Accurate Gradient Approximation for Complex Interface Problems in 3D by an Improved Coupling Interface Method

Yu-Chen Shu[a], I-Liang Chern[b,d], Chien C. Chang[c,d]

[a]*Department of Mathematics National Cheng Kung University, Tainan 701, Taiwan*
[b]*Department of Applied Mathematics, National Chiao Tung University, Hsin Chu 300, Taiwan*
[c]*Institute of Applied Mechanics, National Taiwan University, Taipei 106, Taiwan*
[d]*Department of Mathematics National Taiwan University, Taipei 106, Taiwan*
[e]*Mathematics Division, National Center for Theoretical Sciences (Taipei Office), Taipei 106, Taiwan*
[f]*Mathematics Division, National Center for Theoretical Sciences(South), Tainan 701, Taiwan*

Abstract

Most elliptic interface solvers become complicated for complex interface problems at those "exceptional points" where there are not enough neighboring interior points for high order interpolation. Such complication increases especially in three dimensions. Usually, the solvers are thus reduced to low order accuracy. In this paper, we classify these exceptional points and propose two recipes to maintain order of accuracy there, aiming at improving the previous coupling interface method [26]. Yet the idea is also applicable to other interface solvers. The main idea is to have at least first order approximations for second order derivatives at those exceptional points. Recipe 1 is to use the finite difference approximation for the second order derivatives at a nearby interior grid point, whenever this is possible. Recipe 2 is to flip domain signatures and introduce a ghost state so that a second-order method can be applied. This ghost state is a smooth extension of the solution at the exceptional point from the other side of the interface. The original state is recovered by a post-processing using nearby states and jump conditions. The choice of recipes is determined by a classification scheme of the exceptional points. The method renders the solution and its gradient uniformly second-order accurate in the entire computed domain. Numerical examples are provided to illustrate the second order accuracy of the presently proposed method in approximating the gradients of the original states for some complex interfaces which we had tested previous in two and three dimensions, and a real molecule (1D63) which is double-helix shape and composed of hundreds of atoms.

Keywords: elliptic interface problems, coupling interface method, exceptional points, complex interfaces, second-order method for gradient
PACS: 41.20.Jb, 52.65.Ww, 02.60.Cb, 02.70.Bf

Email addresses: ycshu@mail.ncku.edu.tw (Yu-Chen Shu), chern@math.ntu.edu.tw (I-Liang Chern), mechang@iam.ntu.edu.tw (Chien C. Chang)

1. Introduction

The elliptic interface problem involves solving the following partial differential equation

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

where ε is positive and Γ is the interface, across which, the coefficient $\varepsilon(\cdot)$ is discontinuous. On the interface, two interface conditions are given:

$$[u] = \tau, \ [\varepsilon\nabla u \cdot \mathbf{n}] = \sigma \text{ on } \Gamma, \tag{2}$$

where $\mathbf{n}$ is the normal direction to the interface, see Fig. 1. On the boundary $\partial\Omega$, the Dirichlet boundary condition is applied. Such elliptic interface problems appear commonly in two phase flows in fluid dynamics, fluid-structure interaction, electromagnetism, bio-mechanics, etc., see review articles [1, 2].

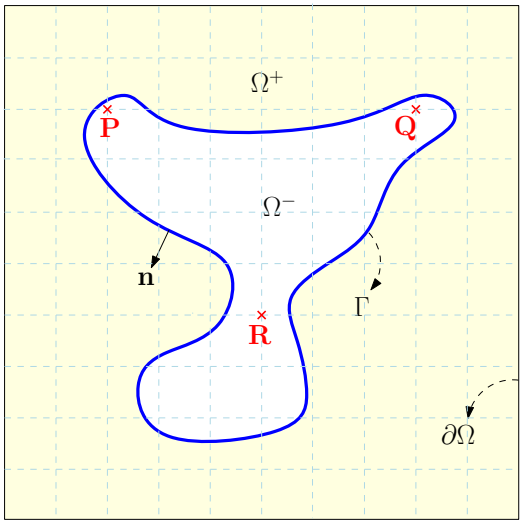


Figure 1: Domain for interface problems. The blue curve Γ is the interface which separates the domain Ω into Ω^- and Ω^+. The normal to the interface is denoted as $\mathbf{n}$. The light blue lines are the grid lines of the uniform mesh. Red cross markers labeled by **P**, **Q**, and **R** are called exceptional points since there are not enough interior grid points on the same side for high-order interpolation.

There are many elliptic interface numerical solvers. For example, for boundary integral approaches, we refer to [3]; for body-fitting approaches, we refer to [4, 5]; for finite element approaches, we refer readers to [6, 7, 8, 9], and for finite volume approaches, we refer to [10]. When there is a moving interface, the finite difference approaches are most suitable. The basic idea is to embed the interface into a Cartesian grid, in which a standard finite difference method can be applied. We roughly classify them into two categories, the dimensional splitting and un-splitting methods. For the un-splitting methods, we refer readers to [11, 12] for regularization methods; [1, 13] for the immersed boundary method; [14, 15, 2] for the immersed interface method (IIM); [16] for the multidimensional piecewise polynomials method, and [17, 18] for the virtual node method. For the dimension splitting methods, we refer readers to [19, 20, 21, 22, 23, 24, 25]. Our previous

work [26], the coupling interface method (CIM) belongs to the latter. The convergence and accuracy analysis of finite difference approaches can be found in [27, 28].

In some moving interface problems, accurate approximation for gradients is necessary [29, 30, 31, 32]. A straight forward way is to take numerical differentiation from nearby computed solutions, but this approach lowers down the order of accuracy by one, see, for example, the convergence results in [18]. Even worse, a high order approximation for gradients may not be possible at those points where there are not enough neighboring grid points available on the same side for interpolation, such as the **P**, **Q** and **R** in Fig.1. We call such points *exceptional points*. Many of such exceptional points are generic, which means that they cannot be removed by refining meshes, and they are typical in three dimensions. The magnitudes of errors thus depend on how the interfaces cut through the underlying regular grids. One observes that the error fluctuates against the mesh size.

In this paper, we classify these exceptional points and propose recipes to maintain high order accuracy there. These recipes are to improve the previous method, the coupling interface method [26], but the idea is also applicable to other elliptic interface solvers. The main idea is the follows. First, a suitable mesh size is suggested based on the maximal curvature of the interface to eliminate certain kinds of exceptional points other than the generic ones. Then, we propose two recipes to have first order approximation for second order derivatives on generic exceptional points. Recipe 1 uses the finite difference approximation for the second order derivatives at an adjacent interior grid, if it is available. Recipe 2 is to flip the domain signature of the exceptional point so that it is no longer an exceptional point and a second-order method can be applied. The state so computed is a ghost state, which is the smooth extension of the solution at the exceptional point from the other side of the interface. The original state can be recovered by a post-processing using nearby states and jump conditions. The idea of ghost states has been introduced in [21]. The difference is that the state in [21] is used as an intermediate state during discretization, while our ghost states are computed in the discrete systems. The choice of which recipe to use is determined by a classification of the exceptional points. The improved method renders the solution and its gradient uniformly second-order accurate in the entire computed domain. Careful numerical tests are performed to show a second-order accuracy for gradients in two and three dimensions, including a real molecule (1D63) composed of hundreds of atoms.

1.1. Brief review of the Coupling Interface Method

In our previous work [26], we proposed the coupling interface method, which is a dimension splitting method based on a Cartesian grid. In each dimension, it uses piecewise polynomials to approximate the solution from both sides of the interface. The information from different dimensions is linked through jump conditions, leading to a coupled linear equation for the principal second order derivatives. This approach reduces the size of stencils, and thus has mild restriction on the interfaces. This gives it more potential to handle complex interface problems. Indeed, the first-order method (CIM1) only requires the interface intersects each grid segment at most once. The second-order method (CIM2) basically requires that at an interface point, there are two adjacent grid points sitting in the same region on each side of the interface in each dimensional direction. For the convenience of readers, we summarize the key procedures of CIM2 below. The high-order approximation for the gradient of the solution is also included.

Suppose that $\Omega = [0,1]^d$ in $d > 1$ dimensions and the underlying Cartesian grid is uniform with mesh size $h = 1/N$, where N is the number of intervals in one edge of the cube Ω. Let $\mathbf{x_i} = \mathbf{i}h$ be a grid point, where the symbol $\mathbf{i} = (i_1, i_2, \cdots, i_d)$ is a multi-index with $i_k \in \mathbb{N} \cup \{0\}$. Let $\mathbf{e}_k$, $k = 1, 2, \cdots, d$ denote the unit coordinate vectors which we call the coordinate directions. We call $\mathbf{x_{i \pm e_k}}$ the adjacent grid points of $\mathbf{x_i}$.

Let us suppose the interface Γ cuts the domain Ω into two regions $\Omega^-(\phi(\mathbf{x}) < 0)$ and $\Omega^+(\phi(\mathbf{x}) > 0)$. We assume that Γ intersects any grid segment $\overline{\mathbf{x_j x_{j+e_k}}}$ at most once. Let us call the intersection the interface point. A grid point $\mathbf{x_i}$ is called an *interior point* if all its adjacent points are in the same region of $\mathbf{x_i}$. The rest of the grid points are called on-front points. They are further classified into *regular on-front points* (where CIM2 can be applied) and *exceptional points*. Their definitions will be clear in the description of CIM2 below.

Now, let us briefly explain the discretization procedure of CIM2 at a regular on-front point $\mathbf{x_j} \in \Omega^-$. We will only explain the case when for all k, $\mathbf{x_{j-e_k}}$ are in the same region of $\mathbf{x_j}$ and $\mathbf{x_{j+e_k}}$ are not. The discretization of (1) by CIM2 at $\mathbf{x_j}$ proceeds in three steps.

1. Dimension-by-dimension discretization: We apply the one-dimensional CIM2 at $\mathbf{x_j}$ along each coordinate direction.

$$\frac{\partial^2 u}{\partial x_k^2}(\mathbf{x_j}) = \frac{1}{h^2}\left(\mathcal{L}_{\mathbf{j},k}(u_{\mathbf{j-e}_k}, u_{\mathbf{j}}, u_{\mathbf{j+e}_k}, u_{\mathbf{j+2e}_k}) + b_{\mathbf{j},k}[u]_{\hat{\mathbf{x}}_k} + c_{\mathbf{j},k} h [\varepsilon \nabla u \cdot \mathbf{e}_k]_{\hat{\mathbf{x}}_k}\right) + O(h), \tag{3}$$

   where $\mathcal{L}_{\mathbf{j},k}(u_{\mathbf{j-e}_k}, u_{\mathbf{j}}, u_{\mathbf{j+e}_k}, u_{\mathbf{j+2e}_k})$ is a linear combination of $u_{\mathbf{j-e}_k}$, $u_{\mathbf{j}}$, $u_{\mathbf{j+e}_k}$, and $u_{\mathbf{j+2e}_k}$. This one dimensional formula can be derived using piecewise parabola approximation and the two jump conditions, see [26]

2. Representation of one-dimensional jump data in terms of the multi-dimensional jump data and the one-side gradient: This gives

$$[\varepsilon \nabla u \cdot \mathbf{e}_k]_{\hat{\mathbf{x}}_k} = \mathbf{n}_k \cdot \mathbf{e}_k [\varepsilon \nabla u \cdot \mathbf{n}_k]_{\hat{\mathbf{x}}_k} + \mathbf{t}_k \cdot \mathbf{e}_k \left(\varepsilon(\hat{\mathbf{x}}_k^+)[\nabla u \cdot \mathbf{t}_k]_{\hat{\mathbf{x}}_k} + [\varepsilon]_{\hat{\mathbf{x}}_k} \nabla u^-(\hat{\mathbf{x}}_k) \cdot \mathbf{t}_k\right), \tag{4}$$

   where $\mathbf{t}_k$ is the unit vector parallel to the projection of $\mathbf{e}_k$ onto the tangent plane of Γ at $\hat{\mathbf{x}}_k$.

3. Approximation of the one-side gradient and the cross derivatives:

$$\nabla u^-(\hat{\mathbf{x}}_k) = \frac{1}{h}\left[u_{\mathbf{j}} - u_{\mathbf{j-e}_\ell} + \frac{h^2}{2}\frac{\partial^2 u}{\partial x_\ell^2}(\mathbf{x_j}) + \alpha_k h^2 \frac{\partial^2 u}{\partial x_k \partial x_\ell}(\mathbf{x_j})\right]_{\ell=1}^{d} + O(h^2), \tag{5}$$

   where $\alpha_k h = \|\hat{\mathbf{x}}_k - \mathbf{x_j}\|$. Here we make another constraint for an on-front point, namely, $\mathbf{x_{j-e_k-e_\ell}}$ is also in the same region of $\mathbf{x_i}$. Then the above cross derivatives can be simply approximated by

$$\frac{\partial^2 u}{\partial x_k \partial x_\ell}(\mathbf{x_j}) = \frac{1}{h^2}\left(u_{\mathbf{j}} - u_{\mathbf{j-e}_k} - u_{\mathbf{j-e}_\ell} + u_{\mathbf{j-e}_k-\mathbf{e}_\ell}\right) + O(h). \tag{6}$$

   Note that we use (5) and (6) to approximate the gradient of the solution after u is solved. In [26], we showed the second-order accuracy for the gradient of solution when there are no exceptional points in the computational domain.

After substituting (4), (5) and (6) in (3), a linear system for the principal second order derivatives is obtained:

$$\mathbf{M}\left[\frac{\partial^2 u}{\partial x_k^2}(\mathbf{x_j})\right]_{k=1}^{d} = \frac{1}{h^2}\left[(\mathcal{L}_{\mathbf{j},k} + \mathcal{T}_{\mathbf{j},k})\, u_{\mathbf{j}} + \mathcal{J}_{\mathbf{j},k}\right]_{k=1}^{d} + O(h), \tag{7}$$

where $\mathcal{L}_{\mathbf{j},k}$, $\mathcal{T}_{\mathbf{j},k}$ and $\mathcal{J}_{\mathbf{j},k}$ are

$$\mathcal{L}_{\mathbf{j},k} u_{\mathbf{j}} = \mathcal{L}_{\mathbf{j},k}(u_{\mathbf{j}-\mathbf{e}_k}, u_{\mathbf{j}}, u_{\mathbf{j}+\mathbf{e}_k}, u_{\mathbf{j}+2\mathbf{e}_k}), \tag{8}$$

$$\mathcal{T}_{\mathbf{j},k} u_{\mathbf{j}} = c_{\mathbf{j},k}[\varepsilon]_{\hat{\mathbf{x}}_k} \mathbf{t}_k \cdot \mathbf{e}_k \sum_{\ell=1}^{d} \left(u_{\mathbf{j}} - u_{\mathbf{j}-\mathbf{e}_\ell} + \alpha_k h^2 \frac{\partial^2 u}{\partial x_k \partial x_\ell}(\mathbf{x_j}) \right) \mathbf{t}_k \cdot \mathbf{e}_\ell, \tag{9}$$

$$\mathcal{J}_{\mathbf{j},k} = b_{\mathbf{j},k}[u]_{\hat{\mathbf{x}}_k} + c_{\mathbf{j},k} h \left([\varepsilon \nabla u \cdot \mathbf{n}]_{\hat{\mathbf{x}}_k} \mathbf{n}_k \cdot \mathbf{e}_k + \varepsilon(\hat{\mathbf{x}}_k^+)[\nabla u \cdot \mathbf{t}_k]_{\hat{\mathbf{x}}_k} \mathbf{t}_k \cdot \mathbf{e}_k \right), \tag{10}$$

and the entries of the matrix $\mathbf{M}$ are

$$\mathbf{M}_{k,\ell} = \delta_{k,\ell} - \sigma_k [\varepsilon]_{\hat{\mathbf{x}}_k} \left(\frac{1}{2} + \alpha_k \delta_{k,\ell} \right) (\mathbf{t}_k \cdot \mathbf{e}_k)(\mathbf{t}_k \cdot \mathbf{e}_\ell). \tag{11}$$

Here $\delta_{k,\ell}$ is 1 for $k = \ell$ and 0 otherwise. It has been shown in [26] that if the maximum curvature of the interface is not too large (say,$O(1/h)$), then $\mathbf{M}$ is invertible. Thus, we obtain a first-order approximation of the principal second order derivatives at $\mathbf{x_j}$, and thus a first order approximation for Eq. (1) at $\mathbf{x_j}$.

As we can see the requirements for a *regular on-front point* $\mathbf{x_j}$ are

(i) for each coordinate direction, there are two grid points (including $\mathbf{x_j}$) on the same side and two grids on the opposite side;

(ii) there is at least one grid point, say $\mathbf{x}_{\mathbf{j}-\mathbf{e}_k-\mathbf{e}_\ell}$, available on the same side to compute cross derivative $\partial^2 u/\partial x_k \partial x_\ell$.

Those grid points which do not meet the above requirement are called *exceptional points*. An exceptional point is called *type-I* if it does not meet the requirement (i), and *type-II* if it satisfies (i) but not (ii).

2. Classification of type-I exceptional points and the corresponding recipes

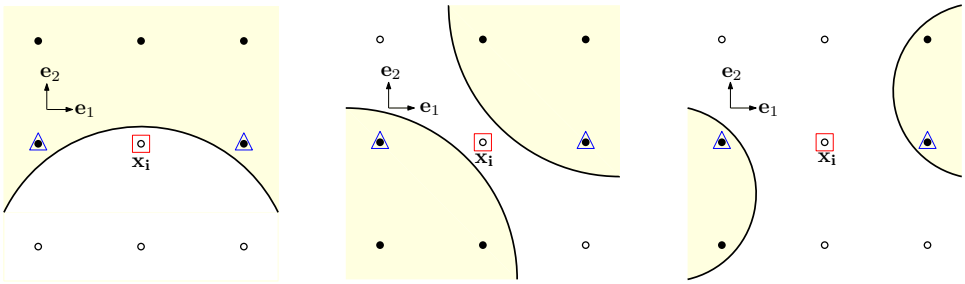
2.1. An indicator to classify type-I exceptional points

The treatments of type-I exceptional points are based on the following indicator $\mathbf{G}(\mathbf{x_i})$ which measures the number of adjacent grid points in the same region. Let $\mathbf{x_i}$ be a grid point, we define a vector $\mathbf{G}(\mathbf{x_i}) = (g_1(\mathbf{x_i}), g_2(\mathbf{x_i}), \cdots, g_d(\mathbf{x_i}))$ by:

$$g_k(\mathbf{x_i}) = \begin{cases} 2, & \text{if } \mathbf{x}_{\mathbf{i}-\mathbf{e}_k}, \mathbf{x_i} \text{ and } \mathbf{x}_{\mathbf{i}+\mathbf{e}_k} \text{ are in the same region;} \\ 0, & \text{if } \mathbf{x}_{\mathbf{i}-\mathbf{e}_k} \text{ and } \mathbf{x}_{\mathbf{i}+\mathbf{e}_k} \text{ are not in the region that } \mathbf{x_i} \text{ belongs to;} \\ 1, & \text{otherwise.} \end{cases} \tag{12}$$

In other words, $g_k(\mathbf{x_i}) = 2$ (resp. 1, 0) means that there are two (resp. one, zero) neighboring grid points of $\mathbf{x_i}$ in $\mathbf{e}_k$-direction are on the same side. It is obvious that a grid point satisfies condition (i) above if and only if $g_k(\mathbf{x_i})$, $g_k(\mathbf{x}_{\mathbf{i}\pm\mathbf{e}_k}) \geq 1$ for all k. Thus, a

grid point violates condition (i) (that is, a type I point) if and only if it satisfies either (i) $g_k(\mathbf{x_i}) = 0$ or (ii) $g_k(\mathbf{x_{i-e_k}}) = 0$ or (iii) $g_k(\mathbf{x_{i+e_k}}) = 0$ for some k. Below, we discuss their treatments in two and three dimensions separately. For simplicity of explanation, we may assume $g_1(\mathbf{x_i}) \le g_2(\mathbf{x_i}) \le g_3(\mathbf{x_i})$. Other cases are simply an orientation or rotation of this case.



(a) $G(\mathbf{x_i}) = (0, 1)$ (b) $G(\mathbf{x_i}) = (0, 0)$ (c) $G(\mathbf{x_i}) = (0, 2)$

Figure 2: Type-I exceptional points in two dimensions. The region colored by light yellow is Ω^+ while the white region is Ω^-. The circles and disks are grid points in Ω^- and Ω^+, respectively.

2.2. Type-I exceptional points in two dimensions with $g_1(\mathbf{x_i}) = 0$

There are three sub-cases, namely $G(\mathbf{x_i}) = (0, 1), (0, 0)$, and $(0, 2)$. See Figs. 2(a), 2(b) and 2(c). The grid points marked by squares are the exceptional points under discussion.

Case $G(\mathbf{x_i}) = (0, 1)$, see Fig.2(a). This situation happens when a grid point is near an interface point whose tangent is nearly perpendicular to some coordinate direction. In this case, we cannot obtain first order approximations for $u_{x_1x_1}(\mathbf{x_i})$ and $u_{x_1x_2}(\mathbf{x_i})$ by CIM2. We provide two recipes to get an $O(h)$ approximation for the second order derivatives at $\mathbf{x_i}$.

- **Recipe 1.** The first recipe is to replace the second order derivatives $u_{x_1x_1}(\mathbf{x_i})$ by an adjacent one where an $O(h)$ approximation is available. More precisely, we replace $u_{x_1x_1}(\mathbf{x_i})$ by $u_{x_1x_1}(\mathbf{x_{i-e_2}})$ in (3), that is,

$$\frac{\partial^2 u}{\partial x_1^2}(\mathbf{x_i}) = \frac{\partial^2 u}{\partial x_1^2}(\mathbf{x_{i-e_2}}) + O(h) = \frac{1}{h^2}(u_{\mathbf{i-e_2-e_1}} - 2u_{\mathbf{i-e_2}} + u_{\mathbf{i-e_2+e_1}}) + O(h), \quad (13)$$

  Meanwhile we also replace $\frac{\partial^2 u}{\partial x_1 \partial x_2}(\mathbf{x_i})$ by $\frac{\partial^2 u}{\partial x_1 \partial x_2}(\mathbf{x_{i-e_2}})$ in (6). These replacements are then used in the Eq. (7) for later derivation of $\frac{\partial^2 u}{\partial x_2^2}$ in CIM2. There may be more than one possible adjacent point to choose, or no such adjacent point available. In the former case, we choose the one with the smallest stencil size. In the latter case, we select further away adjacent points until an $O(h)$ approximation is available. In practice, we have never meet this case.

  It should be cautious that the resulting matrix may have bad condition number after such replacement. In order to study this issue, we consider such a replacement on

the standard discrete Laplacian in the domain $[0,1]\times[0,1]$. The standard discrete Laplacian can also be approximated by one of the following four discretizations derived by shifting one grid point (i.e. recipe 1).

$$\begin{bmatrix} 1 & -1 & 1 \\ 0 & -2 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 1 & -1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ -1 & -2 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 1 & -2 & -1 \\ 0 & 0 & 1 \end{bmatrix}. \tag{14}$$

In [26], we explained that, in practice, the number of exceptional points in of $O(1)$, independent of mesh sizes. To mimic such situations, we randomly choose four grid points in the domain and replace the discrete Laplacian by one of the above discretizations randomly. We repeat the test 100 times and see the difference of the condition number between the discrete Laplacian and the resulting matrix which recipe 1 is applied at four random grid points. The box-and-whisker diagram for the differences is shown in Figure 3. We use three mesh sizes: $h = 1/20$, $1/40$, and $1/80$. The condition numbers for discrete Laplacian are about 161.45, 647.79, and 2593.16 for $h = 1/20$, $1/40$, and $1/80$, respectively. Some differences are greater than the first quartile plus 1.5 times the interquartile range when h is $1/20$ and $1/40$, see the plus markers in Figure 3. It is due to the locations of those random points are too closed. Even with that, we found that the difference is almost $O(1)$ when the mesh size h decreases. It shows that the change of condition number is of $O(1)$ in average and independent of the mesh size if we apply the recipe 1 only at few exceptional points.

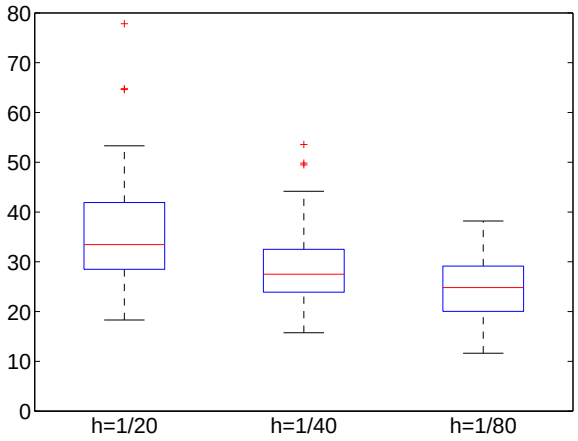


Figure 3: Boxplot of 100 tests for the difference of the condition number between the standard discrete Laplacian and the resulting matrix which recipe 1 is applied at four random grid points. Here we use three mesh sizes: $h = 1/20$, $1/40$, and $1/80$. The condition numbers for standard discrete Laplacian are about 161.45, 647.79, and 2593.16 for $h = 1/20$, $1/40$, and $1/80$, respectively. It shows the change of the condition number is of $O(1)$ and independent of the mesh size.

- **Recipe 2.** This is an alternative recipe to treat the exceptional point as shown in Fig. 2(a). First, we flip the domain signature of $\mathbf{x_i}$ from Ω^- to Ω^+. That is, we treat $\mathbf{x_i}$ as if it were on the Ω^+ side. After flipping, $\mathbf{x_i}$ becomes a regular on-front points,

and so do its two adjacent points. Thereby we can apply CIM2 at these points, as will be shown in detail below. The state so computed, denoted by $\tilde{u}(\mathbf{x_i})$, is called a *ghost state*. It is a smooth extrapolation of u from Ω^+-side. The advantage of this approach is that all states so computed are second order accurate. The original states will be recovered by a post processing.

In detail, as shown in Figure 2(a), after flipping the domain signature, the point $\mathbf{x_i}$ is no longer an exceptional point. Therefore, we can approximate $\frac{\partial^2 \tilde{u}}{\partial x_1^2}$ easily by the central finite differencing:

$$\frac{\partial^2 \tilde{u}}{\partial x_1^2} = \frac{1}{h^2}(u_{\mathbf{i}-\mathbf{e}_1} - 2\tilde{u}_{\mathbf{i}} + u_{\mathbf{i}+\mathbf{e}_1}) + O(h^2). \tag{15}$$

Along the e_2 direction, we can approximate $\frac{\partial^2 \tilde{u}}{\partial x_2^2}$ by the one-dimensional CIM2 formula:

$$\frac{\partial^2 \tilde{u}}{\partial x_2^2}(\mathbf{x_i}) = \frac{1}{h^2}\left(\mathcal{L}_{\mathbf{i},k}(u_{\mathbf{i}-2\mathbf{e}_2}, u_{\mathbf{i}-\mathbf{e}_2}, \tilde{u}_{\mathbf{i}}, u_{\mathbf{i}+\mathbf{e}_2}) + b_{2,k}[u]_{\hat{\mathbf{x}}_k} + c_{2,k}h[\varepsilon \nabla u \cdot \mathbf{e}_2]_{\hat{\mathbf{x}}_2}\right) + O(h). \tag{16}$$

Here, we treat $\mathbf{x_i}$ and $\mathbf{x}_{\mathbf{i}+\mathbf{e}_2}$ are in domain Ω^+, while $\mathbf{x}_{\mathbf{i}-\mathbf{e}_2}$ and $\mathbf{x}_{\mathbf{i}-2\mathbf{e}_2}$ in domain Ω^-. Note that the relative distance becomes negative or greater than 1 in the approximation. The issue for the condition number and the corresponding treatment are discussed later. Up to this point, we get the dimension-by-dimension discretization at the first step in CIM2. After the second and third steps in CIM2, we can get an $O(h)$ approximation for $\frac{\partial^2 \tilde{u}}{\partial x_2^2}(\mathbf{x_i})$.

Notice that $\tilde{u}_{\mathbf{i}}$ is expected to be the extrapolation of u from Ω^+-side. Therefore, we should also replace the source term of Eq. (1) by an extrapolation of f from Ω^+-side, see Fig. 2(a). For instance, the source term at $\mathbf{x_i}$ may be approximated by

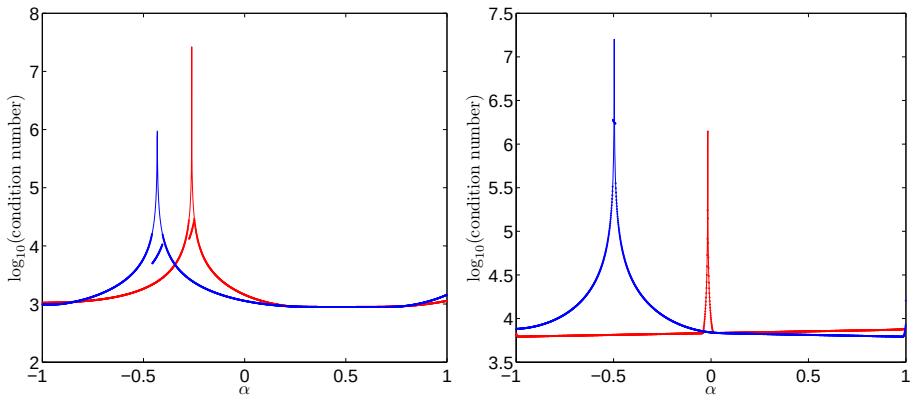
$$\tilde{f}_{\mathbf{i}} = \frac{1}{2}\left(f_{\mathbf{i}-\mathbf{e}_1} + f_{\mathbf{i}+\mathbf{e}_1}\right) + O(h^2). \tag{17}$$

Since the flips affect the classification of other grid points, we design a global flipping process as the follow. First, we look over all exceptional points with $\mathbf{G}(\mathbf{x}) = (0,0)$, $(0,1)$, $(1,0)$ in two dimensions and $\mathbf{G}(\mathbf{x}) = (0,0,0)$, $(0,0,1)$, $(0,1,0)$, or $(1,0,0)$ in three dimensions, see Table 1. We flip the domain signatures of these points simultaneously. We repeat this global flipping process until no more such points exist or a maximum number of iterations is reached. If the latter cases do occur, we apply recipe 1 at such grid points. Otherwise, recipe 2 is applied.

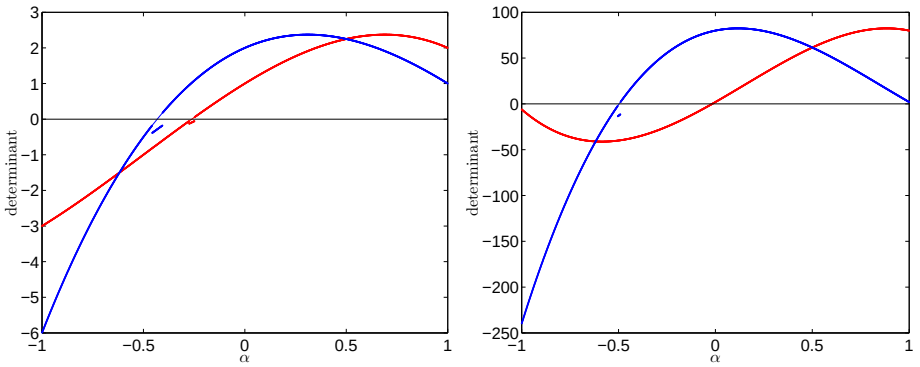
We remark that such flipping may produce a matrix with bad condition number due to the determinant of the coupling matrix $\mathbf{M}$ of Eq. (11) may be near zero. We illustrate this phenomenon and propose a threshold for det $\mathbf{M}$ to avoid such situations by the following simple example. We consider the one dimensional elliptic interface problem on domain $[-1,1]$. We discretize it into $2N$ subintervals uniformly. We assume that the interface point is situated at αh, where $h = 1/2N$. We apply the flipping domain recipe at $x_N = 0$. Figure 4 (a), (b) show the condition numbers (log-scale) versus α with $\alpha \in (-1,1)$ for the case $\varepsilon^- = 1$ and $\varepsilon^+ = 2$ and

the case $\varepsilon^- = 2$ and $\varepsilon^+ = 80$, respectively. Figure 4 (c), (d) are the corresponding determinant of $\mathbf{M}$. We see that the condition numbers are poor when det $\mathbf{M}$ is closed to zero. Otherwise, the condition numbers are well controlled if det $\mathbf{M}$ is away from zero.

From this study, we set up a threshold, say 0.1265, for $|\det \mathbf{M}|$ for the case $\varepsilon^- = 2$ and $\varepsilon^+ = 80$ from our experiments. If $|\det \mathbf{M}|$ is less than this threshold value, then we apply recipe 1 instead. We adopt GMRES[36] as a smoother in the algebraic multigrid solver[37, 38, 39] to solve the corresponding system, with the control of det $\mathbf{M}$, the computation is relatively stable.



(a) condition number for $\varepsilon^- = 1$ and $\varepsilon^+ = 2$ (b) condition number for $\varepsilon^- = 2$ and $\varepsilon^+ = 80$



(c) determinant for $\varepsilon^- = 1$ and $\varepsilon^+ = 2$ (d) determinant for $\varepsilon^- = 2$ and $\varepsilon^+ = 80$

Figure 4: The semi-log plot of the condition number versus the interface location α. The recipe 2 is applied when $\alpha < 0$ at the middle of the domain.

The state $\tilde{u}_{\mathbf{i}}$ so obtained is the extrapolated state at $\mathbf{x}_{\mathbf{i}}$, but all states nearby are second order accurate. We can recover the original state $u_{\mathbf{i}}$ by

$$u_{\mathbf{i}} = 2u_{\mathbf{i}-\mathbf{e}_2} - u_{\mathbf{i}-2\mathbf{e}_2} + h^2 \frac{\partial^2 u}{\partial x_2^2}(\mathbf{x}_{\mathbf{i}-\mathbf{e}_2}) + O(h^3), \tag{18}$$

where $\frac{\partial^2 u}{\partial x_2^2}(\mathbf{x}_{\mathbf{i}-\mathbf{e}_2})$ is replaced by an $O(h)$ approximation derived by CIM2 proce-

dures that we mentioned before with the help of the ghost state $\tilde{u}_{\mathbf{i}}$.

Case $G(\mathbf{x_i}) = (0,0)$, see Fig.2(b). The above flipping domain-signature approach can also be used for the case $\mathbf{G}(\mathbf{x_i}) = (0,0)$. By flipping domain signature, we can approximate $\frac{\partial^2 \tilde{u}}{\partial x_k^2}(\mathbf{x_i})$ by the standard central finite difference method. Then $\tilde{u}$ can be solved and thereby $u|_\Omega^+$ can be found. To reconstruct $u_{\mathbf{i}}$, we solve a local discrete Poisson equation at $\mathbf{x_i}$ with boundary being the four neighboring interface intersections, at which $u|_\Omega^-$ can be obtained from $u|_\Omega^+$ and the jump conditions.

Case $G(\mathbf{x_i}) = (0,2)$, see Fig.2(c). In this case, we adopt recipe 1 above. We may approximate $\frac{\partial^2 u}{\partial x_1^2}(\mathbf{x_i})$ by searching the approximations at $\mathbf{x_{i\pm e_2}}$, $\mathbf{x_{i+e_1+e_2}}$, and $\mathbf{x_{i-e_1-e_2}}$.

2.3. Type-I exceptional points in two dimensions with $g_1(\mathbf{x_{i\pm e_1}}) = 0$.

Let us discuss the case $g_1(\mathbf{x_{i+e_1}}) = 0$ only. The other case has the same treatment. The grid $\mathbf{x_i}$ is adjacent to another type-I exceptional point $\mathbf{i}+\mathbf{e}_1$ with $g_1(\mathbf{x_{i+e_1}}) = 0$. In this case, $g_1(\mathbf{x_i}) = 0, 1$ or 2. We find that $g_1(\mathbf{x_i}) \neq 2$ because it contradicts $g_1(\mathbf{x_{i+e_1}}) = 0$. If $g_1(\mathbf{x_i}) = 0$, it will be handled in the previous subsection. Thus, the only case under discussion is $g_1(\mathbf{x_i}) = 1$. This corresponds to those triangle markers in Fig. 2.

Our goal is to find a first-order approximation for $\frac{\partial^2 u}{\partial x_1^2}(\mathbf{x_i})$. We have two recipes for the grid point $\mathbf{x_{i+e_1}}$. The first recipe is to adopt recipe 2 at $\mathbf{x_{i+e_1}}$, from which we get a ghost state $\tilde{u}_{\mathbf{i}+\mathbf{e}_1}$. We then approximate $\frac{\partial^2 u}{\partial x_k^2}(\mathbf{x_i})$ by

$$\frac{\partial^2 u}{\partial x_1^2}(\mathbf{x_i}) = \frac{1}{h^2}\left(u_{\mathbf{i}-\mathbf{e}_1} - 2u_{\mathbf{i}} + \tilde{u}_{\mathbf{i}+\mathbf{e}_1}\right) + O(h^2). \tag{19}$$

If recipe 1 is used at $\mathbf{x_{i+e_1}}$, we notice that $\mathbf{x_{i-e_1}}$, $\mathbf{x_i}$ and $\mathbf{x_{i+2e_1}}$ are in the same region. Therefore, we can get a first order approximation for $\frac{\partial^2 u}{\partial x_1^2}(\mathbf{x_i})$ by

$$\frac{\partial^2 u}{\partial x_1^2}(\mathbf{x_i}) = \frac{1}{h^2}\left(\frac{2}{3}u_{\mathbf{i}-\mathbf{e}_1} - u_{\mathbf{i}} + \frac{1}{3}u_{\mathbf{i}+2\mathbf{e}_1}\right) + O(h). \tag{20}$$

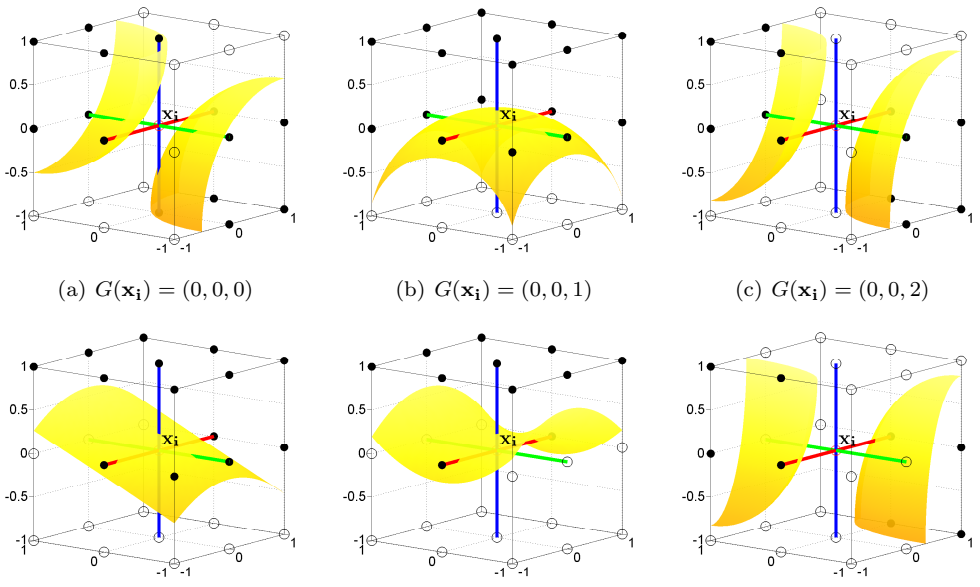
At this point, we have completely solved all the cases of type-I exceptional points in two dimensions.

2.4. Type-I exceptional points in three dimensions

For the study in three dimensions, without lose of generality, we may assume that $g_1 \leq g_2 \leq g_3$ at the exceptional point for simple explanation. Under this assumption, we recall that $\mathbf{x_i}$ is an exceptional point if and only if $g_1(\mathbf{x_i}) = 0$, or $g_1(\mathbf{x_{i-e_1}}) = 0$, or $g_1(\mathbf{x_{i+e_1}}) = 0$. Similar to the discussion in two dimensions, we study the case $g_1(\mathbf{x_i}) = 0$ and the cases $g_1(\mathbf{x_{i\pm e_1}}) = 0$ separately.

Case $g_1(\mathbf{x_i}) = 0$. These exceptional points are classified into six categories based on the values of $\mathbf{G}(\mathbf{x_i})$, see Fig. 5. They can be handled by recipes 1 and 2 as above. We summarized their treatments in Table 1.

Case $g_1(\mathbf{x_{i\pm e_1}}) = 0$. The treatment for this case is identical to the same case in two dimensions. We thus skip the redundant explanation.



(a) $G(\mathbf{x_i}) = (0, 0, 0)$ (b) $G(\mathbf{x_i}) = (0, 0, 1)$ (c) $G(\mathbf{x_i}) = (0, 0, 2)$

(d) $G(\mathbf{x_i}) = (0, 1, 1)$ (e) $G(\mathbf{x_i}) = (0, 1, 2)$ (f) $G(\mathbf{x_i}) = (0, 2, 2)$

Figure 5: Exceptional points in three dimensions. The interface is colored by gradient of yellow. The circles and disks are grid points in Ω^- and Ω^+, respectively. Red, green, and blue segments are parallel to $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$, respectively. Note that the unit of each axes is the mesh size h.

	Recipe 1 (shift)	Recipe 2 (flip)
$\mathbf{G(x_i)} = (0, 0)$	$-$	$\surd$
$\mathbf{G(x_i)} = (0, 1)$	$\triangle$	$\surd$
$\mathbf{G(x_i)} = (0, 2)$	$\surd$	$-$
$\mathbf{G(x_i)} = (0, 0, 0)$	$-$	$\surd$
$\mathbf{G(x_i)} = (0, 0, 1)$	$\triangle$	$\surd$
$\mathbf{G(x_i)} = (0, 0, 2)$	$\surd$	$-$
$\mathbf{G(x_i)} = (0, 1, 1)$	$\surd$	$\triangle$
$\mathbf{G(x_i)} = (0, 1, 2)$	$\surd$	$-$
$\mathbf{G(x_i)} = (0, 2, 2)$	$\surd$	$-$

Table 1: Summary of the solutions for type-I exceptional points. The symbol $\surd$ means that it is our first solution for the case. The symbol $\triangle$ means that we use this solution when the first solution does not work. The symbol $-$ means that this solution does not work for the case.

2.5. Choosing recipe 1 or recipe 2?

The decision to choose recipe 1 or recipe 2 is based on the following rule. First, let $n(\mathbf{x_i}) = \sum_{k=1}^{d} g_k(\mathbf{x_i})$ be the number of adjacent points of $\mathbf{x_i}$ in the same region. If $n(\mathbf{x_i}) < 2$, then we apply recipe 2 unless the corresponding determinant is small than a given tolerant value. Otherwise, we apply recipe 1. The reasons are the follows. Recipe 2 cannot be applied for the cases $\mathbf{G}(\mathbf{x_i}) = (0, 2)$ in two dimensions and $\mathbf{G}(\mathbf{x_i}) = (0, 0, 2)$, $(0, 1, 2)$, and $(0, 2, 2)$ in three dimensions, because the exceptional point $\mathbf{x_i}$ remains an exceptional point after flipping the domain signature. For the case $\mathbf{G}(\mathbf{x_i}) = (0, 1, 1)$, we can choose either recipe 1 or 2. We choose recipe 1 for logical simplicity. All these cases are summarized by $n(\mathbf{x_i}) \geq 2$. On the other hand, if $n(\mathbf{x_i}) = 0$, i.e., there are no available adjacent points, recipe 2 is the only choice. If $n(\mathbf{x_i}) = 1$, both choices are possible. We choose recipe 2 in our code since the stencil of recipe 1 is usually larger.

3. Classification of Type-II exceptional points and corresponding recipes

We recall that a type-II exceptional point is a grid point $\mathbf{x_j}$ where the second-order cross derivative $\frac{\partial^2 u(\mathbf{x_j})}{\partial x_k \partial x_\ell}$ can not be approximated easily by neighboring grid values on the same region. By definition, $\mathbf{x_j}$ is not a type-I exceptional point, thus we have at least one adjacent grid point in the same region as $\mathbf{x_j}$ in each direction, i.e., $g_k(\mathbf{x_j}), g_\ell(\mathbf{x_j}) \geq 1$. Without loss of generality, we may assume that $\mathbf{x_{j-e_k}}$, $\mathbf{x_{j-e_\ell}}$, and $\mathbf{x_j}$ are in the same region but $\mathbf{x_{j-e_k-e_\ell}}$ is not. Other situations are simply a rotation and reflection of this one. Under this assumption, there are six cases in the $\mathbf{e}_k$-$\mathbf{e}_\ell$ plane, see Fig. 6.

Case 1. In case 1 (Fig. 6(a)), the approximation of cross derivative in the first case can be derived from the coupling technique in CIM2. Since $\mathbf{x_{j+e_k-e_\ell}}$ is in the same region of $\mathbf{x_j}$, we can use the following approximation for the cross derivative:

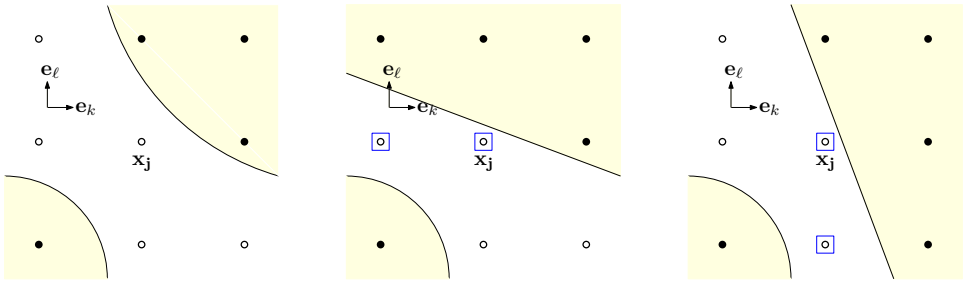
$$\frac{\partial^2 u(\mathbf{x_j})}{\partial x_k \partial x_\ell} = \frac{1}{h^2}(u_{\mathbf{j}} - u_{\mathbf{j-e_k}} - u_{\mathbf{j-e_\ell}} + u_{\mathbf{j+e_k-e_\ell}}) + \frac{\partial^2 u(\mathbf{x_j})}{\partial x_k^2} + O(h). \tag{21}$$

We then use (21) to replace (6) in the derivation of the principal derivatives. Note that $\frac{\partial^2 u(\mathbf{x_j})}{\partial x_k^2}$ is used in the approximation and then coupled in Eq. (7). Besides, since $\mathbf{x_{j-e_k+e_\ell}}$ is in the same region of $\mathbf{x_j}$, we can also use the following approximation for the cross derivative:

$$\frac{\partial^2 u(\mathbf{x_j})}{\partial x_k \partial x_\ell} = \frac{1}{h^2}(u_{\mathbf{j}} - u_{\mathbf{j-e_k}} - u_{\mathbf{j-e_\ell}} + u_{\mathbf{j-e_k+e_\ell}}) + \frac{\partial^2 u(\mathbf{x_j})}{\partial x_\ell^2} + O(h). \tag{22}$$

Note that $\frac{\partial^2 u(\mathbf{x_j})}{\partial x_\ell^2}$ is used in the approximation and then coupled in Eq. (7). Both replacements, (21) and (22), change the coupling matrix. Fortunately, the resulting coupling matrix is still solvable in all our tests.

Case 2. In cases 2-6, the design of finite difference approximations of cross derivatives depends on how we deal with the neighboring type-I exceptional points. For example, in case 2, at $\mathbf{x_{j-e_k}}$, we can use recipe 2 (flip) or recipe 1 (shift). If we use recipe 2, then we need to flip the domain signature of $\mathbf{x_j}$ to get enough grid points to compute the cross-derivative. If we choose recipe 1, then we can use (21) to approximate the cross derivative.



(a) case 1 (b) case 2 (c) case 3

(d) case 4 (e) case 5 (f) case 6

Figure 6: Exceptional points of type-II in two dimensions. The region colored by light yellow is Ω^+ while the white region is Ω^-. The circles and disks are grid points in Ω^- and Ω^+, respectively.

Case 3. If we flip the domain signature of $\mathbf{x}_{\mathbf{j}-\mathbf{e}_\ell}$, we will need to flip the domain signature of $\mathbf{x}_\mathbf{j}$. Alternatively, if choose recipe 1 at $\mathbf{x}_{\mathbf{j}-\mathbf{e}_\ell}$, then we can use (22) to approximate the cross derivative.

Cases 4-6. In these cases, we choose recipe 2. We flip the domain signatures at the those blue squares in Figs. 6(d), 6(e), and 6(f) so that the cross derivative at the ghost state can be obtained.

4. A practical issue – how fine of mesh we should choose

It is worth noticing that many cases in the above two sections can be avoided if the underlying mesh is fine enough. We call an exceptional point *non-generic* if it can be removed (no longer being an exceptional point) after mesh refinement. We claim that all type-II and some type-I exceptional points are non-generic provided the interface is a smooth closed curve. The characterization of type-I exceptional points are listed in Table 2. However, the case $\mathbf{G}(\mathbf{x_i}) = (0, 1)$ is generic. Let us consider a circle centered at $(0, 0)$ with radius $1/\sqrt{2}$. Suppose the domain is $[-1, 1] \times [-1, 1]$ and the mesh size is $h = 2/N$, where N is the number of intervals on $[-1, 1]$. Let $(kh, \ell h)$ be the grid point of the uniform grid, where k and ℓ are integers. The nearest grid point inside the circle on the x-axis is $(0, \lfloor 1/(\sqrt{2}h)\rfloor h)$. We can know whether $(\pm h, \lfloor 1/(\sqrt{2}h)\rfloor h)$ is inside the circle or not by checking the signs of the following function:

$$h^2 + \left(\left\lfloor \frac{1}{\sqrt{2}h} \right\rfloor h\right)^2 - \frac{1}{2},$$

or

$$\left(\frac{2}{N}\right)^2 + \left(\left\lfloor \frac{N}{2\sqrt{2}} \right\rfloor \frac{2}{N}\right)^2 - \frac{1}{2}.$$

It is easy to write a code to check the signs and the signs for most of N are negative. However, the signs are positive for some N, e.g., $N = 99$, 577, 3940, 665857, 45239074, and more. We show that in many cases of h, $(0, \lfloor 1/(\sqrt{2}h)\rfloor h)$ is inside the circle, while $(\pm h, \lfloor 1/(\sqrt{2}h)\rfloor h)$ is outside the circle even after mesh refinement. Thus, such kind of exceptional point cannot be removed after mesh refinement. That is why we call it generic.

In order to clarify how fine of a mesh we should choose to avoid complicated cases, we introduce two indicators. First, we define the *bottle-neck width* of the interface Γ by

$$\delta(\Gamma) := \inf_{\mathbf{x}\in\Gamma}\{r | \bar{B}_r(\mathbf{x}) \cap \Gamma \text{ contains at least two non-empty disjoint subsets}\}, \qquad (23)$$

if this set is non-empty. Here, $\bar{B}_r(\mathbf{x}) = \{\mathbf{y} | \|\mathbf{y} - \mathbf{x}\|_2 \le r\}$. If this set is empty, then we define $\delta(\Gamma)$ to be the diameter of Γ (the maximum distance between any two points on Γ). Second, we define the *maximum curvature* of Γ to be $\kappa_\infty = \sup_{\mathbf{x}\in\Gamma} \kappa_\Gamma(\mathbf{x})$, where $\kappa_\Gamma(\mathbf{x})$ is the (un-signed) curvature of Γ in 2D, and the largest principal curvature (in absolute value) of Γ in 3D at $\mathbf{x} \in \Gamma$.

We claim that: under the conditions (i) $h < \delta(\Gamma)/\sqrt{5}$ and (ii) $\kappa_\infty h < 4/5$, we can avoid the cases: (i) type-I exceptional points with $\mathbf{G}(\mathbf{x_i}) = (0, 0)$, $(0, 2)$ in two dimensions,

	Generic
$\mathbf{G}(\mathbf{x_i}) = (0,0)$	$-$
$\mathbf{G}(\mathbf{x_i}) = (0,1)$	$\surd$
$\mathbf{G}(\mathbf{x_i}) = (0,2)$	$-$
$\mathbf{G}(\mathbf{x_i}) = (0,0,0)$	$-$
$\mathbf{G}(\mathbf{x_i}) = (0,0,1)$	$\surd$
$\mathbf{G}(\mathbf{x_i}) = (0,0,2)$	$-$
$\mathbf{G}(\mathbf{x_i}) = (0,1,1)$	$\surd$
$\mathbf{G}(\mathbf{x_i}) = (0,1,2)$	$\surd$
$\mathbf{G}(\mathbf{x_i}) = (0,2,2)$	$-$

Table 2: The symbol $\surd/-$ means that it is a generic/non-generic case.

and (ii) all type-II exceptional points in two and three dimensions. In addition, we also conjecture that under the same conditions, we can avoid the type-I exceptional points with $\mathbf{G}(\mathbf{x_i}) = (0,0,0)$, $(0,0,2)$, $(0,2,2)$ in three dimensions. Some of the proofs of the above claim will be given in the Appendix.

5. Numerical Results and Discussions

Below we test our method in two and three dimensions. The domain is chosen to be $\Omega = [-1,1]^2$ or $[-1,1]^3$ in 2D and 3D, respectively. The underlying Cartesian grid is uniform with N intervals on each edge of the computational domain Ω, i.e., $h = 2/N$. Let u_e denote the exact solution of Eq. (1), and u stand for the numerical solution obtained by our method. For the numerical tests, the exact solution is chosen first. Then the source term, the boundary condition and the jump conditions are derived by plugging the exact solution into the equation. We focus on the accuracy for the numerical solution and its gradient. The solution error $\|u - u_e\|_\infty$ is the maximum error over all grid points. The gradient error $\|\nabla u - \nabla u_e\|_{\infty,\Gamma}$ is the maximum error of the gradient component $\nabla u - \nabla u_e$ over all intersections of interface and grid lines, where the gradient ∇u is computed from Eq. (5). All the experiments are tested on a machine with Intel i7-3930K CPU and 32G memory.

Below, we shall compare the following two methods.

- Our new method (CIM2 + new recipes), we shall called it the Improved Coupling Interface Method (ICIM);
- the Hybrid Coupling Interface Method (HCIM) [26]. This method adopts CIM2 at regular points, and CIM1 at exceptional points.

Example 1. We choose a flower-like interface which was introduced in [26]. To test the robustness of our new method (ICIM), we rotate this flower so that the relative locations of the interface points (intersections of the interface and the Cartesian meshes) in grid segments are quite arbitrary. The flower-like interface is described in polar coordinate by

$$r = a(1 + b\sin(\omega(\theta + t) + \psi)), \tag{24}$$

where $a > 0$, $0 < b < 1$, $\omega \in \mathbb{Z}$, ψ is a phase parameter, and t is a time parameter for rotating the flower. Note that the maximum curvature of this interface is

$$\kappa_\infty = \frac{|\omega b^2 + b - 1|}{a(1-b)^2}. \tag{25}$$

The elliptic coefficients and exact solution are given in prior as follows:

$$\varepsilon(x,y) = \begin{cases} 80 & \text{if } (x,y) \in \Omega^+ \\ 2 & \text{if } (x,y) \in \Omega^- \end{cases} \tag{26}$$

$$u_e(x,y) = \begin{cases} x^2 + y^2 + xy + x^3 & \text{if } (x,y) \in \Omega^+ \\ 4x^2 + 3y^2 + 5xy & \text{if } (x,y) \in \Omega^- \end{cases}. \tag{27}$$

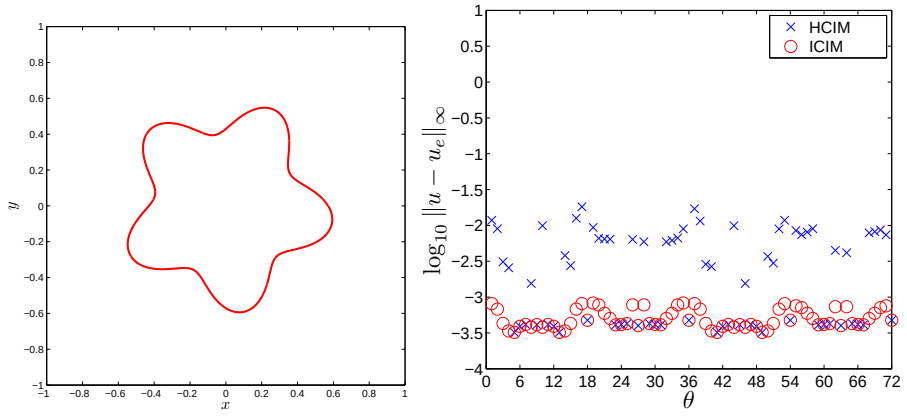
The prescribed jump conditions and the source term are computed from (1).

Case 1. We take $\omega = 5$, $a = b = 0.5$, $\psi = \pi/4$ for the first testing interface, see Fig. 7(a). We test robustness and accuracy here.

Robustness check. The robustness is tested by rotating the interface. The time t varies from 0 to $2\pi/5$ with $\Delta t = \pi/180$. The underlying Cartesian mesh is fixed. The bottleneck size $\delta(\Gamma) \approx 0.7529$, which is computed numerically. The maximum curvature $\kappa_\infty = 6$ is computed from (25). Therefore, we use

$$h = \frac{1}{20} = 0.05 < \min\{\frac{4}{5\kappa_\infty} \approx 0.13, \frac{1}{\sqrt{5}}\delta(\Gamma) \approx 0.3367\}.$$

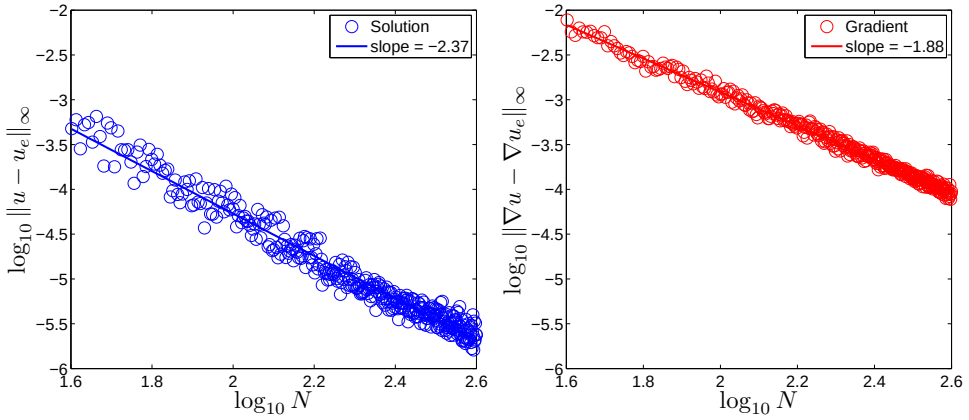
The error computed from HCIM fluctuates in time because CIM1 is used at exceptional points. However, the error computed from ICIM is quite uniform in time and much smaller, see Fig. 7(b).



(a) Interface (b) Robustness

Figure 7: Robustness test for the flower-like interface with $n = 5$, $a = b = 0.5$, and $\psi = \pi/4$. (a) the interface at $t = 0$. (b) The semi-log plot of the error versus the time t, $t \in [0, 2\pi/5]$. The unit of x-axis is degree. The cross and circle denote the errors computed from HCIM and ICIM, respectively.

Accuracy check. In this test, we fix $t = 0$ (i.e. the interface is steady). We refine the mesh with N (number of grid in each dimension) ranging from 40 to 400 with increment $\Delta N = 1$. The convergence result is shown in Fig. 8. We see that the fluctuation of errors computed by ICIM is smaller. In addition, the least squares fit of the errors show that the convergence of u is of second order and the accuracy of ∇u is slightly smaller than second order.



(a) Solution (b) Grad

Figure 8: The log-log plots of the error of solution (a) and its gradient (b) versus the number of intervals N with the interface in 7(a). The slopes of the lines in (a) and (b) are computed by least-square fit.

Case 2. In this test, we choose the same flower-like interface but with $\omega = 12$, $a = b = 0.5$, $\psi = \pi/4$, see Fig. 9(a). The bottle-neck size $\delta(\Gamma) \approx 0.0105$, and the $\kappa_\infty = 20$. Therefore, we choose

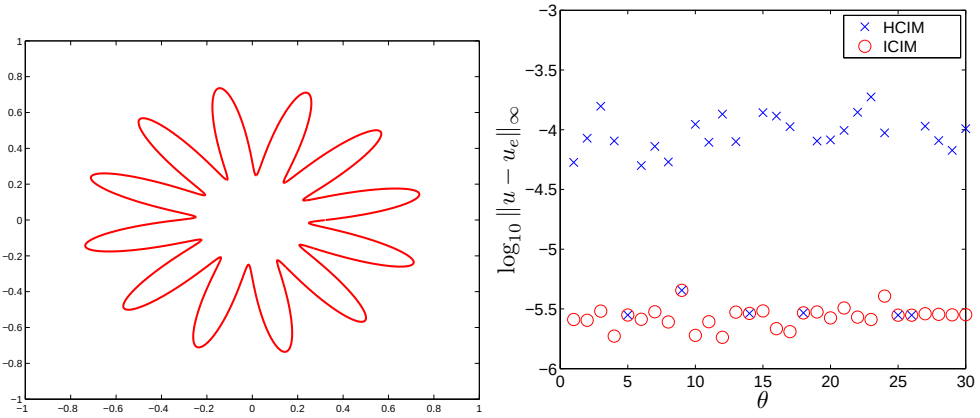
$$h = \frac{1}{320} = 0.0031 < \min\{\frac{4}{5\kappa_\infty} = 0.04, \frac{1}{\sqrt{5}}\delta(\Gamma) \approx 0.0047\}.$$

Robustness. We rotate the interface with time t varying from 0 to $\pi/6$ with $\Delta t = \pi/180$. We choose $h = \frac{1}{320}$. The error computed from HCIM fluctuates in time, whereas the error computed from ICIM is almost uniform in time and much smaller, see Fig. 9(b).

Accuracy check. In the accuracy test, we choose $h = 1/320$, $1/640$, and $1/1280$. We also choose $t = 7\pi/180$. Table 3 shows the second-order convergence in both u and its gradient ∇u.

mesh size	$\|u-u_e\|_\infty$	order	$\|\nabla u-\nabla u_e\|_{\infty,\Gamma}$	order
1/320	2.99×10^{-6}	–	1.63×10^{-4}	–
1/640	6.85×10^{-7}	2.13	3.26×10^{-5}	2.32
1/1280	1.78×10^{-7}	1.94	8.29×10^{-6}	1.97

Table 3: Convergence results for the interface in 9(a).



(a) Interface (b) Robustness check

Figure 9: Robustness test for the flower-like interface when $n = 12$, $a = b = 0.5$, and $\psi = \pi/4$. (a) the interface at $t = 0$. (b) The semi-log plot of the error versus the time t, $t \in [0, \pi/6]$. The unit of x-axis is degree. The cross and circle denote the errors computed from HCIM and ICIM, respectively.

Example 2. The interface in three dimensions can be very complex in practice, e.g., the surface of a molecular composed by hundreds or hundreds of atoms. In order to show the convergence clearly, we test several simple interfaces first. Here we use the three-dimensional interfaces in [26]: eight balls, ellipsoid, peanut, donut, banana, and popcorn. The level-set functions are listed below and the figures are plotted in Fig. 10:

- Eight balls: $\phi(x, y, z) = \min_{0\le k\le 7}\{\sqrt{(x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2}\} - 0.3$, where $(x_k, y_k, z_k) = ((-1)^{\lfloor k/4\rfloor} \times 0.5, (-1)^{\lfloor k/2\rfloor} \times 0.5, (-1)^k \times 0.5)$.
- Ellipsoid: $\phi(x, y, z) = 2x^2 + 3y^2 + 6z^2 - 1.3^2$.
- Peanut: $\phi(x, y, z) = \phi(r, \theta, \psi) = r - 0.5 - 0.2\sin(2\theta)\sin(\psi)$, where $x = r\cos\theta\sin\psi$, $y = r\sin\theta\sin\psi$, $z = r\cos\psi$.
- Donut: $\phi(x, y, z) = (\sqrt{x^2 + y^2} - 0.6)^2 + z^2 - 0.3^2$.
- Banana: $\phi(x, y, z) = (7x+6)^4+2401y^4+3601.5z^4+98(7x+6)^2(y^2+z^2)+4802y^2z^2-94(7x + 6)^2 + 3822y^2 - 4606z^2 + 1521$.
- Popcorn: $\phi(x, y, z) = \sqrt{x^2 + y^2 + z^2} - r_0 - \sum_{k=0}^{11} 2\exp(25((x - x_k)^2 + (y - y_k)^2 + (z - z_k)^2))$, where

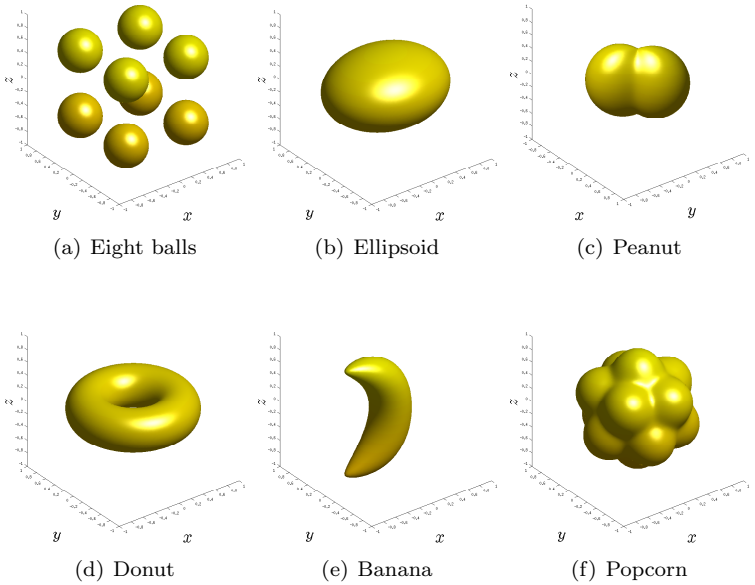
$$\begin{aligned}(x_k, y_k, z_k) &= \frac{r_0}{\sqrt{5}}\left(2\cos(\frac{2k\pi}{5} - \lfloor\frac{k}{5}\rfloor\pi), 2\sin(\frac{2k\pi}{5} - \lfloor\frac{k}{5}\rfloor\pi), (-1)^{\lfloor\frac{k}{5}\rfloor}\right), 0 \le k \le 9\\ &= r_0(0, 0, (-1)^{k-10}), 10 \le k \le 11.\end{aligned}$$

The elliptic coefficients and exact solution are given in prior:

$$\varepsilon(x,y,z) = \begin{cases} 80 & \text{if } (x,y,z) \in \Omega^+ \\ 2 & \text{if } (x,y,z) \in \Omega^- \end{cases} \tag{28}$$

$$u_e(x,y,z) = \begin{cases} xy + x^4 + y^4 + xz^2 + \cos(2x + y^2 + z^3) & \text{if } (x,y,z) \in \Omega^+ \\ x^3 + xy^2 + y^3 + z^4 + \sin(3(x^2 + y^2)) & \text{if } (x,y,z) \in \Omega^- \end{cases}. \tag{29}$$

The prescribed jump conditions and the source term are computed from (1). In the following tests, the number of intervals N ranges from 50 to 140 with the increment $\Delta N = 5$.



(a) Eight balls (b) Ellipsoid (c) Peanut

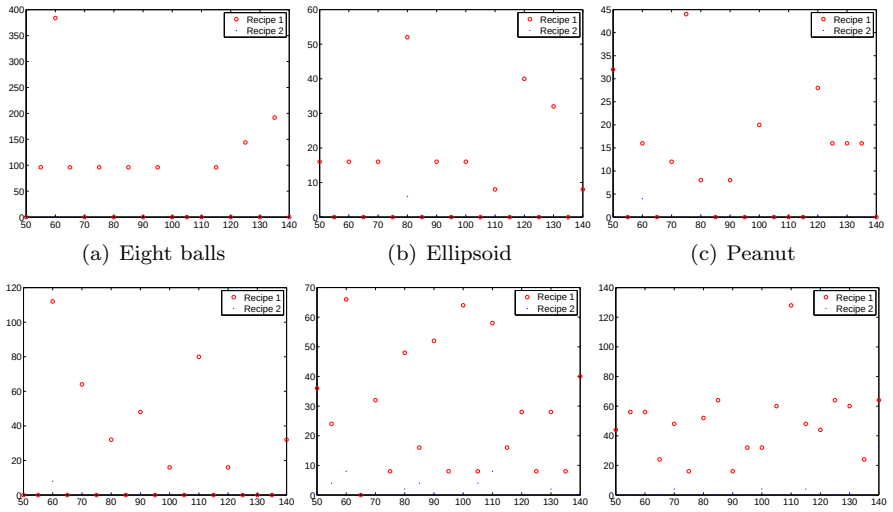
(d) Donut (e) Banana (f) Popcorn

Figure 10: The six interfaces: (a) eight balls; (b) ellipsoid; (c) peanut; (d) donut; (e) banana; (f) popcorn.

Figure 11 shows the number of grid points where recipe 1 and 2 are applied for the above six interfaces. By the rule for choosing recipe 1 or 2 illustrated in section 2.5, there are more exceptional points with recipe 1 than recipe 2.

Figure 12 shows the convergence results of the above six interfaces. The log-log plots of the error versus the number of intervals in one side shows the clearly second order convergence in maximum norm for the solutions. The convergence of the solution gradient is slightly below second order.

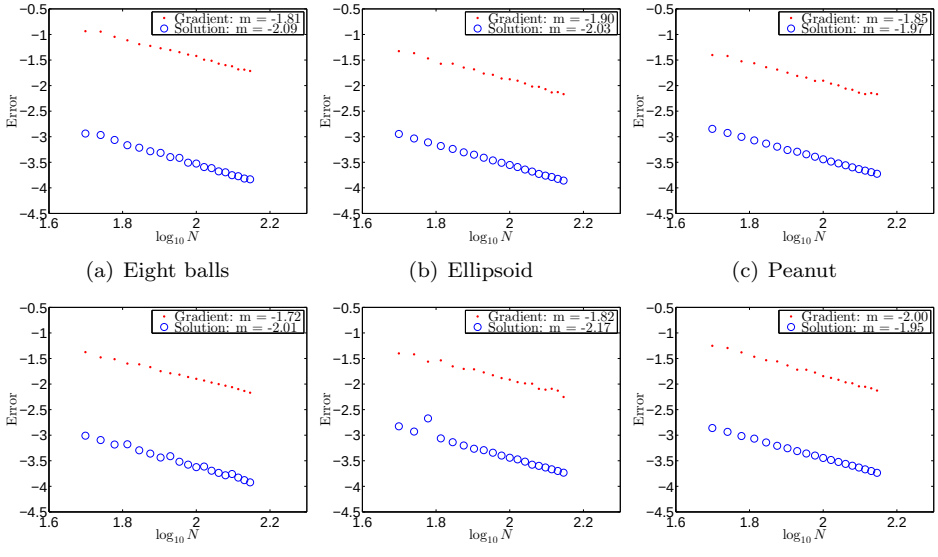
In order to show the difference between the original coupling interface method, we compare the errors computed by a first order method (CIM1), a hybrid order method (HCIM), and the improved coupling interface method (ICIM), see Fig. 13. The hybrid order method means that we apply CIM1 on those exceptional points. In Fig. 13(a), we see that the errors of the solution computed by ICIM is smaller than those of the other



(a) Eight balls (b) Ellipsoid (c) Peanut

(d) Donut (e) Banana (f) Popcorn

Figure 11: The number of grid points where recipe 1 and 2 are applied versus the number of intervals in one edge of domain Ω for the six interfaces: (a) eight balls; (b) ellipsoid; (c) peanut; (d) donut; (e) banana; (f) popcorn. In each figure, the number of intervals N ranges from 50 to 140 with the increment $\Delta N = 5$. Red circles and blue dots are the number which recipe 1 and 2 are applied, respectively.

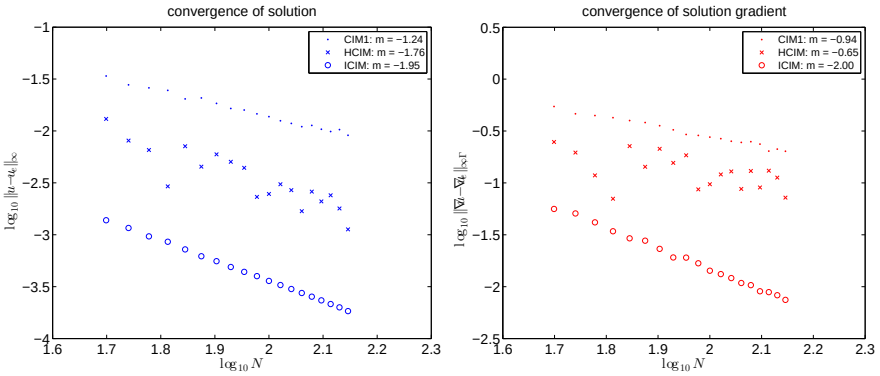


(a) Eight balls (b) Ellipsoid (c) Peanut

(d) Donut (e) Banana (f) Popcorn

Figure 12: The log-log plots of the error versus the number of intervals in one side for the six interfaces: (a) eight balls; (b) ellipsoid; (c) peanut; (d) donut; (e) banana; (f) popcorn. In each figure, the number of intervals N ranges from 50 to 140 with the increment $\Delta N = 5$. Red dots and blue circles are the maximum error of solution gradient $\|\nabla u - \nabla u_e\|_{\infty,\Gamma}$ and the maximum error of solution $\|u - u_e\|_\infty$, respectively.

two methods. However, in Fig. 13(b), it shows that the approximation for the solution gradient computed by CIM1 or HCIM is only of first order accuracy. In the meanwhile, the approximation for the solution gradient computed by ICIM is clearly of second order accuracy.



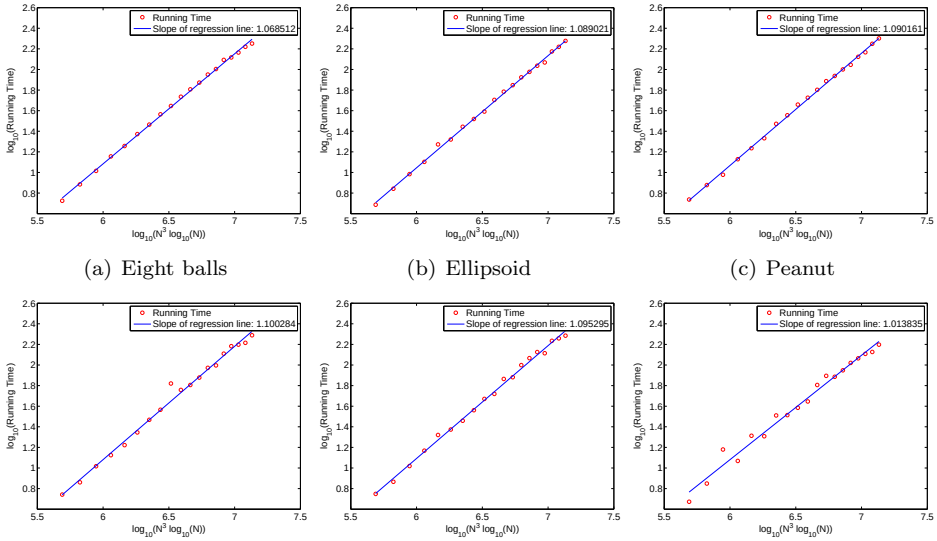
(a) Solution (b) Gradient

Figure 13: The log-log plots of the error versus the number of intervals in one side for the popcorn interface. (a) the comparison for the errors of solution; (b) the comparison for the errors of solution gradient. In each figure, the number of intervals N ranges from 50 to 140 with the increment $\Delta N = 5$. Blue dots, crosses, circles in (a) are the maximum errors of solution $\|u - u_e\|_\infty$ computed by CIM1, HCIM, and ICIM, respectively. Red dots, crosses, circles in (b) are the maximum error of solution gradient $\|\nabla u - \nabla u_e\|_{\infty,\Gamma}$ computed by CIM1, HCIM, and ICIM, respectively.

To study the efficiency issue, Figure 14 shows the log-log plot for the run time versus $N^3 \log_{10} N$. The slopes of the regression lines are almost 1. Since these are three dimensional runs, this result shows the run time is essential linearly proportionality to the number of unknowns. We also report the peak of memory usage from the system information. The peaks of memory usage are around 234MB and 1576MB if the total number of unknowns are 60^3 and 120^3, respectively. The memory usage is proportional to the total number of unknowns. In addition, we adopt the Gauss Seidel iteration as a smoother in the algebraic multigrid method for the six cases. If we change the smoother to the GMRES method, the computational time increases a little.

Example 3. Our third test interface is a molecular surface whose ID in the protein data bank (PDB-ID) is 1D63[33], see Fig. 15(a). From PDB, one can generate a PQR file by PDB2PQR[34] software with so-called AMBER force field. The PQR file contains the structure of the molecule which consists of the positions ($\mathbf{p}_i$), the partial charges (q_i), and the radius (r_i) of its atoms. The Van der Waal's surface of the molecule is the union of each sphere centered at $\mathbf{p}_i$ with radius r_i. However, there is a sharp corner at the intersection of two spheres. There are many methods to smooth the Van der Waal surfaces, such as the solvent excluded surface (SES). Since the solvent excluded surface is not represented by a level-set function, we construct another smoothed Van der Waal surface by using a smooth characterized function χ_η:

$$\chi_\eta(x) = \frac{1}{2}\left(1 + \tanh\left(\frac{x}{\eta}\right)\right). \tag{30}$$



(a) Eight balls (b) Ellipsoid (c) Peanut

(d) Donut (e) Banana (f) Popcorn

Figure 14: The log-log plots of running time versus the number of intervals in one edge of domain Ω for the six interfaces: (a) eight balls; (b) ellipsoid; (c) peanut; (d) donut; (e) banana; (f) popcorn. In each figure, the x-axis represents $\log_{10}(N^3 \log_{10}(N))$ when N ranges from 50 to 140 with the increment $\Delta N = 5$. Red circles are the running times. Blue lines are regression lines.

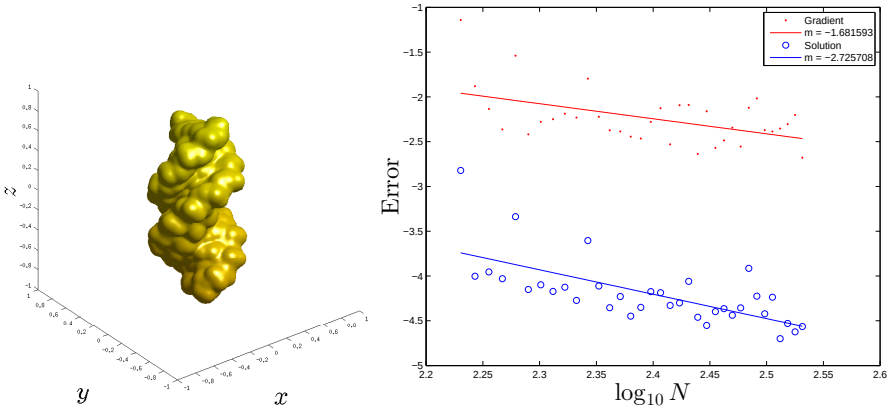
Then the level-set function for the smoothed Van der Waal surface is defined by

$$\phi(\mathbf{x}) = c - \sum_i \chi_\eta(r_i - \|\mathbf{x} - \mathbf{p}_i\|) \tag{31}$$

In the numerical tests, we choose $c = 0.25$ and $\eta = 1/40$. The smoothed molecular surface for 1D63 is shown in Fig. 15(a). It looks like structure rotating in the upward direction. Since it is a complex interface, the mesh size should be small enough to resolve the structure of the interface. The number of intervals (N) in one edge of Ω ranges from 170 to 340 with increment $\Delta N = 5$. The coefficients and the exact solution are the same as those in example 2. In this test, we adopt GMRES[36] as a smoother in the algebraic multigrid solver[37, 38, 39] to solve the corresponding system, which costs more time, but is relatively stable compared to the Gauss Seidel iterations. In Fig. 15(b), the errors fluctuate since the mesh size is not fine enough. However, from the least squares fit, we may see that the convergence rate is still of second order for the solution and slightly below second order (about 1.68) for the gradients. Besides, it takes 4480.11 seconds (around a hour and fifteen minutes) for computing the solution and the gradients when $N = 340$.

Example 4. Our final test is a diamond-like interface which has sharp corners and edges. The level-set function is given by

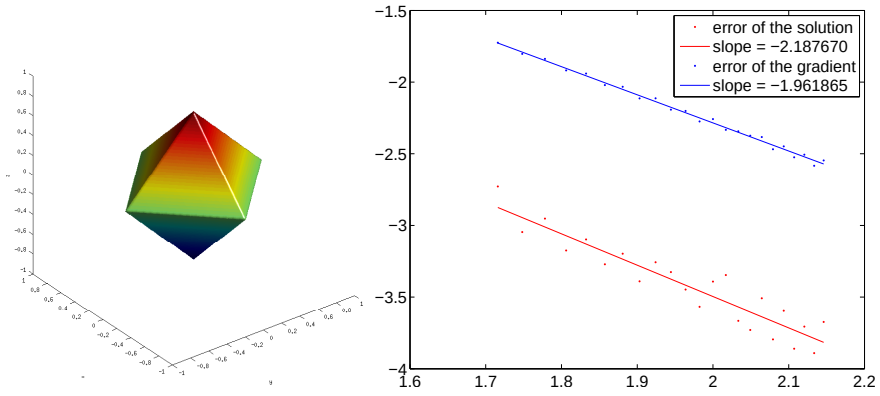
$$\phi(x, y, z) = |x| + |y| + |z| - c. \tag{32}$$



(a) Interface (b) Robustness

Figure 15: Convergence result for the 1D63 interface. (a) the smoothed Van der Waal surface of 1D63 defined by (31) with $c = 0.25$ and $\eta = 1/40$. (b) The log-log plot of the error versus the number of intervals in one side.

In the test, we choose $c = 0.7501$ in order to have six critical exceptional points along three coordinate directions, see Fig. 16(a). The coefficients and the exact solution are the same as those in example 2. For testing this situation, we keep N to be a multiple of 4, i.e., N ranges from 52 to 140 with increment $\Delta N = 4$. Figure 16(b) is the log-log plot of the error versus N. It shows the second order convergence for the solution and the gradient. Therefore, our recipes still work for cornered interface.



(a) Interface (b) Robustness

Figure 16: Convergence result for the diamond interface. (a) the diamond interface with $c = 0.7501$. (b) The log-log plot of the error versus the number of intervals in one edge.

6. Concluding Remarks

In this paper, we propose an improved coupling interface method for accurate approximation for the gradient in two and three dimensions of solutions of complex interface problems. We provide two recipes to improve accuracy at those exceptional points where there are not enough grid points for interpolation. Recipe 1 is to approximate second order derivatives at the exceptional points by a finite difference approximation at a nearby grid point based on a searching rule. Recipe 2 is to flip its domain signature and introduce a ghost state so that a second-order method can be applied. This ghost state is the smooth extension of the solution at the exceptional point from the other side of the interface. The original state is recovered by a post-processing using nearby states and jump conditions. The recipe choosing rule is based on a classification of the exceptional points. To decrease complexity of the classification and the number of exceptional points, a suitable mesh size in terms of maximum curvature and size of bottle-neck is also suggested. The improved method renders the solution and its gradient uniformly second-order accurate in the entire computed domain. In addition, the idea is also applicable to other interface solvers. We test our method in two and three dimensions with complex interfaces, including the surface of a real molecule (1D63) which is double-helix shape and composed hundreds of atoms. Numerical results show uniformly second-order accuracy for the solution and its gradients. The presently proposed recipes can be readily applied to eigenvalue problems with complex/sharp interfaces such as electromagnetic periodic structures (photonic crystals, plasmonic crystals, etc.) [35].

Acknowledgements

The work was supported in part by the National Science Council of the Republic of China under Contract No. NSC 100-2115-M-006-011-MY2 and NSC 102-2115-M-006-016.

Appendix A. Proofs of Propositions

In the following discussion, we assume that the interface Γ is a two-dimensional closed and smooth curve.

Lemma Appendix A.1. *Suppose* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are on the simple closed smooth curve* Γ *with the condition* $\overline{\mathbf{x}_1\mathbf{x}_2} < \delta(\Gamma)$. *The two points* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are connected in* $B_{\overline{\mathbf{x}_1\mathbf{x}_2}}(\mathbf{x}_1)$ *by a sub-curve* $\gamma \subset \Gamma$. *Suppose that* γ *is parametrized by* $\mathbf{x}(t)$ *with* $\mathbf{x}(0) = \mathbf{x}_1$ *and* $\mathbf{x}(1) = \mathbf{x}_2$. *The distance* $\overline{\mathbf{x}(t)\mathbf{x}_1}$ *is a non-decreasing function when* $t \in (0, 1)$.

Proof. By the definition of $\delta(\Gamma)$, we can show that $\mathbf{x}_1$ and $\mathbf{x}_2$ are connected by a sub-curve $\gamma \subset \Gamma$ in the ball centered at $\mathbf{x}_1$ with radius $\delta(\Gamma)$ easily. Suppose that γ is parametrized by $\mathbf{x}(t)$ with $\mathbf{x}(0) = \mathbf{x}_1$ and $\mathbf{x}(1) = \mathbf{x}_2$. If $\mathbf{x}(t)$ decreases at $t_0 \in (0, 1)$, there exists δ such that $\mathbf{x}(t_0 - \delta) > \mathbf{x}(t_0 + \delta)$. However, $\mathbf{x}(t_0 - \delta)$ is outside of the ball centered at $\mathbf{x}_1$ with radius $\overline{\mathbf{x}_1\mathbf{x}(t_0 + \delta)}$. It implies that $\mathbf{x}(t_0 + \delta)$ is not connected in the ball centered at $\mathbf{x}_1$ with radius $\overline{\mathbf{x}_1\mathbf{x}(t_0 + \delta)}$. By the definition of $\delta(\Gamma)$, $\overline{\mathbf{x}_1\mathbf{x}(t_0 + \delta)} > \delta(\Gamma)$. But $\overline{\mathbf{x}_1\mathbf{x}(t_0 + \delta)} < \delta(\Gamma)$. It deduces a contradiction. Therefore, the distance $\overline{\mathbf{x}(t)\mathbf{x}_1}$ is a non-decreasing function when $t \in (0, 1)$. □

Before we prove the propositions, we make a conjecture here:

Conjecture Appendix A.1. *Suppose that Γ is a simple closed smooth curve in two dimensions. Three different points $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ are selected sequentially on Γ. Let γ be the sub-curve of Γ which starts from $\mathbf{x}_1$, passes $\mathbf{x}_2$, and ends at $\mathbf{x}_3$. Then we have*

$$\max_{\mathbf{x}\in\gamma} \kappa_\Gamma(\mathbf{x}) \geq r_c^{-1}, \tag{A.1}$$

where r_c is the radius of the circumcircle of $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$.

This conjecture can estimate a lower bound of the maximum curvature κ_∞ in a piece of curve. We can say that the following propositions are true for the curve Γ on which the above conjecture is valid.

With this conjecture, we have the following two propositions.

Proposition Appendix A.1. *If $\kappa_\infty h < \frac{\sqrt{3}}{2}$ and $h < \frac{1}{2}\delta(\Gamma)$, then the case $\mathbf{G}(\mathbf{x_i}) = (0,2)$ does not occur.*

Proof. Suppose $\mathbf{G}(\mathbf{x_i}) = (0,2)$ does occur, this means that Γ cuts through the segments $[\mathbf{x_{i-e_1}}, \mathbf{x_i}]$ and $[\mathbf{x_i}, \mathbf{x_{i+e_1}}]$, see Figure 2(c) . Let $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ be the corresponding interface points, respectively. Since $\overline{\hat{\mathbf{x}}_1\hat{\mathbf{x}}_2} < 2h < \delta(\Gamma)$, $\hat{\mathbf{x}}_2$ is connected to $\hat{\mathbf{x}}_1$ in $\bar{B}_r(\hat{\mathbf{x}}_1)$ with $r = \overline{\hat{\mathbf{x}}_1\hat{\mathbf{x}}_2}$. However, there is no intersection in the segment from $\mathbf{x_{i-e_2}}$ to $\mathbf{x_{i+e_2}}$. In order to connect $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$, Γ should pass through the segment from $\mathbf{x_{i+e_2}}$ to $\mathbf{x_{i+2e_2}}$ or the segment from $\mathbf{x_{i-e_2}}$ to $\mathbf{x_{i-2e_2}}$. Without lose of generality, we assume that there is an intersection $\hat{\mathbf{x}}_3$ in the segment from $\mathbf{x_{i+e_2}}$ to $\mathbf{x_{i+2e_2}}$. We now search among all possibilities of the locations of $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$, and $\hat{\mathbf{x}}_3$. Let $\alpha_k = \overline{\hat{\mathbf{x}}_k\mathbf{x_i}}/h$, we have

$$0 \leq \alpha_1 \leq 1, \sqrt{(1+\alpha_1)^2} - \alpha_1 \leq \alpha_2 \leq 1, 1 \leq \alpha_3 \leq \sqrt{\alpha_2^2 + 2\alpha_1\alpha_2}. \tag{A.2}$$

Let R be the radius of the circumcircle which passes through $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$, and $\hat{\mathbf{x}}_3$.

$$R = \sqrt{\left(\frac{\alpha_2+\alpha_1}{2}\right)^2 + \left(\frac{1}{2\alpha_3}(\alpha_3^2 - \alpha_1\alpha_2)\right)^2} \tag{A.3}$$

We can easily check that (A.3) reaches its maximum $R = \frac{2\sqrt{3}h}{3}$ when $\alpha_1 = \alpha_2 = 1$ and $\alpha_3 = \sqrt{3}$. This implies $\kappa_\infty h \geq \frac{\sqrt{3}}{2}$ and violates the assumption. Therefore, $\mathbf{G}(\mathbf{x_i}) = (0,2)$ does not happen. □

Proposition Appendix A.2. *If $\kappa_\infty h < \frac{\sqrt{4+2\sqrt{2}}}{2}$ and $h < \frac{1}{2}\delta(\Gamma)$, $\mathbf{G}(\mathbf{x_i}) = (0,0)$ does not happen.*

Proof. If $\mathbf{G}(\mathbf{x_i}) = (0,0)$ happens, we can find at least four intersections of the interface, two of them are in the segment from $\mathbf{x_{i-e_1}}$ to $\mathbf{x_{i+e_1}}$, and the other two are in the segment from $\mathbf{x_{i-e_2}}$ to $\mathbf{x_{i+e_2}}$, see Figure 2 (a). Suppose these are $\hat{\mathbf{x}}_1 = \mathbf{x_i} - \alpha_1 h\mathbf{e}_1$, $\hat{\mathbf{x}}_2 = \mathbf{x_i} - \alpha_2 h\mathbf{e}_2$, $\hat{\mathbf{x}}_3 = \mathbf{x_i} + \alpha_3 h\mathbf{e}_1$, and $\hat{\mathbf{x}}_4 = \mathbf{x_i} + \alpha_4 h\mathbf{e}_2$, where $\alpha_k > 0$. These four intersections should be connected to each other since if the distance of each two points is smaller than $2h$ then it is also smaller than $\delta(\Gamma)$. Without loss of generality, we assume that Γ passes through

$\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$, $\hat{\mathbf{x}}_3$, and $\hat{\mathbf{x}}_4$ sequentially. And then it intersects the segment from $\mathbf{x}_{\mathbf{i}-\mathbf{e}_1+\mathbf{e}_2}$ to $\mathbf{x}_{\mathbf{i}+\mathbf{e}_2}$ at $\hat{\mathbf{x}}_5 = \mathbf{x}_{\mathbf{i}+\mathbf{e}_2} - \alpha_5 h$. Since the distance is smaller than $\delta(\Gamma)$, we have

$$\overline{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_1} \le \overline{\hat{\mathbf{x}}_3\hat{\mathbf{x}}_1} \le \overline{\hat{\mathbf{x}}_4\hat{\mathbf{x}}_1} \le \overline{\hat{\mathbf{x}}_5\hat{\mathbf{x}}_1} \tag{A.4}$$

$$\overline{\hat{\mathbf{x}}_3\hat{\mathbf{x}}_2} \le \overline{\hat{\mathbf{x}}_4\hat{\mathbf{x}}_2} \le \overline{\hat{\mathbf{x}}_5\hat{\mathbf{x}}_2} \tag{A.5}$$

$$\overline{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_3} \le \overline{\hat{\mathbf{x}}_1\hat{\mathbf{x}}_3} \tag{A.6}$$

$$\overline{\hat{\mathbf{x}}_4\hat{\mathbf{x}}_3} \le \overline{\hat{\mathbf{x}}_5\hat{\mathbf{x}}_3} \tag{A.7}$$

$$\overline{\hat{\mathbf{x}}_3\hat{\mathbf{x}}_4} \le \overline{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_4} \le \overline{\hat{\mathbf{x}}_5\hat{\mathbf{x}}_4} \tag{A.8}$$

$$\overline{\hat{\mathbf{x}}_4\hat{\mathbf{x}}_5} \le \overline{\hat{\mathbf{x}}_5\hat{\mathbf{x}}_3} \le \overline{\hat{\mathbf{x}}_2\hat{\mathbf{x}}_5} \le \overline{\hat{\mathbf{x}}_1\hat{\mathbf{x}}_5} \tag{A.9}$$

Therefore, we have the following inequalities:

$$0 \le \alpha_1 \le 1 \tag{A.10}$$

$$(\sqrt{2}-1)\alpha_1 \le \alpha_3 \le \sqrt{1+\alpha_1^2} - \alpha_1 \tag{A.11}$$

$$0 \le \alpha_5 \le \alpha_1 - \sqrt{(\alpha_1+\alpha_3)^2 - 1} \tag{A.12}$$

Let R be the radius of the circumcircle which passes through $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_3$, and $\hat{\mathbf{x}}_5$. We have

$$R = \sqrt{\left(\frac{\alpha_3+\alpha_1}{2}\right)^2 + \left(\frac{1-(\alpha_1-\alpha_5)(\alpha_3+\alpha_5)}{2}\right)^2} \tag{A.13}$$

We can easily check that (A.13) reaches its maximum $R = \sqrt{2-\sqrt{2}}h$ when $\alpha_1 = 1$, $\alpha_3 = \sqrt{2}-1$ and $\alpha_5 = 0$. This implies $\kappa_\infty h \ge \frac{\sqrt{4+2\sqrt{2}}}{2}$ and violates the assumption. Therefore, $\mathbf{G}(\mathbf{x}_\mathbf{i}) = (0,0)$ does not happen. □

For type-II exceptional points, we have the following proposition

Proposition Appendix A.3. *If $\kappa_\infty h < \frac{4}{5}$ and $h < \frac{1}{\sqrt{5}}\delta(\Gamma)$, type-II exceptional points does not happen.*

Proof. We will discuss all the cases in Fig. 6 For case 5 in the Fig. 6(e), we can find two intersections $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$, and $\hat{\mathbf{x}}_3$ in the segments $\overline{\mathbf{x}_{\mathbf{i}-\mathbf{e}_k}\mathbf{x}_{\mathbf{i}-\mathbf{e}_k-\mathbf{e}_\ell}}$, $\overline{\mathbf{x}_{\mathbf{i}-\mathbf{e}_k}\mathbf{x}_{\mathbf{i}-\mathbf{e}_k+\mathbf{e}_\ell}}$, and $\overline{\mathbf{x}_\mathbf{i}\mathbf{x}_{\mathbf{i}+\mathbf{e}_k}}$, respectively. Since $h \le \frac{1}{\sqrt{5}}\delta(\Gamma)$, $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$, and $\hat{\mathbf{x}}_3$ are connected. For these three points, the maximum radius of the circumcircle which passes through $\hat{\mathbf{x}}_1$, $\hat{\mathbf{x}}_2$, and $\hat{\mathbf{x}}_3$ is $\frac{5}{4}$ when $\hat{\mathbf{x}}_1 = \mathbf{x}_{\mathbf{i}-\mathbf{e}_k-\mathbf{e}_\ell}$, $\hat{\mathbf{x}}_2 = \mathbf{x}_{\mathbf{i}-\mathbf{e}_k+\mathbf{e}_\ell}$, and $\hat{\mathbf{x}}_3 = \mathbf{x}_{\mathbf{i}+\mathbf{e}_k}$. This gives us $\kappa_\infty h \le \frac{4}{5}$.

For the other cases, we can find at least $\hat{\mathbf{x}}_4$, $\hat{\mathbf{x}}_5$, and $\hat{\mathbf{x}}_6$ in the segments $\overline{\mathbf{x}_\mathbf{i}\mathbf{x}_{\mathbf{i}+\mathbf{e}_k}}$, $\overline{\mathbf{x}_\mathbf{i}\mathbf{x}_{\mathbf{i}+\mathbf{e}_\ell}}$, and $\overline{\mathbf{x}_{\mathbf{i}-\mathbf{e}_k-\mathbf{e}_\ell}\mathbf{x}_{\mathbf{i}-\mathbf{e}_k}}$, respectively. Since $h \le \frac{1}{\sqrt{5}}\delta(\Gamma)$, $\hat{\mathbf{x}}_4$, $\hat{\mathbf{x}}_5$, and $\hat{\mathbf{x}}_6$ are connected. For these three points, the maximum radius of the circumcircle which passes through $\hat{\mathbf{x}}_4$, $\hat{\mathbf{x}}_5$, and $\hat{\mathbf{x}}_6$ is $\frac{5\sqrt{2}}{6}$ when $\hat{\mathbf{x}}_4 = \mathbf{x}_{\mathbf{i}+\mathbf{e}_k}$, $\hat{\mathbf{x}}_5 = \mathbf{x}_{\mathbf{i}+\mathbf{e}_\ell}$, and $\hat{\mathbf{x}}_6 = \mathbf{x}_{\mathbf{i}-\mathbf{e}_k-\mathbf{e}_\ell}$. This gives us $\kappa_\infty h \le \frac{3\sqrt{2}}{5}$.

Since $\frac{4}{5} < \frac{3\sqrt{2}}{5}$, we conclude that if $\kappa_\infty h < \frac{4}{5}$ and $h < \frac{1}{\sqrt{5}}\delta(\Gamma)$, type-II exceptional points does not happen. □

[1] C. Peskin, The immersed boundary method, Acta Numerica 11 (2002) 479–517.

[2] Z. Li, K. Ito, The Immersed Interface Method, Numerical Solutions of PDEs Involving Interfaces and Irregular Domains, SIAM Frontiers in Applied Mathematics, 2006.
[3] J. T. Beale, A grid-based boundary integral method for elliptic problems in three dimensions, SIAM Journal on Numerical Analysis 42 (2004) 599 ??620.
[4] Z. Li, W. Wang, I. Chern, M. Lai, New formulations for interface problems in polar coordinates, SIAM Journal on Scientific Computing 25 (2003) 224–245.
[5] W. Wang, A jump condition capturing finite difference scheme for elliptic interface problems, SIAM Journal on Scientific Computing 25 (2004) 1479–1496.
[6] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numerische Mathematik 79 (1998) 175–202.
[7] J. Glimm, O. Mcbryan, A computational model for interfaces, Advances in Applied Mathematics 6 (1985) 422–435.
[8] J. Huang, J. Zou, A mortar element method for elliptic problems with discontinuous coefficients, IMA Journal of Numerical Analysis 22 (2002) 549–576.
[9] Z. Li, T. Lin, X. Wu, New Cartesian grid methods for interface problems using the finite element formulation, Numerische Mathematik 96 (2003) 61–98.
[10] H. Johansen, P. Colella, A cartesian grid embedded boundary method for Poisson's equation on irregular domains, Journal of Computational Physics 147 (1998) 60 ??85.
[11] A. Tornberg, B. Engquist, Regularization techniques for numerical approximation of pdes with singularities, Journal of Scientific Computing 19 (2003) 527–552.
[12] A. Tornberg, B. Engquist, Numerical approximations of singular source terms in differential equations, Journal of Computational Physics 200 (2004) 462–488.
[13] C. Peskin, Numerical-analysis of blood-flow in heart, J. Comput. Phys. 25 (1977) 220–252.
[14] R. Leveque, Z. LI, The immersed interface method for elliptic-equations with discontinuous coefficients and singular sources, SIAM Journal on Numerical Analysis 31 (1994) 1019–1044.
[15] Z. Li, K. Ito, Maximum principle preserving schemes for interface problems with discontinuous coefficients, SIAM Journal on Scientific Computing 23 (2001) 339–361.
[16] T. Chen, J. Strain, Piecewise-polynomial discretization and Krylov-accelerated multigrid for elliptic interface problems, Journal of Computational Physics 227 (2008) 7503–7542.
[17] J. Bedrossian, J. H. von Brecht, S. Zhu, E. Sifakis, J. M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains, Journal of Computational Physics 229 (18) (2010) 6405 – 6426.
[18] J. L. H. Jr., L. Wang, E. Sifakis, J. M. Teran, A second order virtual node method for elliptic problems with interfaces and irregular domains in three dimensions, Journal of Computational Physics 231 (4) (2012) 2015 – 2048.
[19] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM Journal on Numerical Analysis 21 (1984) 285 ??299.
[20] A. Mayo, High order accurate particular solutions of the biharmonic equation on general regions, in: Fast algorithms for structured matrices: theory and applications (South Hadley, MA, 2001), Vol. 323 of Contemp. Math., Amer. Math. Soc., Providence, RI, 2003, pp. 233–243.
[21] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), Journal of Computational Physics 152 (1999) 457–492.
[22] A. Wiegmann, K. Bube, The explicit-jump immersed interface method: Finite difference methods for PDEs with piecewise smooth solutions, SIAM Journal on Numerical Analysis 37 (2000) 827–862.
[23] P. Berthelsen, A decomposed immersed interface method for variable coefficient elliptic equations with non-smooth and discontinuous solutions, Journal of Computational Physics 197 (2004) 364–386.
[24] Y. Zhou, S. Zhao, M. Feig, G. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, Journal of Computational Physics 213 (1) (2006) 1 – 30.
[25] S. Yu, G. Wei, Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities, Journal of Computational Physics 227 (1) (2007) 602 – 632.
[26] I. L. Chern, Y. C. Shu, A coupling interface method for elliptic interface problems, J. Comput. Phys. 225 (2007) 2138–2174.
[27] X.-D. Liu, T. C. Sideris, Convergence of the ghost fluid method for elliptic equations with interfaces, Mathematics of Computation 72 (2003) 1731 ?? 1746.
[28] J. T. Beale, A. T. Layton, On the accuracy of finite difference methods for elliptic problems with interfaces, Communications in Applied Mathematics and Computational Science 1 (2006) 91 ??119.
[29] P. Macklin, J. Lowengrub, Evolving interfaces via gradients of geometry-dependent interior Poisson

problems: application to tumor growth, Journal of Computational Physics 203 (2005) 191 – 220.
[30] B. Li, Minimization of electrostatic free energy and the Poisson-Boltzmann equation for molecular solvation with implicit solvent, SIAM Journal on Mathematical Analysis 40 (6) (2009) 2536 – 2566.
[31] L.-T. Cheng, B. Li, M. White, S. Zhou, Motion of a cylindrical dielectric boundary, SIAM Journal on Applied Mathematics 73 (1) (2013) 594 – 616.
[32] B. Li, Y. Zhao, Variational implicit solvation with solute molecular mechanics: From diffuse-interface to sharp-interface models, SIAM Journal on Applied Mathematics 73 (1) (2013) 1 – 23.
[33] D. Brown, M. Sanderson, E. Garman, S. Neidle, Crystal structure of a berenil-d(cgcaaatttgcg) complex. an example of drug-dna recognition based on sequence-dependent structural features, Journal of Molecular Biology 226 (1992) 481 – 490.
[34] D. TJ, N. JE, M. JA, B. NA., PDB2PQR: an automated pipeline for the setup, execution, and analysis of Poisson-Boltzmann electrostatics calculations, Nucleic Acids Research 32 (2004) W665 – W667.
[35] Y. C. Shu, C. Y. Kao, I. L. Chern, C. C. Chang, Augmented coupling interface method for solving eigenvalue problems with sign-changed coefficients, J. Comput. Phys. 229 (2010) 9246-9268
[36] Y. Saad, M.H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. (1986) 856-869
[37] A. Brandt, S. McCormick, J. Ruge, Algebraic multigrid (AMG) for sparse matrix equations, Sparsity and Its Applications (D.J. Evans, ed.), Cambridge Univ. Press (1984).
[38] A. Brandt, Algebraic multigrid theory: The symmetric case, Appl. Math. Comput. (1986), 23-56
[39] J. W. Ruge, K. Stüben, Algebraic multigrid (AMG), in: S.F. McCormick (Ed.), Multigrid Methods, Frontiers in & Applied Mathematics, Vol. 5, SIAM, Philadelphia (1986)