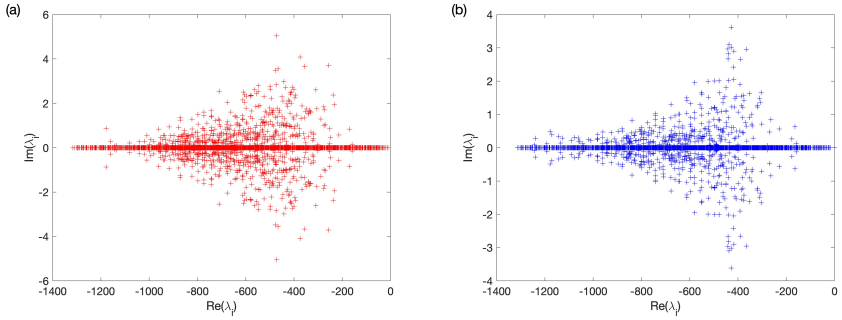

FIGURE 6. Eigenvalue distribution of the coefficient matrix for Example 3.2 with $N = 22$. Cases: (a) $\beta^- = 1$ and $\beta^+ = 100$, and (b) $\beta^- = 100$ and $\beta^+ = 1$. All eigenvalues exhibit negative real parts.

γ_m, which are coefficients in the underdetermined linear system (49). These coefficients are instrumental in computing the correction terms for a specific irregular grid point (x_i, y_j, z_l). To facilitate understanding, we labeled the 27 grid points in the FD scheme corresponding to this irregular grid point using red ink. The labeling sequence begins with the lowest left corner grid point as number 1, and it increases by one as it moves to the right and upward. Refer to Figure 7 for a visual representation of this labeling scheme.

As discussed in Section 2.6, to determine the coefficients τ_m of the FD scheme for a particular irregular point (labeled as number 14), we added the coefficients of the standard seven-point stencil of the FD scheme for a Poisson equation (4) in the absence of an interface. Specifically, the central point is assigned a coefficient of $-6/h^2 \approx -661.00$, while the remaining six grid points (labeled as numbers 5, 11, 13, 15, 17, and 23) are assigned a coefficient of $1/h^2 \approx 110.25$. This addition was performed to the γ_m values obtained from solving the linear system (49). Notably,

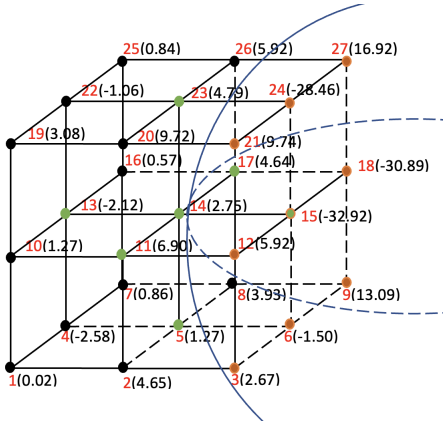

FIGURE 7. The γ_m coefficients for a point (x_i, y_j, z_l) near the interface in Example 3.2. Standard seven-point stencil grid points are indicated with green circles, interface grid points with orange, and other stencil points with black.

the larger coefficients in the FD scheme arise primarily from the values of the seven-point stencil, providing insight into the stability of the original method.

Condition number analysis: In addition to the eigenvalue analysis, we investigate the condition number for two scenarios involving β jumps for Example 3.2. The results are summarized in Table 7, where the interface is represented as a sphere. These results demonstrate that the condition number remains stable across various

TABLE 7. Condition number analysis for Example 3.2 with varying β ratios and spherical Interface.

N	$[\beta] = 100$	$[\beta] = 10^4$
10	128	128
20	566	567
40	2101	2105
80	8039	8053

scenarios, providing strong evidence of the robustness of our numerical method against different jumps in β.

Applicability for general interfaces: In this part of the analysis, we extend the applicability of the Direct IIM proposed in this paper to non-spherical interfaces, addressing more general surface interfaces. We utilize the same elliptic interface problem as defined in (66), with the identical boundary conditions as described in (69). However, the interface is modeled as an ellipsoid instead of a sphere, defined by the equation:

$$\phi(x, y, z) = x^2 + 2y^2 + z^2 - 0.25. \tag{70}$$

This modification allows us to test the method's effectiveness in the general interfaces. The grid refinement analysis for these scenarios is presented in Tables 8 through 11. These results indicate that the method still preserves second-order accuracy even with non-spherical interfaces.

Furthermore, we present a comparative analysis of the solution, its error, and interface representations for spherical and ellipsoidal cases in Figure 8. This analysis demonstrates the method's adaptability to different interface shapes.

Additionally, to assess the computational efficiency of our method for different interface geometries, we have conducted CPU time analysis for both spherical and ellipsoidal cases. The results in Table 12 illustrate the CPU times (in seconds) required for Example 3.2 with each geometry. This evaluation helps understand the method's performance in terms of computational demand for varying interface configurations.

TABLE 8. A grid refinement analysis for Example 3.2 with ellipsoid $\beta^+ = 1$ and $\beta^- = 1$.

N	$E(u)$	order	$E(u_{\boldsymbol{n}})$	order	$E(u_\eta)$	order	$E(u_\zeta)$	order
10	6.40E-02		1.09E-01		1.23E-01		1.23E-01	
20	1.83E-02	1.81	2.68E-02	2.02	4.05E-02	1.60	3.96E-02	1.63
40	4.85E-03	1.91	8.52E-03	1.65	1.14E-02	1.82	1.18E-02	1.74
80	1.25E-03	1.96	2.22E-03	1.94	2.71E-03	2.08	3.04E-03	1.96
160	3.16E-04	1.98	5.55E-04	2.00	6.99E-04	1.96	6.95E-04	2.13

TABLE 9. A grid refinement analysis for Example 3.2 with ellipsoid $\beta^+ = 10^3$ and $\beta^- = 1$.

N	$E(u)$	order	$E(u_{\boldsymbol{n}})$	order	$E(u_\eta)$	order	$E(u_\zeta)$	order
10	8.21E-02		1.44E-01		3.53E-01		4.08E-01	
20	2.30E-02	1.84	7.61E-02	0.91	9.84E-02	1.84	1.02E-01	2.00
40	6.02E-03	1.93	1.22E-02	2.64	2.24E-02	2.14	2.51E-02	2.02
80	1.54E-03	1.97	3.78E-03	1.69	7.40E-03	1.60	8.40E-03	1.58
160	3.90E-04	1.98	1.09E-03	1.80	2.22E-03	1.74	2.44E-03	1.78

TABLE 10. A grid refinement analysis for Example 3.2 with ellipsoid $\beta^+ = 10^4$ and $\beta^- = 1$.

N	$E(u)$	order	$E(u_{\boldsymbol{n}})$	order	$E(u_\eta)$	order	$E(u_\zeta)$	order
10	8.22E-02		1.44E-01		3.53E-01		4.08E-01	
20	2.30E-02	1.84	7.84E-02	0.87	9.87E-02	1.84	1.02E-01	2.00
40	6.02E-03	1.93	1.23E-02	2.68	2.25E-02	2.13	2.52E-02	2.02
80	1.54E-03	1.97	4.15E-03	1.56	7.41E-03	1.60	8.42E-03	1.58
160	3.90E-04	1.98	1.11E-03	1.90	2.22E-03	1.74	2.45E-03	1.78

Example 3.3.

In this example, both the solution $u(x, y, z)$ and its flux have jump discontinuities. The governing differential equation is given by:

$$(\beta u_x)_x + (\beta u_y)_y + (\beta u_z)_z = f. \tag{71}$$

The domain Ω is defined as $[-1, 1] \times [-1, 1] \times [-1, 1]$, and the interface Γ is a sphere represented by the zero level set of the function:

$$\phi(x, y, z) = x^2 + y^2 + z^2 - 0.25. \tag{72}$$

TABLE 11. A grid refinement analysis for Example 3.2 with ellipsoid $\beta^+ = 1$, $\beta^- = 10^2$.

N	$E(u)$	order	$E(u_{\boldsymbol{n}})$	order	$E(u_\eta)$	order	$E(u_\varsigma)$	order
10	1.76E-01		4.33E-01		3.95E-01		3.74E-01	
20	1.67E-01	0.08	3.75E-01	0.21	1.92E-01	1.04	1.94E-01	0.94
40	8.68E-02	0.94	9.67E-02	1.95	4.48E-02	2.10	4.61E-02	2.07
80	2.63E-02	1.72	3.68E-02	1.40	6.86E-03	2.71	6.50E-03	2.83
160	6.90E-03	1.93	1.05E-02	1.81	1.74E-03	1.98	1.65E-03	1.98

TABLE 12. CPU time in seconds for Example 3.2 with sphere and ellipsoid.

N	$[\beta] = 100$		$[\beta] = 10^4$	
	Sphere	Ellipsoid	Sphere	Ellipsoid
10	22.09	25.54	22.06	22.56
20	57.19	60.55	57.17	53.40
40	252.98	255.52	252.95	238.09
80	813.03	756.04	812.06	726.38

The source term f is defined as follows:

$$f(x,y,z) = \begin{cases} 3\beta^- e^{x+y+z} & \text{if } (x,y,z) \in \Omega^-, \\ -\beta^+\pi^2(\sin(\pi x) + \sin(\pi y) + \sin(\pi z)) & \text{if } (x,y,z) \in \Omega^+. \end{cases} \tag{73}$$

The jump in solution $u(x,y,z)$ and its flux is computed by the equation (74). Boundary conditions are defined by the exact solution:

$$u(x,y,z) = \begin{cases} e^{x+y+z} & \text{if } (x,y,z) \in \Omega^-, \\ \sin(\pi x) + \sin(\pi y) + \sin(\pi z) & \text{if } (x,y,z) \in \Omega^+. \end{cases} \tag{74}$$

TABLE 13. Grid refinement analysis for Example 3.3.

N	$[\beta] = 1$		$[\beta] = 5$		$[\beta] = 10$		$[\beta] = 50$	
	$E(u)$	Order	$E(u)$	Order	$E(u)$	Order	$E(u)$	Order
10	4.85E-02		9.77E-02		1.09E-01		1.18E-01	
20	1.45E-02	1.74	3.94E-02	1.31	4.53E-02	1.26	5.04E-02	1.22
40	3.74E-03	1.95	1.60E-02	1.30	1.88E-02	1.27	2.13E-02	1.24
80	1.03E-03	1.86	5.81E-03	1.46	8.60E-03	1.13	9.86E-03	1.11
160	2.65E-04	1.97	1.51E-03	1.94	2.39E-03	1.85	3.20E-03	1.62

TABLE 14. CPU time in seconds for varying values of $[\beta]$ for Example 3.3.

N	$[\beta] = 1$	$[\beta] = 10$	$[\beta] = 50$
10	40.17	26.95	24.76
20	150.27	69.24	64.06
40	713.69	301.04	283.76
80	852.53	947.23	902.97

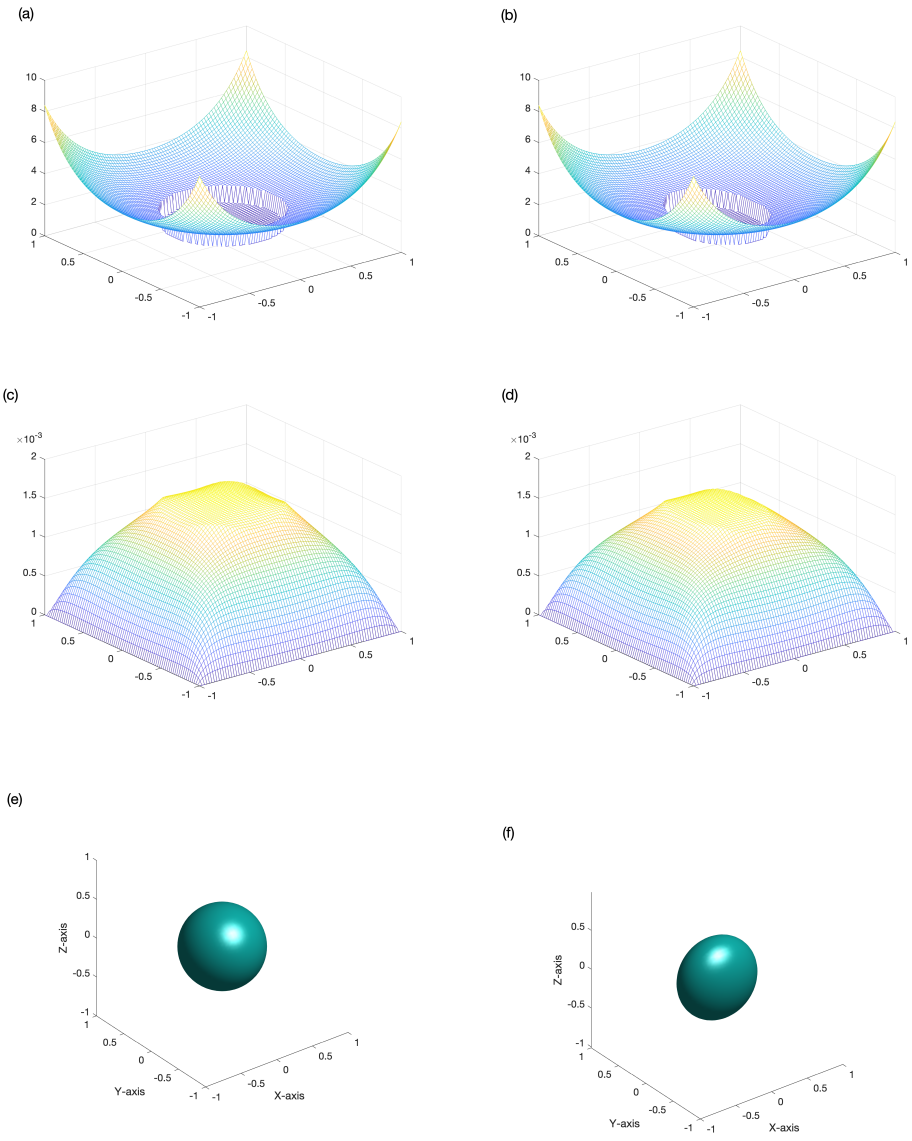

FIGURE 8. Comparative visualization for Example 3.2 with $[\beta] = 100$ and $N = 80$: (a) Numerical solution with a spherical interface, (b) Numerical solution with an ellipsoidal interface, (c) Error analysis for the spherical interface, (d) Error analysis for the ellipsoidal interface, (e) Interface representation with a sphere, and (f) Interface representation with an ellipsoid.

We have presented the grid refinement analyses and CPU time evaluations for Example 3.3 in Tables 13 and 14. We have also included Figures 9a and 9b, illustrating the numerical solution and its corresponding error for Example 3.3.

Our observations reveal that the error is of second order when the jump in β is zero or relatively small. We can still expect second-order accuracy for larger jump

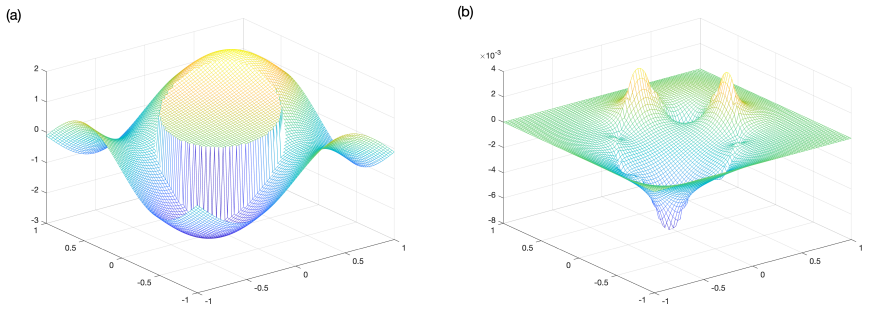

FIGURE 9. (a) Plot of numerical solution for Example 3.3, (b) Plot of error of the numerical solution with $N = 80$, $\beta^- = 1$ and $\beta^+ = 100$.

ratios when the meshes are fine enough. An alternative approach is to use the preconditioning technique described in [19]. Despite this, it successfully captures the discontinuities in the solution and maintains a decent order of accuracy.

4. Conclusion and future work

This paper introduces a direct methodology for addressing three-dimensional elliptic interface problems characterized by piecewise constant coefficients. This approach represents an advancement of the foundational work by Chen et al. [1], initially developed for two-dimensional elliptic interface problems. By adapting and expanding this method to three-dimensional scenarios, we ensure a user-friendly implementation accessible to a broad audience, including those not experts in the field.

The method exhibits second-order accuracy for moderate coefficient jumps. Stability analysis, including eigenvalue and condition number assessments, confirms the method's robustness against variations in the coefficient jump of β. We can still expect second-order accuracy for larger jump ratios when the meshes are fine enough. An alternative approach is to use the preconditioning technique described in [19].

Through comparative analysis, we demonstrated the method's adaptability to general interfaces, such as spherical and ellipsoidal shapes.

Additionally, we have conducted an extensive CPU analysis using the AMD Ryzen Threadripper PRO 5965WX. To further emphasize the practicality and accessibility of our method, we also carried out computational tests on a widely used personal computing device, a standard MacBook. This step not only demonstrates the versatility of our approach but also its adaptability to common computing environments.

We also plan to enhance the efficiency of our method by integrating a multigrid solver for solving linear systems within the finite difference (FD) scheme. This advancement aims to replace the current use of the backslash operator in MATLAB, offering a more robust and efficient computational framework.

References

[1] X. Chen, X. Feng, and Z. Li, A direct method for accurate solution and gradient computations for elliptic interface problems, Numer. Algorithms, vol. 80, no. 3, pp. 709–740, 2019.

[2] R. J. LeVeque and Z. Li, Immersed interface methods for stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput., vol. 18, no. 3, pp. 709–735, 1997.
[3] Z. Li, H. Zhao, and H. Gao, A numerical study of electro-migration voiding by evolving level set functions on a fixed cartesian grid, J. Comput. Phys., vol. 152, no. 1, pp. 281–304, 1999.
[4] D. F. M. Li, Zhilin and J. T. Heine, A numerical method for diffusive transport with moving boundaries and discontinuous material properties, Int. J. Numer. Anal. Methods Geomech., vol. 21, no. 9, pp. 653–662, 1997.
[5] A. L. Fogelson, A mathematical model and numerical method for studying platelet adhesion and aggregation during blood clotting, J. Comput. Phys., vol. 56, no. 1, pp. 111–134, 1984.
[6] B. Soni, Zhilin Li, Fast and accurate numerical approaches for stefan problems and crystal growth, Numer. Heat Transf. B: Fundam., vol. 35, no. 4, pp. 461–484, 1999.
[7] R. Cortez and M. Minion, The blob projection method for immersed boundary problems, J. Comput. Phys., vol. 161, no. 2, pp. 428–453, 2000.
[8] A.-K. Tornberg and B. Engquist, Regularization techniques for numerical approximation of pdes with singularities, J. Sci. Comput., vol. 19, no. 1-3, pp. 527–552, 2003.
[9] A.-K. Tornberg and B. Engquist, Numerical approximations of singular source terms in differential equations, J. Comput. Phys., vol. 200, no. 2, pp. 462–488, 2004.
[10] C. S. Peskin, The immersed boundary method, Acta Numer., vol. 11, pp. 479–517, 2002.
[11] A. M. Roma, C. S. Peskin, and M. J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys., vol. 153, no. 2, pp. 509–534, 1999.
[12] Z. Li, The immersed interface method using a finite element formulation, Appl. Numer. Math., vol. 27, no. 3, pp. 253–267, 1998.
[13] X.-D. Liu, R. P. Fedkiw, and M. Kang, A boundary condition capturing method for poisson's equation on irregular domains, J. Comput. Phys., vol. 160, no. 1, pp. 151–178, 2000.
[14] S. Yu, Y. Zhou, and G. Wei, Matched interface and boundary (mib) method for elliptic problems with sharp-edged interfaces, J. Comput. Phys., vol. 224, no. 2, pp. 729–756, 2007.
[15] I. Babuška, The finite element method for elliptic equations with discontinuous coefficients, Computing, vol. 5, no. 3, pp. 207–213, 1970.
[16] J. H. Bramble and J. T. King, A finite element method for interface problems in domains with smooth boundaries and interfaces, Adv. Comput. Math., vol. 6, no. 1, pp. 109–138, 1996.
[17] Z. Chen and J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math., vol. 79, no. 2, pp. 175–202, 1998.
[18] Z. Li, T. Lin, and X. Wu, New cartesian grid methods for interface problems using the finite element formulation, Numer. Math., vol. 96, no. 1, pp. 61–98, 2003.
[19] Z. Li, A fast iterative algorithm for elliptic interface problems, SIAM J. Math. Anal., vol. 35, no. 1, pp. 230–254, 1998.
[20] P. N. Swarztrauber, Fast poisson solver, In Studies in Numerical Analysis, vol. 24, pp. 319–370, 1984.
[21] R. J. LeVeque, Clawpack and amrclaw— software for high-resolution godunov methods, in 4-th ntl. Conf. on Wave Propagation, Citeseer, 1998.
[22] M. J. Berger and R. J. Leveque, Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems, SIAM J. Math. Anal., vol. 35, no. 6, pp. 2298–2316, 1998.
[23] S. Osher and R. P. Fedkiw, Level set methods and dynamic implicit surfaces, vol. 153. Springer, 2003.
[24] S. Osher and J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations, J. Comput. Phys., vol. 79, no. 1, pp. 12–49, 1988.
[25] J. A. Sethian, Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science, vol. 3. Cambridge university press, 1999.
[26] L. Adams, A multigrid algorithm for immersed interface problems, in NASA Conference Publication, pp. 1–14, 1996.
[27] P. M. De Zeeuw, Matrix-dependent prolongations and restrictions in a blackbox multigrid solver, J. Comput. Appl. Math., vol. 33, no. 1, pp. 1–27, 1990.
[28] Z. Li, An overview of the immersed interface method and its applications, Taiwan. J. Math., vol. 7, no. 1, pp. 1–49, 2003.
[29] R. J. LeVeque and Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Math. Anal., vol. 31, no. 4, pp. 1019–1044, 1994.

[30] K. Pan, Y. Tan, and H. Hu, An interpolation matched interface and boundary method for elliptic interface problems, J. Comput. Appl. Math., vol. 234, no. 1, pp. 73–94, 2010.
[31] Z. Li, D. Wang, and J. Zou, Theoretical and numerical analysis on a thermo-elastic system with discontinuities, J. Comput. Appl. Math., vol. 92, no. 1, pp. 37–58, 1998.
[32] Z. Li and X. Yang, An immersed finite element method for elasticity equations with interfaces, Contemp. Math., vol. 383, pp. 285–298, 2005.
[33] X. Yang, Immersed interface method for elasticity problems with interfaces. North Carolina State University, 2004.
[34] T. Y. Hou, Z. Li, S. Osher, and H. Zhao, A hybrid method for moving interface problems with application to the hele–shaw flow, J. Comput. Phys., vol. 134, no. 2, pp. 236–252, 1997.
[35] A. Wiegmann and K. P. Bube, The immersed interface method for nonlinear differential equations with discontinuous coefficients and singular sources, SIAM J. Math. Anal., vol. 35, no. 1, pp. 177–200, 1998.
[36] S. Jin and X. Wang, Robust numerical simulation of porosity evolution in chemical vapor infiltration: Ii. two-dimensional anisotropic fronts, J. Comput. Phys., vol. 179, no. 2, pp. 557–577, 2002.
[37] K. Ito, K. Kunisch, and Z. Li, Level-set function approach to an inverse interface problem, Inverse Probl., vol. 17, no. 5, p. 1225, 2001.
[38] G. Chen, Accurate Gradient Computation for Elliptic Interface Problems with Discontinuous and Variable Coefficients. PhD dissertation, North Carolina State University, 2015.
[39] K. Gamage, A Direct Method for Modeling and Simulations of Elliptic and Parabolic Interface Problems. PhD dissertation, Old Dominion University, 2022.

[1]Department of Mathematics, Washington and Lee University, 204 W Washington St, Lexington, VA 24450, USA
E-mail: kgamage@wlu.edu
URL: https://my.wlu.edu/directory/profile?ID=x28854

[2] Mathematics & Statistics, Old Dominion University, 5115 Hampton Blvd, Norfolk, VA 23529, USA
E-mail: YPeng@odu.edu
URL: https://www.odu.edu/directory/yan-peng

[3]Department of Mathematics, Box 8205, North Carolina State University, Raleigh, NC 27695-8205
E-mail: zhilin@math.ncsu.edu
URL: https://zhilin.math.ncsu.edu/