

## Highlights

### **A three-dimensional high-order geometry-aligned immersed finite element method for elliptic interface problems**

Chang Lu, Tao Lin

- A three-dimensional high-order geometry-aligned immersed finite element method for elliptic interface problems.
- Direct geometric evaluation for parameterized surfaces and local Legendre–Monge reconstruction for general interfaces.
- A unified polynomial framework for geometry representation and IFE basis-function construction.
- Algebraic evaluation of higher-order Laplacian normal derivatives without explicit symbolic expansions.
- Saye-type recursive quadrature and optimal  $L^2$  and  $H^1$  convergence rates for curved-interface problems.

# A three-dimensional high-order geometry-aligned immersed finite element method for elliptic interface problems

Chang Lu<sup>a,\*</sup>, Tao Lin<sup>b</sup>

<sup>a</sup>State Key Laboratory of Structural Analysis, Optimization and CAE Software for Industrial Equipment, Dalian University of Technology, , Dalian, 116024, , China

<sup>b</sup>Department of Mathematics, Virginia Tech, 225 Stanger St., 460 McBryde Hall, MC 0123, Blacksburg, 24061, VA, USA

## ARTICLE INFO

**Keywords:**

Immersed finite element  
High-order finite element  
Geometry-aligned approximation  
Elliptic interface problems  
Legendre–Monge reconstruction  
Saye-type quadrature

## ABSTRACT

This paper presents a three-dimensional high-order geometry-aligned immersed finite element (IFE) method for elliptic interface problems with curved interfaces. The framework supports two alternative geometric pathways: direct evaluation from a sufficiently regular global surface parameterization, and high-order element-wise Legendre–Monge reconstruction from surface samples and, when available, consistently oriented normal data. In the latter case, the local height function is approximated by scaled Legendre polynomials and converted into a signed implicit representation. Since the IFE basis functions are constructed within the same Legendre polynomial space, the geometric description and physical finite-element approximation share a unified mathematical foundation, which is the extension of the classical isoparametric concept. This consistent local reconstruction and its implicit representation mitigate error coupling between geometric approximation and finite-element discretization, while facilitating both IFE basis-function construction and numerical integration. A purely algebraic procedure is developed to evaluate higher-order normal derivatives of the Laplacian using the chain rule and metric tensor variations, without explicit symbolic expansions. Numerical integration on interface elements is realized using the Saye-type dimension-reduction scheme. Only one-dimensional root-finding and one-dimensional Gaussian quadrature are required, which greatly lowers the integration complexity. Numerical experiments confirm optimal convergence rates of the present method.

## 1. Introduction

In this paper, we introduce a new high-order geometry-aligned immersed finite element (GA-IFE) approach for the second-order elliptic interface problem in three dimensions:

$$\begin{cases} -\nabla \cdot (\beta \nabla u) = f, & \text{on } \Omega^+ \cup \Omega^-, \\ u|_{\partial\Omega} = g, \end{cases} \quad (1)$$

where, without loss of generality, the domain  $\Omega \subset \mathbb{R}^3$  is assumed to be a tensor-product domain divided by an interface  $\Gamma$  into two subdomains  $\Omega^+$  and  $\Omega^-$ , and the diffusion coefficient  $\beta$  is piecewise constant:  $\beta|_{\Omega^\pm} = \beta^\pm > 0$ . In addition to Eq. (1), the solution  $u$  is assumed to satisfy the following jump conditions at the interface  $\Gamma$ :

$$[[u]]_\Gamma = 0, \quad (2)$$

$$[[\beta \frac{\partial u}{\partial \mathbf{n}}]]_\Gamma = 0, \quad (3)$$

where  $[[v]]_\Gamma = v^+|_\Gamma - v^-|_\Gamma$  with  $v^\pm = v|_{\Omega^\pm}$  denotes the jump of  $v$  across the interface  $\Gamma$ , and  $\mathbf{n}$  is the unit normal vector of  $\Gamma$ . Also, following [3–5], for constructing IFE functions with polynomials of degree  $m \geq 2$ , we assume the source term  $f$  is sufficiently smooth around the interface such that

$$[[\beta \frac{\partial^j u}{\partial \mathbf{n}^j}]]_\Gamma = 0, \quad j = 0, 1, \dots, m-2. \quad (4)$$

\*Corresponding author

✉ changlu@dlut.edu.cn (C. Lu); tlin@vt.edu (T. Lin)

ORCID(s):

The above elliptic interface problem arises in plasma simulations [31, 36, 37, 47], topology optimization [23], multiphase flows [14, 18], and many other applications where it is preferable to use a solver designed on non-body-fitted meshes. These problems typically involve discontinuous coefficients, or prescribed jumps across the interface, making accurate treatment on non-body-fitted meshes a critical challenge [42, 45]. Furthermore, high-precision numerical solutions are increasingly essential for such problems. For instance, phenomena including anomalous wall erosion [13, 61] and high-frequency energy coupling [19] in electrostatic Particle-in-Cell (PIC) simulations heavily rely on the accuracy of field solvers and interface capturing; simulations with low-order accuracy may even fail to resolve such physical behaviors. In addition, memory capacity and bandwidth act as the dominant bottlenecks limiting simulation performance under modern supercomputing architectures. High-order algorithms can exploit the computational throughput of hardware, thereby delivering distinct performance advantages.

Several non-body-fitted-mesh approaches have been developed for these problems, like cut finite element methods (CutFEM) [12, 20–22, 55, 56], immersed interface methods (IIM) [42], extended finite element methods (XFEM) [11, 32, 52], and immersed finite element methods (IFE) [33, 44, 45], to name a few. Within this family, the IFE methods adopt standard polynomial basis functions for non-interface elements, while specially constructed piecewise-polynomial IFE basis functions replace them on interface elements to enforce the jump conditions [44, 46]. The piecewise IFE basis function resembles the well-known Hsieh–Clough–Tocher macro  $C^1$  element [16]. However, achieving high-order accuracy with traditional IFE methods is difficult because the interface within each element is typically approximated by a line segment in two-dimensional (2D) or a planar patch in three-dimensional (3D), limiting convergence and complicating basis construction [2–4, 24].

A recent 2D geometry-conforming IFE (GC-IFE) method [5–7, 9, 51] overcomes this by introducing a Frenet transformation [1] that maps the curved interface to a straight line in local coordinates, enabling the construction of an IFE basis with a universal  $Q_m$ -type polynomial basis. This approach achieves higher-order, exact enforcement of jump conditions, well-conditioned local systems, and Sobolev-conforming IFE spaces with optimal approximation constants independent of interface position. High-order 2D PIC simulations, topology optimization and other algorithms can be then conveniently implemented within the framework of the 2D GC-IFE method. Also, we are working on coupling the GC-IFE method with the PIC method to address challenging physical phenomena, such as anomalous plasma erosion and high-frequency energy coupling in plasma thrusters. Nevertheless, it turns out that many critical physical process are intrinsically 3D and cannot be simplified, like the electron drift instability and the discharge uniformity [30, 43, 47, 60]. In addition, many physical or engineering applications such as electrostatic levitation of dust particles and spacecraft surface charging also heavily relies on 3D simulations [17, 26–29, 29, 48, 49, 58, 59]. For these real-world applications, the programmability and engineering deployability of 3D simulations are also critical. These considerations naturally motivate the next key task: extending the 2D GC-IFE method to three dimensions while preserving ease of implementation.

However, this extension is far from straightforward. First, a general surface in three dimensions requires two local tangential coordinates, and a single global parameterization may be unavailable for complex geometries because of patch transitions, coordinate singularities, and topological obstructions [1]. Second, assembling the interface condition matrix is much more involved because expanding the Laplacian in the local surface coordinates, along with its normal derivatives, yields complicated formulas. Third, high-order numerical integration on 3D interface elements is substantially more complex, as cutting a hexahedron by a curved surface generates numerous subelement configurations with curved faces and edges [22, 40, 41, 54].

In this paper, a 3D high-order GA-IFE framework is developed with two geometric representations for interfaces to address the three challenges mentioned above. When a sufficiently regular global parameterization is available, interface geometric quantities, including surface coordinates and geometric derivatives, are directly evaluated from this analytical parameterization. For general interfaces lacking such parameterization, discrete surface samples are employed to perform element-wise geometric reconstruction using a signed local Monge patch, in which the interface height function is approximated within a scaled Legendre polynomial space, and consistently oriented normals, if available, can be used to enhance the fitting. High-order geometric derivatives can thus be obtained from either analytical parametric surfaces or reconstructed approximate surfaces. The unified Legendre–Monge representation supports IFE basis-function construction and numerical quadrature for a broad range of interface types. In both geometric paradigms, the resultant local geometric quantities serve as consistent inputs for the same IFE construction and quadrature infrastructure. By adopting Legendre polynomial spaces for both geometric reconstruction and field

approximation, the proposed method establishes a unified high-order algebraic framework. It extends the geometry-aligned spirit of the classical isoparametric concept [15] and it shares an essential idea with the isogeometry analysis (IGA) method [10, 35] in the sense of using the exact geometry in finite element computations.

Building upon the aforementioned unified geometric foundation, this work further addresses two critical challenges in 3D IFE discretization: the cumbersome expansion of higher-order differential operators and the intricate numerical integration over interface elements. The overall accuracy of the method is governed by the polynomial approximation order, the quality of the local geometric representation, the resolution of the interface curvature, and the accuracy of the numerical integration. Accordingly, the proposed framework consists of three core components that jointly address these challenges and the underlying geometric approximation issues, as summarized below.

- **Geometric representation:** The framework supports two alternative representations of the interface. When a sufficiently regular global parameterization  $\mathbf{S}(u, v)$  is available, the surface geometry and its derivatives are evaluated directly from the parameterization. For general interfaces, a local signed Legendre–Monge representation is reconstructed element-wise from surface samples and, when available, consistently oriented normal data. The two representations provide the geometric information required by the subsequent IFE construction and can be selected according to the available interface description.
- **Interface conditions and their algebraic treatment:** The 3D interface conditions are assembled on the local surface patch. Their higher-order normal derivatives are evaluated by an algebraic procedure that avoids explicit symbolic expansion of the full curvilinear Laplacian. This procedure is also applicable to 2D settings.
- **High-order quadrature:** On interface elements, we employ the dimension-reduction Saye-type [54] quadrature strategy to integrate over the implicitly defined phase subdomains. This provides a unified high-order integration procedure for the 3D cut-cell geometry.

The remainder of this paper is organized as follows. Section 2 revisits the 2D GC–IFE method and elaborates its extension to the 3D GA–IFE framework. Section 3 details the geometric modeling techniques, including direct parametric surface evaluation and high-order Legendre–Monge reconstruction, along with the computation of essential geometric quantities. Section 4 presents the construction of 3D GA–IFE basis functions and the assembly procedure of the interface condition matrix. Section 5 introduces the full matrix-vector assembly workflow and the tailored high-order quadrature strategy for the discontinuous Galerkin (DG) formulation. Section 6 presents numerical experiments for both parametric and reconstructed curved interfaces, validating the optimal accuracy and convergence performance of the proposed method.

## 2. The GC–IFE Method

This section briefly reviews the 2D GC–IFE method developed in [5–7, 9, 51]. We then present the corresponding 3D formulations and highlight the key issues when extending the method from 2D to 3D.

### 2.1. Review of the 2D GC–IFE Method

Let  $\Omega \subset \mathbb{R}^2$  be a domain partitioned by a smooth interface  $\Gamma$  into two subdomains  $\Omega^+$  and  $\Omega^-$ . The diffusion coefficient  $\beta(\mathbf{x})$  is piecewise constant, i.e.,  $\beta(\mathbf{x}) = \beta^\pm$  for  $\mathbf{x} \in \Omega^\pm$ . Let  $\mathcal{T}_h$  be an interface-independent Cartesian mesh and  $\mathcal{E}_h$  be the set of edges of the solution domain  $\Omega$ . Then, in the 2D GC–IFE method, the Symmetric Interior Penalty Galerkin (SIPG) formulation is used for the elliptic interface problem to seek  $u_h \in \mathcal{H}^s(\mathcal{T}_h, \Gamma; \beta)$  such that:

$$a_h(u_h, v_h) = L_h(v_h), \quad \forall v_h \in \mathcal{H}^s(\mathcal{T}_h, \Gamma; \beta), \quad (5)$$

where  $\mathcal{H}^s(\mathcal{T}_h, \Gamma; \beta)$  is the immersed Sobolev space[5], the bilinear form  $a_h(u_h, v_h)$  is defined as:

$$\begin{aligned} a_h(u_h, v_h) = & \sum_{K \in \mathcal{T}_h} \int_K \beta \nabla u_h \cdot \nabla v_h \, d\mathbf{x} \\ & - \sum_{e \in \mathcal{E}_h} \int_e \left( \langle \beta \nabla u_h \cdot \mathbf{n}_e \rangle \llbracket v_h \rrbracket + \langle \beta \nabla v_h \cdot \mathbf{n}_e \rangle \llbracket u_h \rrbracket - \frac{\sigma_e}{h_e} \llbracket u_h \rrbracket \llbracket v_h \rrbracket \right) ds, \end{aligned} \quad (6)$$

and  $L_h(v_h)$  is defined as:

$$L_h(v_h) = \sum_{K \in \mathcal{T}_h} \int_K f v_h \, d\mathbf{x} + \sum_{e \in \mathcal{E}_h} \int_e \left( -\langle \beta \nabla v_h \cdot \mathbf{n}_e \rangle + \frac{\sigma_e}{h_e} v_h \right) g \, ds, \quad (7)$$

with  $\mathbf{n}_e$ ,  $\llbracket \cdot \rrbracket$  and  $\langle \cdot \rangle$  denoting the normal vector, the standard jump and average operators across the mesh edges  $\mathcal{E}_h$ , respectively.

In the construction of equation (5), if  $K$  is a non-interface element, the standard Legendre basis functions are used. If  $K$  is an interface element that intersected by  $\Gamma$  with corresponding fictitious Frenet element  $\hat{K}_F$ , the IFE basis functions are used and the 2D GC-IFE method is used to construct local IFE basis functions by the Frenet transformation. Let  $\mathbf{g}(\xi) = (g_1(\xi), g_2(\xi))^T$  be the regular parameterization of the interface curve  $\Gamma$ , and let  $\Gamma_K = \Gamma \cap K$  and  $\hat{\Gamma}_{K_F}$  denote the corresponding parameter domain of  $\Gamma_K$ . The unit tangent vector  $\tau(\xi)$  and the unit normal vector  $\mathbf{n}(\xi)$  are defined as:

$$\tau(\xi) = \frac{\mathbf{g}'(\xi)}{\|\mathbf{g}'(\xi)\|}, \quad \mathbf{n}(\xi) = \mathbf{Q}\tau(\xi), \quad \text{where } \mathbf{Q} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \quad (8)$$

The local Frenet coordinate mapping  $\mathbf{x}(\eta, \xi) = \mathbf{g}(\xi) + \eta \mathbf{n}(\xi)$  maps the curvilinear coordinates  $(\eta, \xi)$  to the physical coordinates  $\mathbf{x} = (x, y)^T$ , where  $\eta$  represents the signed normal distance to the interface. Under this transformation, the 2D Laplacian operator  $\Delta_{\mathbf{x}}$  in the curvilinear system becomes:

$$\Delta_{\mathbf{x}} u = \mathcal{L}(\hat{u}) := \hat{u}_{\eta\eta} + J_0(\eta, \xi) \hat{u}_{\xi\xi} + J_1(\eta, \xi) \hat{u}_{\eta\xi} + J_2(\eta, \xi) \hat{u}_{\xi\xi}, \quad (9)$$

where the metric coefficients  $J_0, J_1, J_2$  depend on the signed curvature  $\kappa(\xi)$  and the metric scaling factor  $\psi(\eta, \xi) = (1 + \eta\kappa(\xi))^{-1}$ .

To construct the local IFE space, a well-designed tensor-product Legendre polynomial basis  $\mathcal{R}_i(\eta, \xi) = q_i(\eta)p_s(\xi)$  [6] of degree  $m$  is defined. The piecewise local IFE basis functions  $\hat{\lambda}_j(\eta, \xi)$  are constructed as:

$$\hat{\lambda}_j(\eta, \xi) = \begin{cases} \sum_{i=1}^{(m+1)^2} C_{i,j}^- \mathcal{R}_i(\eta, \xi), & \eta < 0, \\ \sum_{i=1}^{(m+1)^2} C_{i,j}^+ \mathcal{R}_i(\eta, \xi), & \eta > 0. \end{cases} \quad (10)$$

The coefficients  $C^\pm = (C_{i,j}^\pm)$  are determined by enforcing the weak interface jump conditions on  $\hat{\Gamma}_{K_F}$ :

$$\int_{\hat{\Gamma}_{K_F}} \llbracket \hat{\lambda}_j(0, \xi) \rrbracket \hat{p}_{k-1}(\xi) \, d\xi = 0, \quad (11a)$$

$$\int_{\hat{\Gamma}_{K_F}} \llbracket \beta \partial_\eta \hat{\lambda}_j(0, \xi) \rrbracket \hat{p}_{k-1}(\xi) \, d\xi = 0, \quad (11b)$$

$$\int_{\hat{\Gamma}_{K_F}} \llbracket \beta \partial_\eta^n \mathcal{L} \hat{\lambda}_j(0, \xi) \rrbracket \hat{p}_{k-1}(\xi) \, d\xi = 0, \quad (11c)$$

for  $k = 1, \dots, m+1$  and  $n = 0, \dots, m-2$ . This yields the interface condition matrix system:

$$\tilde{\mathbf{A}} \mathbf{C}^+ = \tilde{\mathbf{J}} \mathbf{A} \mathbf{C}^-. \quad (12)$$

By solving the equation (12) and performing appropriate orthogonalization [7], we can obtain the IFE basis functions locally on an interface element with good orthogonality properties.

## 2.2. Key Changes for Dimensional Extension

Extending the GC-IFE method from 2D to 3D requires replacing the curve-based Frenet apparatus with surface-based differential geometry. Table 1 summarizes the mathematical correspondence between the 2D and 3D formulations.

While the formal mathematical extension summarized in Table 1 appears straightforward, the actual implementation requires substantial modifications in three key aspects:

**Table 1**

Comparison of mathematical formulations between 2D and 3D GC-IFE.

| Quantity               | Two-Dimensional (2D)                                             | Three-Dimensional (3D)                                                 |
|------------------------|------------------------------------------------------------------|------------------------------------------------------------------------|
| Interface $\Gamma$     | Curve $\mathbf{g}(\xi) \in \mathbb{R}^2$                         | Surface $\mathbf{S}(u, v) \in \mathbb{R}^3$                            |
| Normal coordinate      | $\eta \in [-d, d]$                                               | $\zeta \in [-d, d]$                                                    |
| Tangent coordinate(s)  | $\xi \in [\xi_0, \xi_1] \subset \mathbb{R}$                      | $(u, v) \in \hat{\Gamma}_{K_F} \subset \mathbb{R}^2$                   |
| Coordinate mapping     | $\mathbf{x}(\eta, \xi) = \mathbf{g}(\xi) + \eta \mathbf{n}(\xi)$ | $\mathbf{x}(u, v, \zeta) = \mathbf{S}(u, v) + \zeta \mathbf{n}(u, v)$  |
| Local polynomial space | $\mathcal{Q}_m(\eta) \otimes \mathbb{P}_m(\xi)$                  | $\mathcal{Q}_m(\zeta) \otimes \mathbb{P}_m(u) \otimes \mathbb{P}_m(v)$ |
| Basis dimension        | $(m+1)^2$                                                        | $(m+1)^3$                                                              |
| Jump test space        | $\mathbb{P}_m(\xi)$ on $\hat{\Gamma}_{K_F}$                      | $\mathbb{P}_m(u) \otimes \mathbb{P}_m(v)$ on $\hat{\Gamma}_{K_F}$      |
| Underlying operator    | $\Delta_{\mathbf{x}} u$                                          | $\Delta_{\mathbf{x}} u$                                                |
| Evaluation strategy    | Explicit curvilinear form $\mathcal{L}(\hat{u})$                 | Algebraic procedure via implicit differentiation                       |

1. **Geometric representation:** In 2D, the interface curve is parameterized by a global coordinate  $\xi$ , and the Frenet frame provides a natural tangent–normal decomposition. In 3D, the interface is a surface with two independent tangential directions. When a sufficiently regular global parameterization  $\mathbf{S}(u, v)$  is available, it can be used directly to define the surface coordinates, the normal coordinate, and the geometric quantities required in the basis-function construction. This direct parametric approach offers considerable convenience. However, such a global parameterization is generally unavailable for arbitrary interfaces, which motivates our local-geometric-reconstruction strategy. Using interface points, together with consistently oriented normals when available, we fit a local Monge graph on an orthonormal tangent plane. The height function, approximated via scaled Legendre polynomials, defines a signed local implicit function. The resulting Monge patch furnishes the local coordinates and geometric derivatives needed for constructing the IFE basis functions. In summary, the proposed 3D framework provides two alternative geometric pathways: the direct evaluation approach through a global parameterization when one is available, and the element-wise Legendre–Monge reconstruction approach for general interfaces.
2. **IFE basis construction and algebraic treatment of interface conditions:** In 2D, basis functions are constructed with a  $(m+1)^2$ -dimensional tensor-product polynomial space, and jump conditions are imposed on a one dimensional (1D) curve segment. In 3D, the underlying polynomial space expands to  $(m+1)^3$  dimensions, and the jumps are enforced on a 2D surface patch. The 3D weak jump conditions become:

$$\iint_{\hat{\Gamma}_{K_F}} \llbracket \hat{\lambda}_j(u, v, 0) \rrbracket P_k(u, v) \, dudv = 0, \quad (13a)$$

$$\iint_{\hat{\Gamma}_{K_F}} \llbracket \beta \partial_{\zeta} \hat{\lambda}_j(u, v, 0) \rrbracket P_k(u, v) \, dudv = 0, \quad (13b)$$

$$\iint_{\hat{\Gamma}_{K_F}} \llbracket \beta \partial_{\zeta}^n (\Delta_{\text{loc}} \hat{\lambda}_j(u, v, 0)) \rrbracket P_k(u, v) \, dudv = 0, \quad (13c)$$

where  $\Delta_{\text{loc}}$  denotes the 3D Laplacian in the local curvilinear coordinates,  $P_k(u, v) = p_{k_1}(u)p_{k_2}(v)$  represents the 2D tangent test polynomials of degree  $m$ . Furthermore, the 3D curvilinear Laplacian involves the full metric tensor, Christoffel symbols, and principal curvatures, making explicit symbolic derivation impractical. Our method replaces explicit expansion by an algebraic kernel that computes the required normal derivatives of  $\Delta_{\text{loc}}$  using the chain rule and the Leibniz rule, together with the variations of the metric tensor. This approach is directly programmable.

3. **Numerical integration:** In 2D, interface elements are cut by a curve, resulting in polygonal subdomains that can be conveniently triangulated. Standard Gaussian quadrature on the reference triangle can then be applied to each triangular subregion, possibly with one curved edge, through a suitable mapping. This approach is standard and efficient, as the subdivision and mapping are well-established in the finite element literature [8, 38, 39, 50, 53, 57, 62]. In 3D, however, the cut geometry becomes substantially more complex: a hexahedral element cut by a curved surface produces subdomains with curved faces and edges, making a direct extension of the triangulation strategy highly nontrivial and often limited to low-order accuracy. To overcome this difficulty, we adopt the dimension-reduction philosophy of Saye [54], which reduces multi-dimensional integrals over implicitly defined domains to a sequence of 1D integrations along coordinate lines.

The following three sections present the detailed implementation of each key aspect:

- **Section 3:** The Local surface representations and geometric derivative evaluation.
- **Section 4:** The construction of 3D IFE basis functions and the algebraic assembly of the interface condition matrix.
- **Section 5:** The matrix/vector assembly and numerical integration for the GA-IFE method within the discontinuous Galerkin framework is presented, including the Saye-type quadrature scheme for 3D interface elements.

### 3. Local surface representations and geometric derivative evaluation

This section describes two alternative approaches for representing interface surfaces in the 3D GA-IFE framework. When a sufficiently smooth global parameterization is available, the interface geometry is used directly. For interface surfaces of general geometry that do not admit such a global parameterization, a local signed Monge patch is reconstructed from high-order surface samples and, when available, consistently oriented normal data. Both approaches provide the local surface coordinates and the associated geometric derivatives required for constructing the IFE basis functions.

Analogous to the 2D GC-IFE method, each interface element  $K$  is associated with a tubular-neighborhood fictitious element  $K_F$  in physical space, which encloses the portion of the interface in a neighborhood of  $K$  and accommodates the interface sampling data. Via a local orthonormal-frame transformation, the physical-space region  $K_F$  is mapped to a fictitious element  $\hat{K}_F$  in the local coordinate system. Figure 1(b) shows the physical-space tubular neighborhood  $K_F$ , the interface surface segment, and the associated sampling points, whereas Figure 1(a) illustrates the corresponding fictitious element  $\hat{K}_F$  in local coordinates. This fictitious-element infrastructure is shared by both geometric strategies: the direct parametric approach and the Legendre–Monge reconstruction approach. It provides the foundation for the unified local-coordinate and elementwise-scaling framework used in the construction of IFE basis functions and in numerical integration.

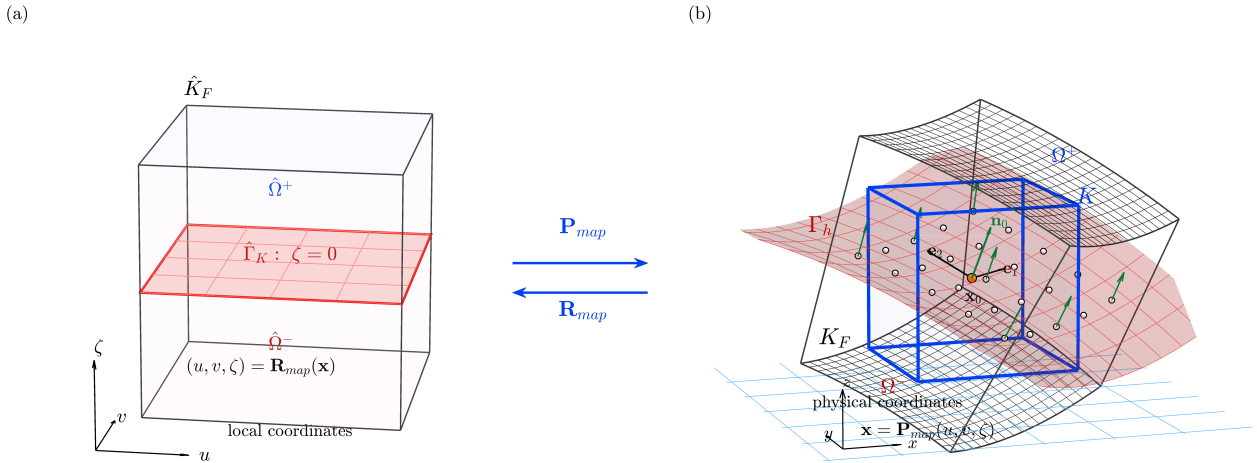


Figure 1: Schematic illustration of the local interface representation: (a) the fictitious element  $\hat{K}_F$  in the local coordinate space; and (b) the tubular neighborhood  $K_F$  of an element  $K$ , the local interface surface, and the sampling points in the physical space.

#### 3.1. Direct construction from a parameterized surface

When the interface  $\Gamma$  is given with a sufficiently smooth global parameterization, the surface geometry can be used directly. Let

$$\mathbf{S}_K(u, v) = \mathbf{S}(u, v), \quad (u, v) \in \hat{\Gamma}_K \subset \mathbb{R}^2, \quad (14)$$

denote the restriction of  $\mathbf{S}(u, v)$  onto the parameter region  $\hat{\Gamma}_K$  corresponding to the neighborhood of interface element  $K$  and the parameters  $(u, v)$  inherited from the global surface parameterization serve as tangential coordinates on the surface. The restricted parameter coordinates  $(u, v)$  are subsequently incorporated into the common local-coordinate and elementwise-scaling procedure described in Section 3.4.

On this parameter subdomain, we further assume that  $\mathbf{S}_K$  is sufficiently smooth and regular, and the orientation of the parameterization is selected consistently with the prescribed sign convention of the interface. The corresponding tangent vectors are

$$\mathbf{S}_{K,u} = \frac{\partial \mathbf{S}_K}{\partial u}, \quad \mathbf{S}_{K,v} = \frac{\partial \mathbf{S}_K}{\partial v}. \quad (15)$$

The second-order tangential derivatives required by the IFE basis-function construction are obtained directly from the parameterization:

$$\mathbf{S}_{K,uu} = \frac{\partial^2 \mathbf{S}_K}{\partial u^2}, \quad \mathbf{S}_{K,uv} = \frac{\partial^2 \mathbf{S}_K}{\partial u \partial v}, \quad \mathbf{S}_{K,vv} = \frac{\partial^2 \mathbf{S}_K}{\partial v^2}. \quad (16)$$

Then let

$$\mathbf{a} = \mathbf{S}_{K,u} \times \mathbf{S}_{K,v}, \quad J_\Gamma = \|\mathbf{a}\|, \quad (17)$$

and the oriented unit normal is defined by

$$\mathbf{n} = \frac{\mathbf{S}_{K,u} \times \mathbf{S}_{K,v}}{\|\mathbf{S}_{K,u} \times \mathbf{S}_{K,v}\|} = \frac{\mathbf{a}}{J_\Gamma}. \quad (18)$$

The derivatives of  $\mathbf{a}$  are

$$\mathbf{a}_u = \mathbf{S}_{K,uu} \times \mathbf{S}_{K,v} + \mathbf{S}_{K,u} \times \mathbf{S}_{K,uv}, \quad (19)$$

$$\mathbf{a}_v = \mathbf{S}_{K,uv} \times \mathbf{S}_{K,v} + \mathbf{S}_{K,u} \times \mathbf{S}_{K,vv}. \quad (20)$$

With the orthogonal projection

$$P = I - \mathbf{n}\mathbf{n}^\top, \quad (21)$$

the first derivatives of the unit normal are

$$\mathbf{n}_u = \frac{P\mathbf{a}_u}{J_\Gamma}, \quad \mathbf{n}_v = \frac{P\mathbf{a}_v}{J_\Gamma}. \quad (22)$$

To compute the second-order derivatives of the unit normal, the third derivatives of the surface parameterization are used. Define

$$\mathbf{a}_{uu} = \mathbf{S}_{K,uuu} \times \mathbf{S}_{K,v} + 2\mathbf{S}_{K,uu} \times \mathbf{S}_{K,uv} + \mathbf{S}_{K,u} \times \mathbf{S}_{K,uuv}, \quad (23)$$

$$\mathbf{a}_{uv} = \mathbf{S}_{K,uuv} \times \mathbf{S}_{K,v} + \mathbf{S}_{K,uu} \times \mathbf{S}_{K,vv} + \mathbf{S}_{K,uv} \times \mathbf{S}_{K,uv} + \mathbf{S}_{K,u} \times \mathbf{S}_{K,uvv}, \quad (24)$$

$$\mathbf{a}_{vv} = \mathbf{S}_{K,uvv} \times \mathbf{S}_{K,v} + 2\mathbf{S}_{K,uv} \times \mathbf{S}_{K,vv} + \mathbf{S}_{K,u} \times \mathbf{S}_{K,vvv}. \quad (25)$$

Then, for  $i, j \in \{u, v\}$ ,

$$\mathbf{n}_{ij} = \frac{P\mathbf{a}_{ij} - \mathbf{n}_j(\mathbf{n}^\top \mathbf{a}_i) - \mathbf{n}(\mathbf{n}_j^\top \mathbf{a}_i)}{J_\Gamma} - \frac{P\mathbf{a}_i(\mathbf{n}^\top \mathbf{a}_j)}{J_\Gamma^2}. \quad (26)$$

### 3.2. Signed local Monge reconstruction

When the interface  $\Gamma$  does not admit a sufficiently smooth global parameterization, we reconstruct a signed local Monge patch for  $\Gamma$  on each interface element using nearby interface sample points. The reconstruction proceeds in four steps: we first select a base point and construct a local orthonormal frame; we then express the neighboring interface samples in this local coordinate system; next, we fit a polynomial height function over the local tangent plane; finally, we use the fitted graph to define a local surface parameterization and a corresponding signed implicit representation.

Specifically, on an interface element  $K$ , the reconstruction starts from a set  $S_K^0$  of points locally sampled from the interface  $\Gamma$ :

$$S_K^0 = \{\mathbf{x}_p \mid \mathbf{x}_p \in \Gamma \cap K_F, \quad 1 \leq p \leq N_{S,0}\}.$$

The sample points in  $S_K^0$  may be generated by sampling a CAD surface or supplied directly as discrete input data. When available, consistently oriented unit normals associated with sample points in  $S_K^0$  may also be incorporated into the reconstruction. However, only the reference normal at the base point is required to construct the local Monge frame; normals at the remaining sample points are optional. Such additional normals provide first-order derivative information and can improve the reconstruction quality, whereas in their absence the procedure reduces to position-only fitting.

We next choose a base point  $\mathbf{x}_0 \in S_K^0$  near the barycenter of the sample set. Its associated unit normal  $\mathbf{n}_0$  is obtained from the CAD surface normal at  $\mathbf{x}_0$  when a CAD representation is available, and otherwise from the oriented discrete normal data. We then complete  $\mathbf{n}_0$  to a right-handed local orthonormal frame  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{n}_0\}$  satisfying

$$\mathbf{e}_1 \cdot \mathbf{e}_2 = 0, \quad \mathbf{e}_i \cdot \mathbf{n}_0 = 0, \quad \mathbf{e}_1 \times \mathbf{e}_2 = \mathbf{n}_0. \quad (27)$$

This frame provides the local coordinates in which the Monge graph is reconstructed. In particular, each sample point  $\mathbf{x}_p \in S_K^0$  is represented by

$$u_p = (\mathbf{x}_p - \mathbf{x}_0) \cdot \mathbf{e}_1, \quad v_p = (\mathbf{x}_p - \mathbf{x}_0) \cdot \mathbf{e}_2, \quad w_p = (\mathbf{x}_p - \mathbf{x}_0) \cdot \mathbf{n}_0. \quad (28)$$

To restrict the reconstruction to a locally coherent portion of the interface, we choose a threshold parameter  $\theta_{\max}$  and retain sample points whose normals are sufficiently aligned with the reference normal:

$$S_K = \{\mathbf{x}_p \in S_K^0, \quad \mathbf{n}_p \cdot \mathbf{n}_0 \geq \cos(\theta_{\max}) > 0\}. \quad (29)$$

The filtered sample set  $S_K$  is then used to construct a polynomial height function  $\eta_h^K(u, v)$ . Since a total-degree polynomial of degree at most  $m_g$  has

$$N_g = \frac{(m_g + 1)(m_g + 2)}{2}$$

coefficients, the sample set must contain sufficiently many points, namely  $|S_K| \geq N_g$ ; otherwise, additional interface sample points should be added.

Let

$$\ell_{K,1} = \max_{p \in S_K} (u_p) - \min_{p \in S_K} (u_p), \quad \ell_{K,2} = \max_{p \in S_K} (v_p) - \min_{p \in S_K} (v_p)$$

denote the tangential parameter ranges of the filtered sample set. We introduce the scaled local coordinates

$$\hat{u} = \frac{u}{\ell_{K,1}}, \quad \hat{v} = \frac{v}{\ell_{K,2}}, \quad (30)$$

and seek the height function as a total-degree Legendre expansion,

$$\eta_h^K(u, v) = \sum_{\substack{i,j \geq 0 \\ i+j \leq m_g}} a_{ij}^K p_i(\hat{u}) p_j(\hat{v}), \quad (31)$$

where  $p_i$  denotes the  $i$ -th standard Legendre polynomial. Thus,  $\eta_h^K$  is a bivariate polynomial of total degree at most  $m_g$ , represented in a Legendre basis.

When normal data are available, they can be used to supplement the position data with first-order derivative information. Let  $S_K^{(n)} \subseteq S_K$  denote the subset of samples for which consistently oriented normals  $\mathbf{n}_p$  are available. For a point  $\mathbf{x}_p \in S_K^{(n)}$ , the unit normal to the Monge graph  $w = \eta_h^K(u, v)$  is parallel to

$$-\eta_{h,u}^K(u_p, v_p) \mathbf{e}_1 - \eta_{h,v}^K(u_p, v_p) \mathbf{e}_2 + \mathbf{n}_0.$$

Matching this direction with the sampled normal  $\mathbf{n}_p$  yields the target slope values

$$q_{u,p} := \eta_{h,u}^K(u_p, v_p) = -\frac{\mathbf{n}_p \cdot \mathbf{e}_1}{\mathbf{n}_p \cdot \mathbf{n}_0}, \quad q_{v,p} := \eta_{h,v}^K(u_p, v_p) = -\frac{\mathbf{n}_p \cdot \mathbf{e}_2}{\mathbf{n}_p \cdot \mathbf{n}_0}. \quad (32)$$

We note that the denominator  $\mathbf{n}_p \cdot \mathbf{n}_0$  is bounded away from zero by the normal-filtering condition in (29).

The coefficient vector  $\mathbf{a}^K = \{a_{ij}^K\}$  is determined from the available position and normal data by the weighted least-squares problem

$$\begin{aligned} \mathbf{a}^K = \arg \min_{\mathbf{a}} \sum_{p \in S_K} \omega_p^{(0)} \left| \eta_h^K(u_p, v_p) - w_p \right|^2 \\ + \lambda_K^2 \sum_{p \in S_K^{(n)}} \omega_p^{(1)} \left[ \left| \partial_u \eta_h^K(u_p, v_p) - q_{u,p} \right|^2 + \left| \partial_v \eta_h^K(u_p, v_p) - q_{v,p} \right|^2 \right]. \end{aligned} \quad (33)$$

Here,  $\lambda_K = O(h_K)$  is a scaling factor that balances the dimensional difference between the height residuals and the slope residuals, while the weights  $\omega_p^{(0)}$  and  $\omega_p^{(1)}$  account for nonuniform sample density and the spatial distance from the patch center.

When oriented normals are available for some samples,  $S_K^{(n)}$  is nonempty and (33) gives a Legendre–Hermite reconstruction that incorporates derivative information. If no sample normals other than  $\mathbf{n}_0$  are available, we set  $S_K^{(n)} = \emptyset$ , so that the slope term vanishes and the reconstruction reduces to the position-only least-squares problem

$$\mathbf{a}^K = \arg \min_{\mathbf{a}} \sum_{p \in S_K} \omega_p^{(0)} \left| \eta_h^K(u_p, v_p) - w_p \right|^2. \quad (34)$$

In both cases, the resulting overdetermined linear system is solved using a column-scaled pivoted QR decomposition or a singular value decomposition (SVD) for numerical robustness.

Once the coefficients have been determined, the fitted height function represents the reconstructed interface locally as a graph over the tangent plane spanned by  $\mathbf{e}_1$  and  $\mathbf{e}_2$ . Lifting this graph back to physical space naturally yields the local surface parameterization

$$\mathbf{S}_h(u, v) = \mathbf{x}_0 + u\mathbf{e}_1 + v\mathbf{e}_2 + \eta_h^K(u, v)\mathbf{n}_0. \quad (35)$$

The image of  $\mathbf{S}_h$  forms the reconstructed local Monge patch. Thus, the height-function reconstruction not only approximates the interface geometry but also provides the local parameterization needed for subsequent geometric calculations.

The fitted height function  $\eta_h^K$  also provides a convenient signed implicit description of the reconstructed interface. For a point  $\mathbf{x}$  in the local neighborhood, define

$$\phi_h^K(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_0) \cdot \mathbf{n}_0 - \eta_h^K((\mathbf{x} - \mathbf{x}_0) \cdot \mathbf{e}_1, (\mathbf{x} - \mathbf{x}_0) \cdot \mathbf{e}_2). \quad (36)$$

Consequently, the reconstructed local interface patch associated with  $K$  is represented by

$$\Gamma_h^K = \{\mathbf{x} \in K_F : \phi_h^K(\mathbf{x}) = 0\}.$$

Its gradient is

$$\nabla_{\mathbf{x}} \phi_h^K = -\eta_{h,u}^K \mathbf{e}_1 - \eta_{h,v}^K \mathbf{e}_2 + \mathbf{n}_0, \quad (37)$$

which satisfies  $\nabla_{\mathbf{x}} \phi_h^K \cdot \mathbf{n}_0 = 1$ . Hence,  $\phi_h^K$  has a nonvanishing gradient, varies monotonically in the  $\mathbf{n}_0$  direction, and naturally inherits the sign convention of  $\Omega^\pm$  without requiring auxiliary offset layers.

### 3.3. Derivatives of the Monge map and unit normal

The local Monge map  $\mathbf{S}_h$  defined in (35) also provides the geometric quantities needed in the subsequent IFE construction. Since the local frame vectors  $\mathbf{e}_1$ ,  $\mathbf{e}_2$ , and  $\mathbf{n}_0$  are fixed with respect to the local coordinates  $(u, v)$ , the derivatives of  $\mathbf{S}_h$  can be evaluated directly from those of the fitted height function  $\eta_h^K$ . In particular, the first-order derivatives are

$$\mathbf{S}_{h,u} = \mathbf{e}_1 + \eta_{h,u}^K \mathbf{n}_0, \quad \mathbf{S}_{h,v} = \mathbf{e}_2 + \eta_{h,v}^K \mathbf{n}_0, \quad (38)$$

while the second-order derivatives are

$$\mathbf{S}_{h,uu} = \eta_{h,uu}^K \mathbf{n}_0, \quad \mathbf{S}_{h,uv} = \eta_{h,uv}^K \mathbf{n}_0, \quad \mathbf{S}_{h,vv} = \eta_{h,vv}^K \mathbf{n}_0. \quad (39)$$

All derivatives of  $\eta_h^K$  are evaluated analytically from the Legendre expansion (31).

The two tangent vectors in (38) determine the local surface normal. Using the orientation  $\mathbf{e}_1 \times \mathbf{e}_2 = \mathbf{n}_0$ , their cross product is

$$\mathbf{a}_h := \mathbf{S}_{h,u} \times \mathbf{S}_{h,v} = -\eta_{h,u}^K \mathbf{e}_1 - \eta_{h,v}^K \mathbf{e}_2 + \mathbf{n}_0. \quad (40)$$

Accordingly, the surface Jacobian is

$$J_{\Gamma,h} := \|\mathbf{a}_h\| = \sqrt{1 + (\eta_{h,u}^K)^2 + (\eta_{h,v}^K)^2}, \quad (41)$$

and the corresponding unit normal is

$$\mathbf{n}_h = \frac{\mathbf{a}_h}{J_{\Gamma,h}} = \frac{\nabla_{\mathbf{x}} \phi_h^K}{\|\nabla_{\mathbf{x}} \phi_h^K\|}. \quad (42)$$

Thus, the unit normal obtained from the Monge map agrees with that induced by the signed local implicit function (36).

We next derive the derivatives of  $\mathbf{n}_h$ . To differentiate the normalization in (42), define the orthogonal projection onto the tangent plane by

$$P_h = I - \mathbf{n}_h \mathbf{n}_h^\top. \quad (43)$$

For  $i \in \{u, v\}$ , differentiation of (40) gives

$$\mathbf{a}_{h,u} = -\eta_{h,uu}^K \mathbf{e}_1 - \eta_{h,uv}^K \mathbf{e}_2, \quad \mathbf{a}_{h,v} = -\eta_{h,uv}^K \mathbf{e}_1 - \eta_{h,vv}^K \mathbf{e}_2. \quad (44)$$

Using the standard derivative formula for a normalized vector, the first-order derivatives of the unit normal can then be written compactly as

$$\mathbf{n}_{h,i} = \frac{P_h \mathbf{a}_{h,i}}{J_{\Gamma,h}}, \quad i \in \{u, v\}. \quad (45)$$

To obtain the second-order derivatives of  $\mathbf{n}_h$ , we differentiate (45) once more. For  $i, j \in \{u, v\}$ , this yields

$$\mathbf{n}_{h,ij} = \frac{P_h \mathbf{a}_{h,ij} - \mathbf{n}_{h,j} (\mathbf{n}_h^\top \mathbf{a}_{h,i}) - \mathbf{n}_h (\mathbf{n}_{h,j}^\top \mathbf{a}_{h,i})}{J_{\Gamma,h}} - \frac{P_h \mathbf{a}_{h,i}}{J_{\Gamma,h}^2} (\mathbf{n}_h^\top \mathbf{a}_{h,j}). \quad (46)$$

The quantities  $\mathbf{a}_{h,ij}$  involve third-order derivatives of the fitted height function:

$$\mathbf{a}_{h,uu} = -\eta_{h,uuu}^K \mathbf{e}_1 - \eta_{h,uuv}^K \mathbf{e}_2, \quad (47)$$

$$\mathbf{a}_{h,uv} = -\eta_{h,uuv}^K \mathbf{e}_1 - \eta_{h,uvv}^K \mathbf{e}_2, \quad (48)$$

$$\mathbf{a}_{h,vv} = -\eta_{h,uvv}^K \mathbf{e}_1 - \eta_{h,vvv}^K \mathbf{e}_2. \quad (49)$$

Because  $\eta_h^K$  is represented in the Legendre basis, these higher-order derivatives are again evaluated analytically from (31).

### 3.4. Local coordinates and elementwise scaling

The preceding two subsections provide a local surface representation and the associated geometric derivatives for the reconstruction-based pathway. The same geometric information is available directly when a sufficiently smooth global parameterization of the interface is given. We now place these two pathways in a common local-coordinate framework, which will subsequently be used for the construction of the 3D IFE basis functions and for numerical integration.

To this end, let  $\mathbf{X}_K(u, v)$  denote the local surface parameterization associated with an interface element  $K$ . For the direct parametric pathway,  $\mathbf{X}_K$  is the restriction  $\mathbf{S}_K(u, v)$  of the given global parameterization to the local parameter patch associated with  $K$ ; for the reconstruction-based pathway,  $\mathbf{X}_K$  is the fitted Monge map  $\mathbf{S}_h(u, v)$  defined in (35). Let  $\mathbf{n}_K(u, v)$  denote the corresponding unit normal. Thus, the two geometric pathways differ only in how  $\mathbf{X}_K$  and  $\mathbf{n}_K$  are obtained, while the local-coordinate construction below is identical for both.

Using  $\mathbf{X}_K$  and  $\mathbf{n}_K$ , we define the local tubular-coordinate map

$$\mathbf{F}_K(u, v, \zeta) = \mathbf{X}_K(u, v) + \zeta \mathbf{n}_K(u, v). \quad (50)$$

Here,  $(u, v)$  are tangential surface coordinates and  $\zeta$  is the signed normal offset. Let  $\varepsilon > 0$  denote a half-width for a tubular neighborhood of  $\Gamma$  on which this normal-coordinate representation is smooth and one-to-one. We assume that the mesh is sufficiently fine so that the physical fictitious neighborhood  $K_F$  associated with each interface element  $K$  lies inside such a tubular neighborhood. Under this assumption, every point  $\mathbf{x} \in K_F$  admits a unique local representation

$$\mathbf{x} = \mathbf{F}_K(u, v, \zeta). \quad (51)$$

It remains to determine the local coordinates  $(u, v, \zeta)$  of a given point  $\mathbf{x} \in K_F$ . The tangential coordinates  $(u, v)$  are first obtained by projecting  $\mathbf{x}$  normally onto the local surface. Equivalently, they satisfy the closest-point system

$$\begin{aligned} (\mathbf{x} - \mathbf{X}_K(u, v)) \cdot \mathbf{X}_{K,u}(u, v) &= 0, \\ (\mathbf{x} - \mathbf{X}_K(u, v)) \cdot \mathbf{X}_{K,v}(u, v) &= 0. \end{aligned} \quad (52)$$

This nonlinear  $2 \times 2$  system is solved for  $(u, v)$  by Newton iteration. Its Jacobian involves the first- and second-order derivatives of  $\mathbf{X}_K$ , which are available analytically in both geometric pathways, see Subsection 3.1 and Subsection 3.3. Once  $(u, v)$  has been determined, the signed normal coordinate follows from

$$\zeta = (\mathbf{x} - \mathbf{X}_K(u, v)) \cdot \mathbf{n}_K(u, v). \quad (53)$$

Thus, solving (52) together with (53) provides the inverse of the tubular-coordinate map (50) on  $K_F$ .

The two geometric pathways use different initial guesses for the Newton iteration. For the Legendre–Monge reconstruction, the local orthonormal frame  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{n}_0\}$  introduced in Subsection 3.2 provides a natural tangent-plane approximation. We therefore set

$$u^{(0)} = (\mathbf{x} - \mathbf{x}_0) \cdot \mathbf{e}_1, \quad v^{(0)} = (\mathbf{x} - \mathbf{x}_0) \cdot \mathbf{e}_2. \quad (54)$$

For the direct parametric pathway, we follow the strategy used in the 2D GC–IFE method [5]. A set of surface sample points and their parameter values is precomputed on the local parameter patch associated with  $K$ . For a given point  $\mathbf{x}$ , the nearby precomputed surface sample provides an initial parameter pair  $(u^{(0)}, v^{(0)})$  for solving (52). With these pathway-specific initializations, the same Newton procedure is used thereafter.

We next use this pointwise coordinate transformation to define elementwise scales for the fictitious neighborhood  $K_F$ . Let  $\mathcal{X}_K = \{\mathbf{x}_j\}_{j=1}^8$  denote the eight vertices of the hexahedral interface element  $K$ . For each vertex  $\mathbf{x}_j \in \mathcal{X}_K$ , we compute its local coordinates  $(u_j, v_j, \zeta_j)$  from (52) and (53). The tangential midpoint and half-widths are then defined by

$$u_{\text{mid}} = \frac{\max_j u_j + \min_j u_j}{2}, \quad u_h = \frac{\max_j u_j - \min_j u_j}{2}, \quad (55)$$

$$v_{\text{mid}} = \frac{\max_j v_j + \min_j v_j}{2}, \quad v_h = \frac{\max_j v_j - \min_j v_j}{2}, \quad (56)$$

and the normal half-width is taken as

$$\zeta_h = \max_j |\zeta_j|. \quad (57)$$

These quantities characterize the elementwise extent of  $K$  in the local tangential and normal directions and provide the natural scales for the local IFE construction.

Accordingly, we introduce the normalized local coordinates

$$\tilde{u} = \frac{u - u_{\text{mid}}}{u_h}, \quad \tilde{v} = \frac{v - v_{\text{mid}}}{v_h}, \quad \tilde{\zeta} = \frac{\zeta}{\zeta_h}. \quad (58)$$

The same normalized coordinate system is therefore used for both the direct parametric and Legendre–Monge reconstruction pathways. In this way, the geometric description of the interface is separated from the subsequent IFE basis construction: once the local surface map and its derivatives are available, both pathways enter the same elementwise-scaled formulation.

**Remark 1.** *For the reconstruction-based pathway, the accuracy and robustness of the local-coordinate transformation depend on the quality of the Monge reconstruction as well as on the Newton projection. In particular, the reconstruction quality is governed jointly by the approximation order  $m$ , the geometric reconstruction order  $m_g$ , the mesh size, and the local interface curvature. If the curvature is large relative to the element size, the tangent-plane initial guess may become less accurate, and the Newton iteration may converge slowly, fail, or converge to an unintended point.*

*The required geometric accuracy also becomes more stringent as the approximation order increases. For a low-order discretization, such as  $m = 1$ , taking  $m_g = m$  may still be adequate under moderate curvature. For a higher-order method, however, errors in either the reconstructed geometry or the closest-point projection can dominate the discretization error and prevent the optimal convergence rate from being observed. Highly curved regions may therefore require a finer mesh, a smaller effective reconstruction patch, an improved Newton initial guess, and/or a geometric reconstruction order satisfying  $m_g > m$ .*

*Finally, the interface sampling region used in the Monge reconstruction must extend sufficiently far in the tangential directions so that the resulting physical fictitious neighborhood  $K_F$  contains the entire interface element  $K$ . This ensures that the local-coordinate transformation, reconstructed surface geometry, and subsequent quadrature are well defined throughout  $K$ .*

## 4. Local IFE Basis Functions and Interface Condition Matrix

This section presents the construction of local IFE basis functions on 3D interface elements. The construction follows the same extension principle as in the 2D setting [5]. We first introduce a polynomial basis in local curvilinear coordinates, then formulate the interface-condition matrix used to enforce the jump conditions weakly over the interface patch, and finally develop the algebraic machinery for computing the required normal derivatives of the Laplacian in physical coordinates.

### 4.1. Local polynomial basis in local curvilinear coordinates

For each interface element  $K \subset \mathbb{R}^3$  intersected by  $\Gamma$ , we establish a local curvilinear coordinate system  $(u, v, \zeta)$  as described in Section 3.4. The interface patch  $\Gamma_K = \Gamma \cap K$  corresponds to the zero-level surface  $\zeta = 0$ , and the two subdomains within  $K$  are distinguished by the sign of  $\zeta$ :  $\zeta < 0$  in  $K^- = K \cap \Omega^-$  and  $\zeta > 0$  in  $K^+ = K \cap \Omega^+$ .

Let  $\mathbb{Q}_m$  denote the space of tensor-product polynomials of degree at most  $m$  in the three local coordinates  $(u, v, \zeta)$ . We choose a basis  $\{R_i(u, v, \zeta)\}_{i=1}^{N_b}$  for  $\mathbb{Q}_m$ , where  $N_b = (m+1)^3$ , in the tensor-product form:

$$R_i(u, v, \zeta) = q_t(\zeta) P_s(u, v), \quad i = s + tN_t + 1, \quad (59)$$

with  $N_t = (m+1)^2$ ,  $s = 0, \dots, N_t - 1$ ,  $t = 0, \dots, m$ . The tangent basis functions  $P_s(u, v)$  are chosen as the tensor products of 1D Legendre polynomials:

$$P_s(u, v) = p_{i_1} \left( \frac{u - u_{\text{mid}}}{u_h} \right) p_{i_2} \left( \frac{v - v_{\text{mid}}}{v_h} \right), \quad i_1, i_2 = 0, \dots, m, \quad (60)$$

where  $s = i_1(m+1) + i_2 + 1$ . Here  $u_{\text{mid}}$ ,  $v_{\text{mid}}$ ,  $u_h$ , and  $v_h$  are the local geometric scaling parameters defined in Section 3.4 that map the tangent coordinates to the standard interval  $[-1, 1]$ .

The normal basis functions  $q_t(\zeta)$  are constructed to have the desirable property that  $q_t(0) = 0$  for  $t \geq 1$  and  $q'_t(0) = 0$  for  $t \geq 2$ , which simplifies the enforcement of the jump conditions on the interface. Following the 2D GC-IFE [6], we define:

$$q_0(\zeta) = 1, \quad q_1(\zeta) = p_1\left(\frac{\zeta}{\zeta_h}\right), \quad q_t(\zeta) = \hat{q}_t\left(\frac{\zeta}{\zeta_h}\right), \quad t \geq 2, \quad (61)$$

where  $\zeta_h$  is the maximum normal deviation defined in Section 3.4, and  $\hat{q}_t(x)$  for  $t \geq 2$  is given by:

$$\hat{q}_t(x) = (2t-1)x p_{t-1}(x) - (2t-1)p_{t-1}(0)x, \quad t \geq 2. \quad (62)$$

These polynomials satisfy  $\hat{q}_t(0) = \hat{q}'_t(0) = 0$  for  $t \geq 2$  and  $q_1(0) = 0$ , which are crucial for the stability and efficiency of the basis construction.

#### 4.2. The IFE space and construction of basis functions

Following the 2D GC-IFE framework, we define the IFE space on the local fictitious element  $\hat{K}_F$  as:

$$\hat{V}_\beta^m(\hat{K}_F) = \left\{ \hat{\lambda} : \hat{K}_F \rightarrow \mathbb{R} \mid \hat{\lambda}|_{\hat{K}_F^\pm} \in \mathbb{Q}_m(\hat{K}_F^\pm), \hat{\lambda} \text{ satisfies the jump conditions (13)} \right\}, \quad (63)$$

where  $\hat{K}_F^\pm = \{\zeta \gtrless 0\}$ . On the physical interface element  $K$ , the IFE space is defined via the pull-back through the local coordinate mapping:

$$V_\beta^m(K) = \left\{ \lambda : K \rightarrow \mathbb{R} \mid \lambda = \hat{\lambda} \circ \mathbf{x}^{-1}, \hat{\lambda} \in \hat{V}_\beta^m(\hat{K}_F) \right\}, \quad (64)$$

where the mapping  $\mathbf{x} = \mathbf{F}_K(u, v, \zeta)$  is defined by (51).

The well-posedness of this space follows from the same argument as in the 2D case (Lemma 1 in [5]). For any polynomial  $\hat{\lambda}^- \in \mathbb{Q}_m(\hat{K}_F^-)$  on one side of the interface, there exists a unique polynomial  $\hat{\lambda}^+ \in \mathbb{Q}_m(\hat{K}_F^+)$  on the other side such that the combined function  $\hat{\lambda}$  satisfies the weak jump conditions (13). This guarantees that the dimension of the IFE space is  $N_b = (m+1)^3$ , and the extension procedure described below is well-posed.

To construct a basis for this space, the IFE basis functions  $\hat{\lambda}_j(u, v, \zeta)$ ,  $j = 1, \dots, N_b$ , are defined piecewise on the two subdomains within the fictitious element:

$$\hat{\lambda}_j(u, v, \zeta) = \begin{cases} \hat{\lambda}_j^-(u, v, \zeta) = \sum_{i=1}^{N_b} C_{i,j}^- R_i(u, v, \zeta), & \zeta < 0, \\ \hat{\lambda}_j^+(u, v, \zeta) = \sum_{i=1}^{N_b} C_{i,j}^+ R_i(u, v, \zeta), & \zeta > 0, \end{cases} \quad (65)$$

where  $\mathbf{C}^- = [C_{i,j}^-]$  and  $\mathbf{C}^+ = [C_{i,j}^+]$  are  $N_b \times N_b$  coefficient matrices to be determined.

Without loss of generality, assume that  $|K^-| \geq |K^+|$ . Then, the construction proceeds by choosing a basis  $\{\hat{\lambda}_j^-\}_{j=1}^{N_b}$  for the polynomial space  $\mathbb{Q}_m$  on  $\hat{K}_F^-$ , and then extending these functions across the interface by enforcing the jump conditions to determine  $\{\hat{\lambda}_j^+\}_{j=1}^{N_b}$ . This extension procedure is well-posed because, for any polynomial on one side, the jump conditions uniquely determine the polynomial on the other side. Computationally, we set  $\mathbf{C}^-$  to the identity matrix or another nonsingular matrix and solve for the other coefficient matrix  $\mathbf{C}^+$  from the interface condition matrix system derived in the following subsection.

#### 4.3. Interface condition matrix

The interface jump conditions, expressed in the weak form on the interface patch  $\Gamma_K$ , consist of the continuity of the solution, the continuity of the normal flux, and the continuity of the higher-order normal derivatives of the governing elliptic operator. In the local curvilinear coordinates, these conditions take the form (13). Substituting the piecewise

polynomial expansion (65) into the jump conditions (13) yields the linear system (12) for the coefficient matrices  $\mathbf{C}^-$  and  $\mathbf{C}^+$ , where the interface condition matrix  $\tilde{\mathbf{A}}$  can be expressed in the following block matrix:

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \tilde{\mathbf{A}}^{(0)} \\ \tilde{\mathbf{A}}^{(1)} \\ \vdots \\ \tilde{\mathbf{A}}^{(m-2)} \end{bmatrix}, \quad (66)$$

and the jump scaling matrix  $\mathbf{J}_\beta$  is defined as:

$$\mathbf{J}_\beta = \text{diag} \left( \underbrace{1, \dots, 1}_{N_t}, \underbrace{\frac{\beta^-}{\beta^+}, \dots, \frac{\beta^-}{\beta^+}}_{mN_t} \right). \quad (67)$$

The entries of the matrix  $\tilde{\mathbf{A}}$  are defined as follows. For the zeroth-order block  $\mathbf{B}_0$  (continuity of the solution):

$$(\mathbf{B}_0)_{k,i} = \iint_{\hat{\Gamma}_K} R_i(u, v, 0) P_k(u, v) du dv, \quad (68)$$

where  $k = 1, \dots, N_t$  and  $i = 1, \dots, N_b$ .

For the first-order block  $\mathbf{B}_1$  (normal flux continuity):

$$(\mathbf{B}_1)_{k,i} = \iint_{\hat{\Gamma}_K} \partial_\zeta R_i(u, v, 0) P_k(u, v) du dv. \quad (69)$$

For the higher-order blocks  $\tilde{\mathbf{A}}^{(n)}$ ,  $n = 0, \dots, m-2$ :

$$(\tilde{\mathbf{A}}^{(n)})_{k,i} = \iint_{\hat{\Gamma}_K} \partial_\zeta^n (\Delta_{\text{loc}} R_i(u, v, 0)) P_k(u, v) du dv. \quad (70)$$

Following the same argument as in the 2D case, the matrix  $\tilde{\mathbf{A}}$  is nonsingular. This ensures that the extension procedure is well-posed: for any choice of  $\mathbf{C}^-$ , the coefficient matrix  $\mathbf{C}^+$  is uniquely determined by (12).

**Remark 2.** We emphasize that all integrals in (68), (69) and (70) are evaluated on the parameter domain  $\hat{\Gamma}_K$  rather than on the physical surface  $\Gamma_K$ . This parametric formulation is adopted intentionally for the local basis construction: it bypasses the evaluation of the surface Jacobian  $J(u, v) = \|\mathbf{g}_u \times \mathbf{g}_v\|$  in the local system, significantly reducing computational cost and improving numerical stability. The smoothness of the Monge patch mapping ensures that this parametric  $L^2$  projection is mathematically equivalent to the physical surface integration.

#### 4.4. Algebraic derivation of Laplacian normal derivatives

A key challenge in assembling the blocks  $\tilde{\mathbf{A}}^{(n)}$  is the evaluation of the normal derivatives of the Laplacian in physical coordinates,  $\partial_\zeta^n (\Delta_{\text{loc}} R(u, v, 0))$ . A direct expansion of  $\Delta_{\text{loc}}$  in the curvilinear coordinates  $\mathbf{u} = (u, v, \zeta)^\top$  can become algebraically cumbersome when the value of  $m$  becomes large, as it involves Christoffel symbols and their derivatives. To avoid this complexity, we develop a recursive algebraic formulation based on the chain rule and the Leibniz rule, using the geometric quantities derived in Section 3.1 or Section 3.3.

Let  $\mathbf{x}(u, v, \zeta)$  be the coordinate transformation defined by (51) for mapping the local curvilinear space to the physical space. At the interface ( $\zeta = 0$ ), the Jacobian matrix  $\mathbf{J}_0$  and its inverse  $\mathbf{M}_0$  are given by:

$$\mathbf{J}_0 = [\partial_u \mathbf{x} \quad \partial_v \mathbf{x} \quad \partial_\zeta \mathbf{x}]|_{\zeta=0} = [\mathbf{S}_u \quad \mathbf{S}_v \quad \mathbf{n}], \quad \mathbf{M}_0 = \mathbf{J}_0^{-1}. \quad (71)$$

Since the mapping  $\mathbf{x}(u, v, \zeta)$  is linear with respect to the transverse coordinate  $\zeta$ , the higher-order normal derivatives of the Jacobian matrix  $\mathbf{J}_{k,0} = \partial_\zeta^k \mathbf{J}_0$  simplify significantly:

$$\mathbf{J}_{1,0} = [\mathbf{n}_u \quad \mathbf{n}_v \quad \mathbf{0}], \quad \mathbf{J}_{k,0} = \mathbf{0} \quad (\forall k \geq 2). \quad (72)$$

To obtain the normal derivatives of the inverse Jacobian matrix  $\mathbf{M}_0^{(k)} := \partial_\zeta^k \mathbf{J}_0^{-1}$ , we differentiate the identity  $\mathbf{M}_0 \mathbf{J}_0 = \mathbf{I}$  recursively. Applying the Leibniz rule yields the following recurrence relation:

$$\mathbf{M}_0^{(0)} = \mathbf{M}_0, \quad \mathbf{M}_0^{(k+1)} = -(k+1) \mathbf{M}_0^{(k)} \mathbf{J}_{1,0} \mathbf{M}_0 \quad (k \geq 0). \quad (73)$$

Consequently, the  $q$ -th normal derivative of the inverse metric tensor  $\mathbf{G}_0^{-1} := \mathbf{M}_0 \mathbf{M}_0^\top$ , denoted by  $\text{inv}\mathbf{G}_0^{(q)} := \partial_\zeta^q \mathbf{G}_0^{-1}$ , can be computed via the Cauchy product:

$$\text{inv}\mathbf{G}_0^{(q)} = \sum_{a=0}^q \binom{q}{a} \mathbf{M}_0^{(a)} (\mathbf{M}_0^{(q-a)})^\top. \quad (74)$$

To transform the Laplacian operator, we define the Hessian matrix of the physical coordinates with respect to the curvilinear coordinates. Let  $\mathbf{H}_c(\zeta) = \nabla_{\mathbf{u}}^2 x_c$  be the Hessian of the  $c$ -th physical coordinate ( $c \in \{x, y, z\}$ ). Due to the linearity of the mapping in  $\zeta$ , we write  $\mathbf{H}_c(\zeta) = \mathbf{H}_{c,0} + \zeta \mathbf{H}_{c,1}$ , where:

$$\mathbf{H}_{c,0} = \begin{bmatrix} (\mathbf{S}_{uu})_c & (\mathbf{S}_{uv})_c & (\mathbf{n}_u)_c \\ (\mathbf{S}_{uv})_c & (\mathbf{S}_{vv})_c & (\mathbf{n}_v)_c \\ (\mathbf{n}_u)_c & (\mathbf{n}_v)_c & 0 \end{bmatrix}, \quad \mathbf{H}_{c,1} = \begin{bmatrix} (\mathbf{n}_{uu})_c & (\mathbf{n}_{uv})_c & 0 \\ (\mathbf{n}_{uv})_c & (\mathbf{n}_{vv})_c & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (75)$$

We then introduce the auxiliary vector  $\mathbf{B}^{(q)} \in \mathbb{R}^3$ , which represents the geometric contribution of the coordinate curvature to the Laplacian:

$$B_c^{(q)} = \text{tr} \left( \text{inv}\mathbf{G}_0^{(q)} \mathbf{H}_{c,0} \right) + q \text{tr} \left( \text{inv}\mathbf{G}_0^{(q-1)} \mathbf{H}_{c,1} \right), \quad c \in \{x, y, z\}, \quad (76)$$

where the second term vanishes for  $q = 0$ .

By applying the chain rule to the identity  $\Delta_{\mathbf{x}} \mathbf{x} = \mathbf{0}$ , the Laplacian of the curvilinear coordinates  $\mathbf{u}$  at  $\zeta = 0$  is expressed as  $\Delta_{\mathbf{x}} \mathbf{u}|_{\zeta=0} = -\mathbf{M}_0 \mathbf{B}^{(0)}$ . Differentiating this relation  $q$  times with respect to  $\zeta$  yields:

$$\partial_\zeta^q (\Delta_{\mathbf{x}} \mathbf{u})|_{\zeta=0} = - \sum_{a=0}^q \binom{q}{a} \mathbf{M}_0^{(a)} \mathbf{B}^{(q-a)}. \quad (77)$$

For any local tensor-product basis function  $R(u, v, \zeta) = q_t(\zeta) P_s(u, v)$ , its physical Laplacian  $\Delta_{\mathbf{x}} R$  at  $\zeta = 0$  is expanded via the chain rule as:

$$\Delta_{\mathbf{x}} R|_{\zeta=0} = \text{tr} \left( \nabla_{\mathbf{u}}^2 R|_{\zeta=0} \mathbf{G}_0^{-1} \right) + \left( \nabla_{\mathbf{u}} R|_{\zeta=0} \right)^\top \left( \Delta_{\mathbf{x}} \mathbf{u}|_{\zeta=0} \right). \quad (78)$$

Finally, by applying the Leibniz rule to differentiate Eq. (78)  $n$  times with respect to  $\zeta$ , we obtain the desired recursive formula at the interface ( $\zeta = 0$ ):

$$\begin{aligned} \partial_\zeta^n (\Delta_{\text{loc}} R(u, v, 0)) &= \partial_\zeta^n (\Delta_{\mathbf{x}} R)|_{\zeta=0} = \sum_{q=0}^n \binom{n}{q} \left[ \text{tr} \left( \partial_\zeta^{n-q} (\nabla_{\mathbf{u}}^2 R)|_{\zeta=0} \text{inv}\mathbf{G}_0^{(q)} \right) \right. \\ &\quad \left. + \left( \partial_\zeta^{n-q} (\nabla_{\mathbf{u}} R)|_{\zeta=0} \right)^\top \partial_\zeta^q (\Delta_{\mathbf{x}} \mathbf{u})|_{\zeta=0} \right]. \end{aligned} \quad (79)$$

Eq. (79) serves as an efficient and algebraic machinery for computations involving the high-order interface matching conditions. The terms  $\partial_\zeta^{n-q} (\nabla_{\mathbf{u}}^2 R)|_{\zeta=0}$  and  $\partial_\zeta^{n-q} (\nabla_{\mathbf{u}} R)|_{\zeta=0}$  are computed directly from the analytical derivatives of the 1D Legendre polynomials  $q_t(\zeta)$  and the 2D polynomials  $P_s(u, v)$ , ensuring both efficiency and high precision.

**Remark 3.** Algorithm 1 in Appendix A illustrates the complete construction procedure of the IFE basis functions. For the orthogonalization of the IFE basis functions, we used the exact same method as in [6].

## 5. Matrix/Vector assembly and numerical integration within the DG framework

This section presents the matrix and vector assembly and numerical integration procedures for the proposed GA-IFE method. We first present the algebraic matrix/vector form of the SIPG discretization. We then describe the hierarchical line-splitting quadrature for both volume and face integrals.

### 5.1. Discrete DG formulation and matrix assembly

Let  $\mathcal{T}_h$  be a partition of the domain  $\Omega$  into hexahedral elements, and let  $\mathcal{E}_h = \mathcal{E}_h^i \cup \mathcal{E}_h^b$  denote the set of all mesh faces, where  $\mathcal{E}_h^i$  and  $\mathcal{E}_h^b$  represent the interior and boundary faces, respectively. We partition the elements into the set of non-interface elements  $\mathcal{T}_h^n$  and the set of interface elements  $\mathcal{T}_h^i$ . Accordingly, the mesh faces are classified into four disjoint subsets:

- **Regular-interior faces**  $\mathcal{E}_h^{in}$ : faces shared by two non-interface elements  $K_1, K_2 \in \mathcal{T}_h^n$ .
- **Interface-affected interior faces**  $\mathcal{E}_h^{ii}$ : interior faces where at least one adjacent element belongs to  $\mathcal{T}_h^i$ .
- **Regular-boundary faces**  $\mathcal{E}_h^{bn}$ : boundary faces within non-interface elements  $K \in \mathcal{T}_h^n$ .
- **Interface-affected boundary faces**  $\mathcal{E}_h^{bi}$ : boundary faces within interface elements  $K \in \mathcal{T}_h^i$ .

The global bilinear form (5) leads to the following global discrete system:

$$(\mathbf{M}_{\text{vol}} + \mathbf{M}_{\text{flux}} + \mathbf{M}_{\text{pen}}) \mathbf{u} = \mathbf{f}, \quad (80)$$

where the global volume stiffness matrix  $\mathbf{M}_{\text{vol}}$ , face flux matrix  $\mathbf{M}_{\text{flux}}$ , penalty matrix  $\mathbf{M}_{\text{pen}}$ , and load vector  $\mathbf{f}$  are assembled from their local element and face counterparts:

$$\mathbf{M}_{\text{vol}} = \sum_{K \in \mathcal{T}_h} \mathbf{K}_{\text{vol}}^K, \quad \mathbf{M}_{\text{flux}} = \sum_{e \in \mathcal{E}_h} \mathbf{K}_{\text{flux}}^e, \quad \mathbf{M}_{\text{pen}} = \sum_{e \in \mathcal{E}_h} \mathbf{K}_{\text{pen}}^e, \quad (81)$$

$$\mathbf{f} = \sum_{K \in \mathcal{T}_h} \mathbf{f}^K + \sum_{e \in \mathcal{E}_h^b} \mathbf{f}^e, \quad (82)$$

where, the  $\sum$  is in the standard sense of merging the local degrees of freedom into the global degrees of freedom in finite element computations.

For a standard element  $K \in \mathcal{T}_h^n$ , or a face  $e \in \mathcal{E}_h^{in}$ , or a face  $e \in \mathcal{E}_h^{bn}$ , the calculations of the local matrices and vectors are standard (see, e.g., [34]).

For an interface element  $K \in \mathcal{T}_h^i$ , the local volume stiffness matrix and local vector are computed using the IFE basis functions  $\{\lambda_i^\pm\}$ :

$$(\mathbf{M}_{\text{vol}}^K)_{ij} = \int_{K^-} \beta^- \nabla \lambda_i^- \cdot \nabla \lambda_j^- \, d\mathbf{x} + \int_{K^+} \beta^+ \nabla \lambda_i^+ \cdot \nabla \lambda_j^+ \, d\mathbf{x}, \quad (83)$$

$$(\mathbf{f}^K)_i = \int_{K^-} f \lambda_i^- \, d\mathbf{x} + \int_{K^+} f \lambda_i^+ \, d\mathbf{x}. \quad (84)$$

For the interface-affected interior and boundary faces, the local matrices and vectors are computed by distinguishing among the following cases.

*Case 1: Interface-affected interior faces* ( $e \in \mathcal{E}_h^{ii}$ ) For an interior face  $e = \partial K_1 \cap \partial K_2$  where at least one adjacent element belongs to  $\mathcal{T}_h^i$ , we define a unified local basis function  $\psi_i$  on  $K \in \{K_1, K_2\}$  to simplify the notation:

$$\psi_i|_K = \begin{cases} \lambda_i^s, & \text{if } K \in \mathcal{T}_h^i \text{ (interface element),} \\ \varphi_i, & \text{if } K \in \mathcal{T}_h^n \text{ (regular element),} \end{cases} \quad (85)$$

where  $\lambda_i^s$  ( $s \in \{-, +\}$ ) denotes the active piece of the IFE basis function in the subdomain containing the face  $e$ , and  $\varphi_i$  denotes the standard Legendre basis function.

**Case 1a: Uncut interior faces ( $e \cap \Gamma = \emptyset$ )** In this case,  $e$  lies entirely within one subdomain  $\Omega^s$  ( $s \in \{-, +\}$ ), the diffusion coefficient is constant on the face ( $\beta_{K_1} = \beta_{K_2} = \beta^s$ ). The local matrices are:

$$(\mathbf{M}_{\text{flux}}^e)_{ij} = - \int_e \left( \langle \beta^s \nabla \psi_i \cdot \mathbf{n}_e \rangle \llbracket \psi_j \rrbracket + \langle \beta^s \nabla \psi_j \cdot \mathbf{n}_e \rangle \llbracket \psi_i \rrbracket \right) ds, \quad (86)$$

$$(\mathbf{M}_{\text{pen}}^e)_{ij} = \int_e \frac{\sigma_e}{h_e} \llbracket \psi_i \rrbracket \llbracket \psi_j \rrbracket ds. \quad (87)$$

**Case 1b: Cut interior faces ( $e \cap \Gamma \neq \emptyset$ )** In this case,  $e$  is intersected by  $\Gamma$ , it is partitioned into  $e^- = e \cap \Omega^-$  and  $e^+ = e \cap \Omega^+$ . Both adjacent elements are interface elements, and the integration is split:

$$(\mathbf{M}_{\text{flux}}^e)_{ij} = - \int_{e^-} \left( \langle \beta^- \nabla \lambda_i^- \cdot \mathbf{n}_e \rangle \llbracket \lambda_j^- \rrbracket + \langle \beta^- \nabla \lambda_j^- \cdot \mathbf{n}_e \rangle \llbracket \lambda_i^- \rrbracket \right) ds \\ - \int_{e^+} \left( \langle \beta^+ \nabla \lambda_i^+ \cdot \mathbf{n}_e \rangle \llbracket \lambda_j^+ \rrbracket + \langle \beta^+ \nabla \lambda_j^+ \cdot \mathbf{n}_e \rangle \llbracket \lambda_i^+ \rrbracket \right) ds, \quad (88)$$

$$(\mathbf{M}_{\text{pen}}^e)_{ij} = \int_{e^-} \frac{\sigma_e}{h_e} \llbracket \lambda_i^- \rrbracket \llbracket \lambda_j^- \rrbracket ds + \int_{e^+} \frac{\sigma_e}{h_e} \llbracket \lambda_i^+ \rrbracket \llbracket \lambda_j^+ \rrbracket ds. \quad (89)$$

**Case 2: Interface-affected boundary faces ( $e \in \mathcal{E}_h^{bi}$ )** For a boundary face  $e = \partial K \cap \partial \Omega$  with  $K \in \mathcal{T}_h^i$ , the Dirichlet boundary condition  $u = g_D$  is weakly enforced using the GA-IFE basis functions. The formulation depends on whether the face  $e$  itself is intersected by the interface  $\Gamma$ .

**Case 2a: Uncut boundary faces ( $e \cap \Gamma = \emptyset$ )** In this case, the boundary face  $e$  is not intersected by  $\Gamma$ , it lies entirely within one subdomain  $\Omega^s$  ( $s \in \{-, +\}$ ). The local matrices and boundary load vector are:

$$(\mathbf{M}_{\text{flux}}^e)_{ij} = - \int_e \left( (\beta^s \nabla \lambda_i^s \cdot \mathbf{n}_e) \lambda_j^s + (\beta^s \nabla \lambda_j^s \cdot \mathbf{n}_e) \lambda_i^s \right) ds, \quad (90)$$

$$(\mathbf{M}_{\text{pen}}^e)_{ij} = \int_e \frac{\sigma_e}{h_e} \lambda_i^s \lambda_j^s ds, \quad (91)$$

$$(\mathbf{f}^e)_i = \int_e \left( -\beta^s \nabla \lambda_i^s \cdot \mathbf{n}_e + \frac{\sigma_e}{h_e} \lambda_i^s \right) g_D ds. \quad (92)$$

**Case 2b: Cut boundary faces ( $e \cap \Gamma \neq \emptyset$ )** In this case, the boundary face  $e$  is intersected by  $\Gamma$ , it is split into  $e^- = e \cap \Omega^-$  and  $e^+ = e \cap \Omega^+$ . The integration is split accordingly:

$$(\mathbf{M}_{\text{flux}}^e)_{ij} = - \int_{e^-} \left( (\beta^- \nabla \lambda_i^- \cdot \mathbf{n}_e) \lambda_j^- + (\beta^- \nabla \lambda_j^- \cdot \mathbf{n}_e) \lambda_i^- \right) ds \\ - \int_{e^+} \left( (\beta^+ \nabla \lambda_i^+ \cdot \mathbf{n}_e) \lambda_j^+ + (\beta^+ \nabla \lambda_j^+ \cdot \mathbf{n}_e) \lambda_i^+ \right) ds, \quad (93)$$

$$(\mathbf{M}_{\text{pen}}^e)_{ij} = \int_{e^-} \frac{\sigma_e}{h_e} \lambda_i^- \lambda_j^- ds + \int_{e^+} \frac{\sigma_e}{h_e} \lambda_i^+ \lambda_j^+ ds, \quad (94)$$

$$(\mathbf{f}^e)_i = \int_{e^-} \left( -\beta^- \nabla \lambda_i^- \cdot \mathbf{n}_e + \frac{\sigma_e}{h_e} \lambda_i^- \right) g_D ds + \int_{e^+} \left( -\beta^+ \nabla \lambda_i^+ \cdot \mathbf{n}_e + \frac{\sigma_e}{h_e} \lambda_i^+ \right) g_D ds. \quad (95)$$

## 5.2. Saye-type quadrature for volume and face integrals

To evaluate the volume integrals in Eqs. (83)–(84) and face integrals in Eqs. (86)–(95), we adopt a Saye-type recursive dimension-reduction quadrature scheme [54] with hierarchical subdivision and 1D root-finding, which supports both 3D volume integration and 2D face integration over element faces. A unified recursive dimension-reduction framework is employed, whereas interface intersection and phase classification are implemented differently depending on the underlying geometric representation.

Let  $\mathcal{D} \subset \mathbb{R}^d$  denote the root integration box, with  $d = 3$  for volume integrals and  $d = 2$  for face integrals. For assembling the local matrices and vectors on an element  $K$ ,  $\mathcal{D} = K$ , and for assembling the local matrices and vectors on an element face  $e$ ,  $\mathcal{D} = e$ . The integration domain is partitioned into two phase subdomains

$$\mathcal{D}^- = \mathcal{D} \cap \Omega^-, \mathcal{D}^+ = \mathcal{D} \cap \Omega^+.$$

On each phase subdomain, the same quadrature procedure is invoked independently under the corresponding phase condition, yielding two sets of quadrature points and weights,

$$\mathcal{Q}^- = \left\{ (\mathbf{x}_q^-, \omega_q^-) \right\}_{q=1}^{N^-}, \quad \mathcal{Q}^+ = \left\{ (\mathbf{x}_q^+, \omega_q^+) \right\}_{q=1}^{N^+}. \quad (96)$$

For an integrand  $f$ , the phase-wise integral approximation is

$$\int_{\mathcal{D}^\pm} f(\mathbf{x}) d\mathbf{x} \approx \sum_{q=1}^{N^\pm} \omega_q^\pm f(\mathbf{x}_q^\pm). \quad (97)$$

Quadrature weights carry volume units for  $d = 3$  and area units for  $d = 2$ . On physical element faces, the  $d = 2$  weights represent surface-measure weights under Cartesian face coordinates.

### 5.2.1. Quadrature construction

Fig. 2 illustrates the *dimension-reduction* quadrature procedure. The construction is described first for  $d = 2$  and then for  $d = 3$ . In both cases, the quadrature is constructed by first generating quasiquadrature nodes on a transverse projection domain and then lifting these nodes along the height direction. The transverse nodes are used to define height lines; they are not, by themselves, the final volume quadrature points.

For a Cartesian integration box  $\mathcal{D}$ , let  $\mathbf{x}_\mathcal{D}$  denote its center. For a parameterized interface, let  $\mathbf{x}_{\Gamma, \mathcal{D}}$  be a closest point on the interface patch under consideration to  $\mathbf{x}_\mathcal{D}$ , and let  $\mathbf{n}_\mathcal{D}$  be the oriented unit normal at  $\mathbf{x}_{\Gamma, \mathcal{D}}$ . In practice, the closest point is approximated by the sample point on the current interface patch that is nearest to  $\mathbf{x}_\mathcal{D}$ . For an implicit interface, the height direction is selected using  $\nabla \phi_h(\mathbf{x}_\mathcal{D})$ .

*Two-dimensional case.* Let

$$\mathcal{D} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \subset \mathbb{R}^2.$$

The current face  $\mathcal{D}$  may contain multiple interface constraints. For an implicit interface representation, these constraints are described by multiple level-set functions  $\phi_h^{(c)}$ ,  $c = 1, \dots, N_\mathcal{D}$ . For a parameterized representation, they may be described by multiple parameterized interface curves. The construction below accounts for all constraints on  $\mathcal{D}$ .

1. Select the height direction. If multiple interface constraints are present on  $\mathcal{D}$ , select any one of them to determine a candidate height direction. For a parameterized interface, choose

$$n = \arg \max_{1 \leq j \leq 2} \left| \mathbf{n}_\mathcal{D}^{(c_0)} \cdot \mathbf{e}_j \right|,$$

where  $\mathbf{n}_\mathcal{D}^{(c_0)}$  is the oriented unit normal of the selected constraint  $c_0$ . For an implicit interface, choose

$$n = \arg \max_{1 \leq j \leq 2} \left| \partial_{x_j} \phi_h^{(c_0)}(\mathbf{x}_\mathcal{D}) \right|,$$

where  $\phi_h^{(c_0)}$  is the level-set function of the selected constraint and  $\mathbf{x}_\mathcal{D}$  is the center of  $\mathcal{D}$ . Here  $\mathbf{e}_j$  is the  $j$ -th Cartesian unit vector.

The candidate height direction must satisfy the admissibility conditions in Sec. 5.2.2 for every interface constraint on  $\mathcal{D}$ , not only for the selected constraint. Without loss of generality, reorder the coordinates so that  $n = 2$ , i.e.,  $y$  is the height coordinate and  $x$  is the transverse coordinate. Thus  $\mathbf{e}_n = \mathbf{e}_y$  and  $\perp n = x$ . The lower and upper height boundaries are

$$\mathcal{D}_{\perp n, \min} = [x_{\min}, x_{\max}] \times \{y_{\min}\}, \quad \mathcal{D}_{\perp n, \max} = [x_{\min}, x_{\max}] \times \{y_{\max}\}.$$

We use the lower boundary as the projection reference; the upper boundary could instead be used if the construction were applied consistently. If the candidate height direction is inadmissible for each constraint, subdivide  $\mathcal{D}$  and apply the procedure in Sec. 5.2.2.

- Determine all interface traces on the lower and upper height boundaries, considering every constraint on  $\mathcal{D}$ . For an implicit interface, define, for  $c = 1, \dots, N_{\mathcal{D}}$ ,

$$\phi_{\min}^{(c)}(x) = \phi_h^{(c)}(x, y_{\min}), \quad \phi_{\max}^{(c)}(x) = \phi_h^{(c)}(x, y_{\max}).$$

The lower- and upper-boundary traces are the roots of  $\phi_{\min}^{(c)}(x) = 0$  and  $\phi_{\max}^{(c)}(x) = 0$ , respectively, for all  $c$ . For a parameterized interface, find all intersections of every interface curve on  $\mathcal{D}$  with both the lower and upper boundary segments, subject to the corresponding interface-parameter bounds and segment bounds. Retain only intersections whose transverse coordinates lie in  $[x_{\min}, x_{\max}]$ .

- Project the upper-boundary traces onto the lower boundary along the height direction. This projection preserves the  $x$ -coordinate. Let  $P_{\perp n}$  be the set of distinct transverse coordinates of all lower-boundary traces and all projected upper-boundary traces, collected over every constraint. For an implicit interface,

$$P_{\perp n} = \bigcup_{c=1}^{N_{\mathcal{D}}} \left( \left\{ x \in (x_{\min}, x_{\max}) \mid \phi_{\min}^{(c)}(x) = 0 \right\} \cup \left\{ x \in (x_{\min}, x_{\max}) \mid \phi_{\max}^{(c)}(x) = 0 \right\} \right).$$

For parameterized interfaces, use the transverse coordinates of the intersections found in Step 2, including the projected upper-boundary intersections. Sort the distinct coordinates and add the endpoints:

$$\bar{x}_0 = x_{\min} < \bar{x}_1 < \dots < \bar{x}_M < \bar{x}_{M+1} = x_{\max}, \quad M = |P_{\perp n}|.$$

The traces of all constraints on the lower and upper boundaries jointly define this transverse subdivision.

- On each subinterval  $[\bar{x}_i, \bar{x}_{i+1}]$ , apply a  $q_{\perp}$ -point Gauss–Legendre rule. Let  $(x_{\text{ref},p}, \omega_{\text{ref},p})$  denote its reference nodes and weights on  $[-1, 1]$ . Map them to the subinterval:

$$\tilde{x}_{i,p} = \frac{\bar{x}_i + \bar{x}_{i+1}}{2} + \frac{\bar{x}_{i+1} - \bar{x}_i}{2} x_{\text{ref},p}, \quad \tilde{\omega}_{i,p}^{\perp} = \frac{\bar{x}_{i+1} - \bar{x}_i}{2} \omega_{\text{ref},p}.$$

Denote the resulting transverse points and weights, over all subintervals, by  $\{(\tilde{x}_k, \tilde{\omega}_k^{\perp})\}$ . These are transverse pseudo-quadrature points used to define height lines; they are not final volume quadrature points.

- For each transverse point  $\tilde{x}_k$ , find all interface intersections on the height line

$$\mathbf{x}_k(t) = (\tilde{x}_k, t), \quad t \in [y_{\min}, y_{\max}],$$

considering every constraint on  $\mathcal{D}$ . For an implicit interface, find the roots of  $\phi_h^{(c)}(\tilde{x}_k, t) = 0$  on this segment for all  $c = 1, \dots, N_{\mathcal{D}}$ . For a parameterized interface, find all intersections of every interface curve on  $\mathcal{D}$  with this height segment, subject to the corresponding interface-patch and segment bounds. Sort the distinct intersection heights and append the segment endpoints:

$$\bar{y}_{k,0} = y_{\min} < \bar{y}_{k,1} < \dots < \bar{y}_{k,L_k} < \bar{y}_{k,L_k+1} = y_{\max}.$$

If there are no interface intersections, this list consists only of  $y_{\min}$  and  $y_{\max}$ .

- Each interval  $[\bar{y}_{k,i}, \bar{y}_{k,i+1}]$  lies within one phase. Apply a  $q_n$ -point Gauss–Legendre rule to each interval. Let  $(y_{\text{ref},p}, \omega_{\text{ref},p})$  denote its reference nodes and weights on  $[-1, 1]$ . Map them to the interval:

$$y_{k,i,p} = \frac{\bar{y}_{k,i} + \bar{y}_{k,i+1}}{2} + \frac{\bar{y}_{k,i+1} - \bar{y}_{k,i}}{2} y_{\text{ref},p}, \quad (98)$$

$$\mathbf{x}_{k,i,p} = (\tilde{x}_k, y_{k,i,p}), \quad (99)$$

$$\omega_{k,i,p} = \tilde{\omega}_k^{\perp} \frac{\bar{y}_{k,i+1} - \bar{y}_{k,i}}{2} \omega_{\text{ref},p}. \quad (100)$$

The points  $\mathbf{x}_{k,i,p}$  and weights  $\omega_{k,i,p}$  are the final points and weights for the quadrature on  $\mathcal{D}$ . Assign the points from each interval to the phase containing its midpoint. For parameterized interfaces, use the prescribed phase ordering; for implicit interfaces, use the signs of all level-set functions  $\phi_h^{(c)}$  to identify the phase.

- Collect the final points and weights over all transverse points and height intervals. Partition the resulting quadrature into  $Q^-$  and  $Q^+$  according to the phase assignments in Step 6.

*Three-dimensional case.* Let

$$\mathcal{D} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}] \subset \mathbb{R}^3.$$

The current box  $\mathcal{D}$  may contain multiple interface constraints, described either by level-set functions  $\phi_h^{(c)}$ ,  $c = 1, \dots, N_D$ , or by multiple parameterized interface patches. The construction below accounts for all constraints on  $\mathcal{D}$ .

1. Select the height direction using the rule in Step 1 of the two-dimensional construction, with  $1 \leq j \leq 3$ . If multiple constraints are present, use one constraint to select a candidate direction, then check its admissibility for every constraint. If necessary, apply the fallback procedure in Sec. 5.2.2. Without loss of generality, assume  $n = 3$ , so  $z$  is the height coordinate and  $(x, y)$  are the transverse coordinates. The lower and upper height faces are

$$\mathcal{D}_{\perp n, \min} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times \{z_{\min}\}, \quad \mathcal{D}_{\perp n, \max} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times \{z_{\max}\}.$$

Use the lower face as the projection reference.

2. Determine all interface traces on the lower and upper height faces, considering every constraint on  $\mathcal{D}$ . For an implicit interface, these are the zero sets

$$\phi_{\min}^{(c)}(x, y) = \phi_h^{(c)}(x, y, z_{\min}), \quad \phi_{\max}^{(c)}(x, y) = \phi_h^{(c)}(x, y, z_{\max}), \quad c = 1, \dots, N_D.$$

For a parameterized interface, find all intersections of every interface patch with each height face, subject to the face and interface-patch bounds.

3. Project the upper-face traces onto the lower face along the height direction. This projection preserves  $(x, y)$ . Let  $\mathcal{P}_{\perp n}$  be the union of the lower-face traces and the projected upper-face traces, collected over all constraints, together with the boundary of the transverse rectangle

$$\mathcal{D}_{\perp n} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}].$$

These curves partition  $\mathcal{D}_{\perp n}$  into subregions. As in two dimensions, the projected traces identify where the number or ordering of interface intersections along a height line may change.

4. On  $\mathcal{D}_{\perp n}$ , construct a two-dimensional quadrature rule using the prescribed quadrature procedure under two-dimensional case. Denote the resulting nodes and weights by

$$\{(\tilde{\mathbf{y}}_k, \tilde{\omega}_k^\perp)\}, \quad \tilde{\mathbf{y}}_k = (\tilde{x}_k, \tilde{y}_k) \in \mathcal{D}_{\perp n}.$$

These are transverse pseudo-quadrature points used to define height lines; they are not final volume quadrature points.

5. For each transverse pseudo-quadrature point  $\tilde{\mathbf{y}}_k = (\tilde{x}_k, \tilde{y}_k)$ , apply Steps 5–7 of the two-dimensional height-line construction and phase-classification procedure to

$$\mathbf{x}_k(t) = (\tilde{x}_k, \tilde{y}_k, t), \quad t \in [z_{\min}, z_{\max}].$$

The resulting final volume points inherit the phase labels from that procedure, and their weights are the corresponding one-dimensional mapped weights multiplied by  $\tilde{\omega}_k^\perp$ . Collect these points and weights in  $Q^-$  and  $Q^+$  according to their inherited phase labels.

### 5.2.2. Height direction admissibility

The Saye-type construction accepts a Cartesian direction  $n$  as the height direction only when the interface is locally single-valued and the associated geometric distortion remains bounded. For an implicit interface, with

$$\mathbf{n} = \frac{\nabla \phi_h}{\|\nabla \phi_h\|},$$

the key admissibility condition is

$$\frac{\|\nabla\phi_h\|}{|\partial_{x_j}\phi_h|} = \frac{1}{|\mathbf{n} \cdot \mathbf{e}_j|} \leq \tau_{\text{adm}}, \quad \text{equivalently} \quad |\mathbf{n} \cdot \mathbf{e}_j| \geq \frac{1}{\tau_{\text{adm}}}, \quad (101)$$

where  $\tau_{\text{adm}}$  is a prescribed admissibility threshold. Following [54], this work takes  $\tau_{\text{adm}} = 4$ . In addition,  $\partial_{x_j}\phi_h$  must preserve a nonzero sign throughout the box. For a parameterized interface, the corresponding condition is imposed directly on  $\mathbf{n}_D \cdot \mathbf{e}_j$ .

When multiple interface restrictions are present in the current box, the same Cartesian direction must be used for all of them. Thus, a candidate direction is admissible only if each restriction satisfies the corresponding condition above, including the required sign preservation throughout the box.

These conditions ensure that the interface remains single-valued in the chosen height direction and that the induced Jacobian factor does not vary excessively. If no direction satisfies these requirements, the box is subdivided before quadrature generation. A sub-box lying entirely within either phase is then integrated using a standard tensor-product Gauss rule, whereas a sub-box still cut by the interface is treated by the aforementioned dimension-reduction framework. Generating quadrature points without this subdivision may produce an invalid height representation and therefore fail to attain the desired integration accuracy. If the prescribed maximum subdivision depth is reached without an admissible direction, a low-order fallback rule is used.

**Remark 4.** Algorithm 2 and Algorithm 3 in Appendix B detail the implementation workflow of the above-mentioned steps. The resulting quadrature sets are stored separately and employed independently during assembly to compute phase-wise volume and face contributions. Algorithm 4 in Appendix C summarizes the complete assembly procedure for local matrices and vectors associated with an interface element  $K$ .

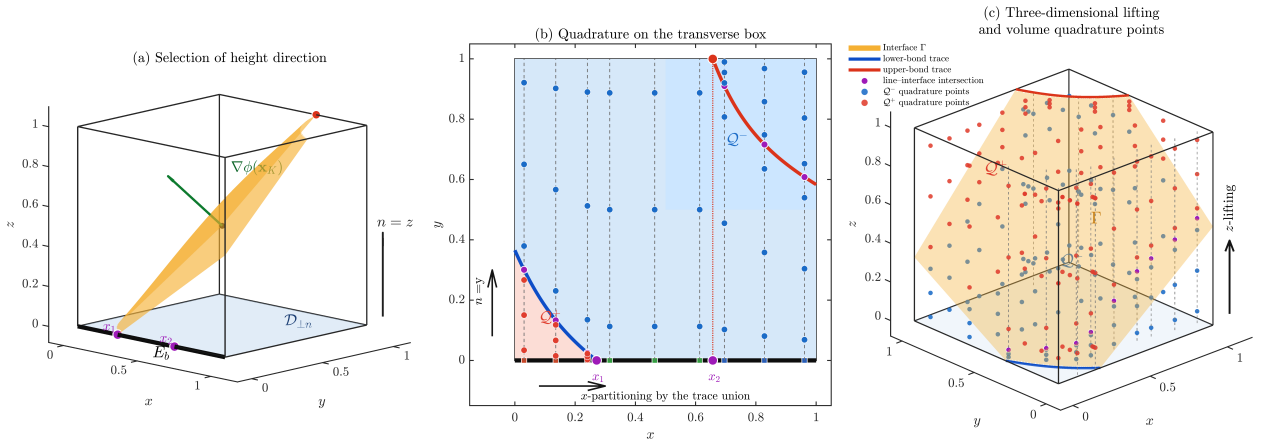


Figure 2: Schematic illustration of the Saye-type dimension-reduction quadrature, applied simultaneously to  $\Omega^-$  and  $\Omega^+$ . (a) A Cartesian coordinate  $z$  is selected as the height direction. (b) The lower-dimensional quadrature is constructed on the transverse box. For the current box, the height direction is  $n = y$ . The lower-dimensional interface consists of the union of the interface traces on the two faces  $y = y_{\min}$  and  $y = y_{\max}$ . Each trace is intersected with its corresponding horizontal edge; the resulting  $x$ -coordinates,  $x_1$  and  $x_2$ , are merged and used to partition the one-dimensional transverse interval. The resulting transverse quadrature points are pseudo-quadrature points, not final volume quadrature points. (c) For each transverse pseudo-quadrature point, the two-dimensional height-line construction is applied to generate the final volume quadrature points and weights.

## 6. Numerical examples

This section presents numerical experiments to validate the accuracy and demonstrate the geometric flexibility of the proposed high-order GA-IFE method. Unless stated otherwise, diffusion coefficients are set to  $\beta^- = 1$  and  $\beta^+ = 10$ , and relative  $L^2$  and  $H^1$  errors are reported. Computations employ uniform Cartesian  $N^3$  meshes with tensor-product Legendre basis functions of polynomial degree  $m = 1, \dots, 4$ . For  $m = 1, \dots, 3$ , we set the geometric

reconstruction order  $m_g = m$ . For  $m = 4$ , the errors and convergence rates continue to benefit from increasing the geometric reconstruction order; accordingly, we adopt  $m_g = m + 2$ . The expected optimal convergence rates are

$$\|u - u_h\|_{L^2(\Omega)} = \mathcal{O}(h^{m+1}), |u - u_h|_{H^1(\Omega)} = \mathcal{O}(h^m).$$

Two distinct geometric workflows are tested. In Example 1, a globally parameterized surface is directly used for all geometric evaluations and quadrature. In Examples 2, 3 and 4, the interface geometry is supplied as sampled point-normal data and locally reconstructed using the Legendre-based procedure from Section 3. For benchmark problems defined by implicit functions, analytical level-set formulas are only used to generate interface samples, oriented normals, and reference plots; they are not adopted within the GA-IFE discretization. Interface samples are generated on sufficiently fine grids. For each interface element, the sampling neighborhood is slightly enlarged beyond the physical element to fully contain the associated fictitious element. For parameterized-surface tests, surface quantities and derivatives are evaluated directly from the analytical parameterization.

A key feature of the present GA-IFE scheme is that both subcell regions inside an interface element share identical element-wise degrees of freedom. Contributions from arbitrarily small cut subregions are assembled consistently with the rest of the element. Unlike many conventional unfitted approaches that introduce separate active-mesh degrees of freedom on each side of the interface, the proposed method requires neither cell aggregation for tiny cut subcells nor high-derivative ghost-penalty corrections.

### 6.1. Example 1:

This example validates the direct-parameterization geometric pathway. The computational domain is  $\Omega = [-1, 1]^3$ . As shown in Fig. 3, the interface is given directly by a globally parameterized paraboloid, and the computational interface is the portion of this parameterized surface lying inside  $\Omega$ :

$$\mathbf{S}(u, v) = \begin{pmatrix} u \\ v \\ \alpha((u - u_c)^2 + (v - v_c)^2) - \gamma \end{pmatrix}, \quad (u, v) \in \hat{\Gamma}, \quad (102)$$

where

$$\hat{\Gamma} = \{(u, v) \in \mathbb{R}^2 : (u - u_c)^2 + (v - v_c)^2 \leq R^2\}. \quad (103)$$

Here,  $(u_c, v_c)$  is the center of the parameter disk in the  $(u, v)$ -plane. Since  $x = u$  and  $y = v$ , it also coincides numerically with the center of the projection of the parameterized surface onto the physical  $xy$ -plane.

A manufactured solution is given by

$$u(\mathbf{x}) = \begin{cases} \frac{1}{\beta^-} P(\mathbf{x}), & \mathbf{x} \in \Omega^-, \\ \frac{1}{\beta^+} P(\mathbf{x}), & \mathbf{x} \in \Omega^+, \end{cases} \quad (104)$$

where  $P(\mathbf{x}) = \phi(\mathbf{x})^5$ , with  $\phi(\mathbf{x}) = z - \alpha((x - u_c)^2 + (y - v_c)^2) + \gamma$ .

For this example, we set  $R > 2$ ,  $\alpha = 0.5$ ,  $\gamma = 0.25$ , and the center offset  $(u_c, v_c) = (0.2, 0.3)$ . Source terms, interface jump conditions, and Dirichlet boundary data follow analytically from (104). Tests adopt mesh resolutions  $N = 6, 9, 12, 15$  and polynomial degrees  $m = 1, \dots, 4$ . Because the interface is supplied via exact global parameterization, this example isolates the performance of the parametric geometric pathway, free of point-normal reconstruction errors.

**Table 2**Relative  $L^2$  errors and convergence rates for the globally parameterized paraboloid interface.

| $N$ | $m = 1$   |        | $m = 2$   |        | $m = 3$   |        | $m = 4$   |        |
|-----|-----------|--------|-----------|--------|-----------|--------|-----------|--------|
|     | Error     | Rate   | Error     | Rate   | Error     | Rate   | Error     | Rate   |
| 6   | 5.5748e-1 | —      | 5.8563e-2 | —      | 5.1103e-3 | —      | 5.1897e-4 | —      |
| 9   | 2.6237e-1 | 1.8588 | 1.6964e-2 | 3.0558 | 7.5306e-4 | 4.7227 | 4.2884e-5 | 6.1494 |
| 12  | 1.5168e-1 | 1.9048 | 7.4064e-3 | 2.8808 | 2.4528e-4 | 3.8993 | 1.0726e-5 | 4.8172 |
| 15  | 9.8377e-2 | 1.9402 | 3.8626e-3 | 2.9174 | 1.0211e-4 | 3.9274 | 3.5253e-6 | 4.9865 |

**Table 3**Relative  $H^1$  errors and convergence rates for the globally parameterized paraboloid interface.

| $N$ | $m = 1$  |        | $m = 2$   |        | $m = 3$   |        | $m = 4$   |        |
|-----|----------|--------|-----------|--------|-----------|--------|-----------|--------|
|     | Error    | Rate   | Error     | Rate   | Error     | Rate   | Error     | Rate   |
| 6   | 5.1586e0 | —      | 8.4079e-1 | —      | 9.9823e-2 | —      | 9.1789e-3 | —      |
| 9   | 3.6158e0 | 0.8764 | 3.7933e-1 | 1.9630 | 2.6699e-2 | 3.2525 | 1.4194e-3 | 4.6038 |
| 12  | 2.7622e0 | 0.9360 | 2.1649e-1 | 1.9495 | 1.1395e-2 | 2.9598 | 4.6364e-4 | 3.8893 |
| 15  | 2.2287e0 | 0.9619 | 1.3944e-1 | 1.9715 | 5.8535e-3 | 2.9851 | 1.9092e-4 | 3.9762 |

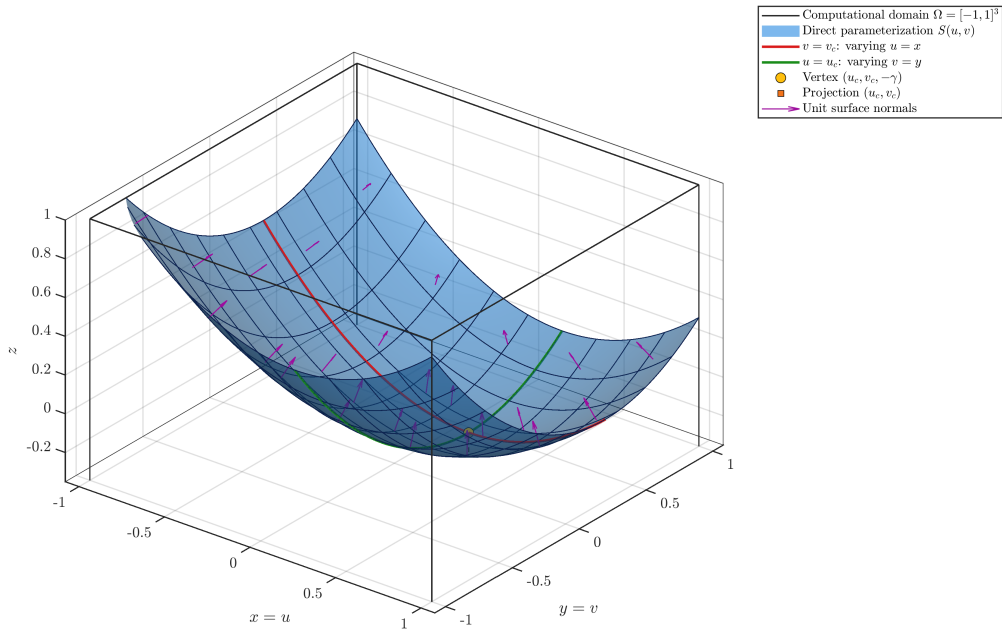


Figure 3: Globally parameterized paraboloid interface.

Tables 2 and 3 report the relative  $L^2$  and  $H^1$  errors for varying mesh resolutions and polynomial degrees. As this test case employs an exact global surface parameterization, geometric reconstruction errors are absent from the numerical results. We observe that the method in the direct parametric geometric pathway achieves optimal convergence rates: the  $L^2$  convergence order approaches  $m + 1$  and the  $H^1$  order is close to  $m$ .

## 6.2. Example 2:

The computational domain is the cube  $\Omega = [-1, 1]^3$ . The interface is a sphere centered at the origin with radius  $r_0 = 1/\sqrt{3}$ , see Figure 4. Mesh resolutions  $N = 6, 9, 12, 15$  are used. On the finest  $15 \times 15 \times 15$  mesh, each local sampling neighborhood contains at least 988 interface samples, providing sufficient data for local geometric reconstruction.

**Table 4**Relative  $L^2$  errors and local convergence rates for Example 2.

| $N$ | $m = 1$   |        | $m = 2$   |        | $m = 3$   |        | $m = 4$   |        |
|-----|-----------|--------|-----------|--------|-----------|--------|-----------|--------|
|     | Error     | Rate   | Error     | Rate   | Error     | Rate   | Error     | Rate   |
| 6   | 6.0781e-2 | —      | 3.7043e-3 | —      | 5.1791e-4 | —      | 1.9393e-5 | —      |
| 9   | 2.5201e-2 | 2.1713 | 1.0211e-3 | 3.1781 | 1.1783e-4 | 3.6523 | 2.7548e-6 | 4.8131 |
| 12  | 1.4464e-2 | 1.9299 | 4.4037e-4 | 2.9234 | 3.8680e-5 | 3.8722 | 6.4396e-7 | 5.0523 |
| 15  | 9.2620e-3 | 1.9976 | 2.2664e-4 | 2.9768 | 1.4040e-5 | 4.5417 | 1.8568e-7 | 5.5731 |

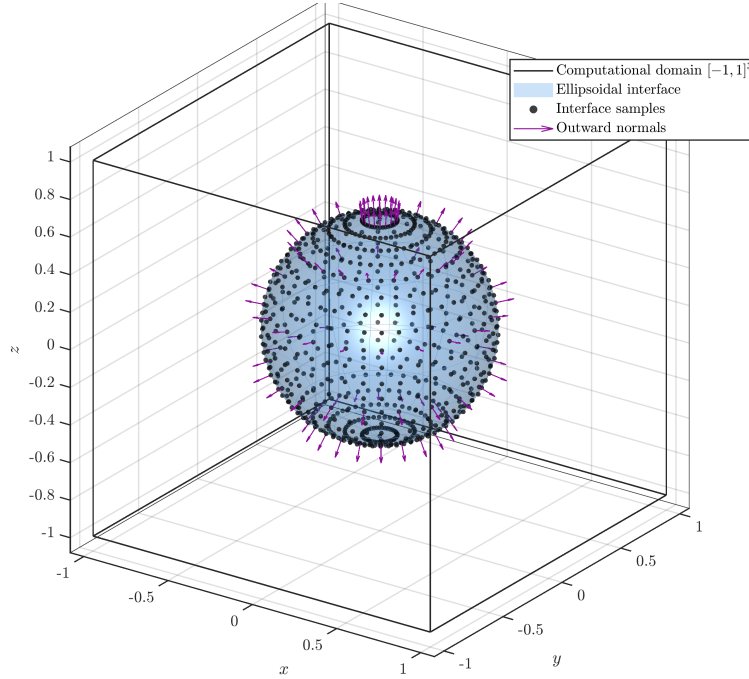


Figure 4: Spherical interface and the corresponding sampling points.

Source terms and boundary conditions are prescribed such that the exact solution reads

$$u(\mathbf{x}) = \begin{cases} \frac{1}{\beta^-} r^a, & \mathbf{x} \in \Omega^-, \\ \frac{1}{\beta^+} r^a + \left( \frac{1}{\beta^-} - \frac{1}{\beta^+} \right) r_0^a, & \mathbf{x} \in \Omega^+, \end{cases} \quad (105)$$

where  $r = \|\mathbf{x}\|$ ,  $r_0 = 1/\sqrt{3}$ , and  $a = 5$ . The corresponding source term is

$$f = -a(a+1)r^{a-2}.$$

Tables 4 and 5 list relative  $L^2$  and  $H^1$  errors for different mesh resolutions and polynomial degrees. Unlike Example 6.1 that uses an exact global parametric representation, this test case relies entirely on element-wise local Monge-patch reconstruction from discrete interface samples and oriented normals. Despite the approximation introduced by local geometric fitting, the numerical results attain near-optimal convergence rates, namely  $\mathcal{O}(h^{m+1})$  for the  $L^2$  norm and  $\mathcal{O}(h^m)$  for the  $H^1$  seminorm. This example demonstrates that the proposed local reconstruction strategy works reliably for smooth surfaces with constant curvature, and it provides a baseline reference for subsequent test cases involving more complicated interface geometries.

**Table 5**Relative  $H^1$  errors and local convergence rates for Example 2.

| $N$ | $m = 1$   |        | $m = 2$   |        | $m = 3$   |        | $m = 4$   |        |
|-----|-----------|--------|-----------|--------|-----------|--------|-----------|--------|
|     | Error     | Rate   | Error     | Rate   | Error     | Rate   | Error     | Rate   |
| 6   | 3.9963e-1 | —      | 5.0002e-2 | —      | 1.4200e-2 | —      | 4.6872e-3 | —      |
| 9   | 2.6959e-1 | 0.9708 | 2.0984e-2 | 2.1415 | 4.0512e-3 | 3.0932 | 1.0825e-3 | 3.6145 |
| 12  | 2.0591e-1 | 0.9367 | 1.2032e-2 | 1.9334 | 1.8426e-3 | 2.7381 | 3.7852e-4 | 3.6525 |
| 15  | 1.6571e-1 | 0.9733 | 7.7042e-3 | 1.9977 | 9.3153e-4 | 3.0554 | 1.5050e-4 | 4.1332 |

**6.3. Example 3:**

The computational domain remains  $\Omega = [-1, 1]^3$ . The interface is an ellipsoid centered at

$$\mathbf{c} = (c_x, c_y, c_z) = (-0.2, -0.2, 0),$$

with semi-axis lengths

$$a_1 = a_2 = \frac{4}{5\sqrt{3}}, \quad a_3 = \frac{1}{\sqrt{3}},$$

illustrated in Figure 5. Simulations are performed for  $N = 6, 9, 12, 15$ . On the finest mesh, each local sampling neighborhood contains at least 1722 samples.

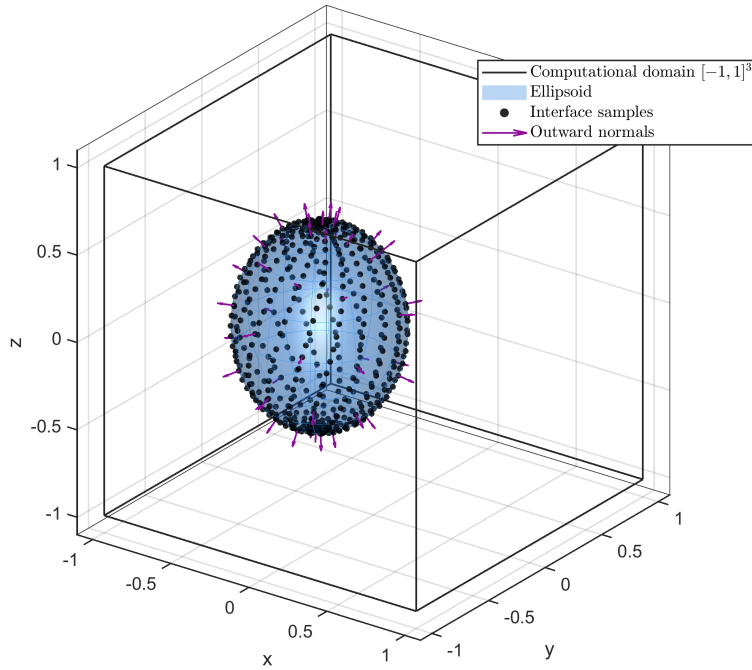


Figure 5: Ellipsoidal interface and the corresponding sampling points.

The manufactured exact solution is

$$u(\mathbf{x}) = \begin{cases} \frac{1}{\beta^-} s(\mathbf{x})^{q/2}, & \mathbf{x} \in \Omega^-, \\ \frac{1}{\beta^+} s(\mathbf{x})^{q/2} + \left( \frac{1}{\beta^-} - \frac{1}{\beta^+} \right), & \mathbf{x} \in \Omega^+, \end{cases} \quad (106)$$

where

$$s(\mathbf{x}) = \frac{(x - c_x)^2}{a_1^2} + \frac{(y - c_y)^2}{a_2^2} + \frac{(z - c_z)^2}{a_3^2}, \quad q = 5.$$

**Table 6**Relative  $L^2$  errors and local convergence rates for Example 3.

| $N$ | $m = 1$   |        | $m = 2$   |        | $m = 3$   |        | $m = 4$   |        |
|-----|-----------|--------|-----------|--------|-----------|--------|-----------|--------|
|     | Error     | Rate   | Error     | Rate   | Error     | Rate   | Error     | Rate   |
| 6   | 2.0484e0  | —      | 1.2771e-1 | —      | 1.2451e-2 | —      | 9.5652e-4 | —      |
| 9   | 9.5789e-1 | 1.8746 | 3.8757e-2 | 2.9410 | 2.9616e-3 | 3.5417 | 1.4096e-4 | 4.7225 |
| 12  | 5.4579e-1 | 1.9553 | 1.6300e-2 | 3.0107 | 9.1615e-4 | 4.0784 | 3.2031e-5 | 5.1507 |
| 15  | 3.4829e-1 | 2.0131 | 8.4015e-3 | 2.9701 | 3.8945e-4 | 3.8336 | 1.0870e-5 | 4.8431 |

**Table 7**Relative  $H^1$  errors and local convergence rates for Example 3.

| $N$ | $m = 1$  |        | $m = 2$   |        | $m = 3$   |        | $m = 4$   |        |
|-----|----------|--------|-----------|--------|-----------|--------|-----------|--------|
|     | Error    | Rate   | Error     | Rate   | Error     | Rate   | Error     | Rate   |
| 6   | 2.0554e1 | —      | 2.1977e0  | —      | 3.8146e-1 | —      | 6.5521e-2 | —      |
| 9   | 1.3988e1 | 0.9491 | 9.9369e-1 | 1.9576 | 9.4022e-2 | 3.4538 | 1.0234e-2 | 4.5791 |
| 12  | 1.0559e1 | 0.9777 | 5.5189e-1 | 2.0442 | 3.4472e-2 | 3.4876 | 2.9500e-3 | 4.3239 |
| 15  | 8.4815e0 | 0.9818 | 3.5351e-1 | 1.9962 | 1.6846e-2 | 3.2088 | 1.2112e-3 | 3.9893 |

Source terms and boundary data are derived analytically from (106).

Tables 6 and 7 summarize relative  $L^2$  and  $H^1$  errors for varying mesh resolutions and polynomial degrees. Compared with the spherical interface in Example 6.2, the ellipsoid exhibits spatially varying principal curvatures due to its unequal semi-axes, which provides a more demanding test for the local Legendre–Monge reconstruction. Moreover, the elongated geometry inevitably produces interface elements with extremely small cut subregions on the Cartesian mesh. Nevertheless, the numerical results maintain near-optimal convergence rates of  $\mathcal{O}(h^{m+1})$  in  $L^2$  and  $\mathcal{O}(h^m)$  in  $H^1$  across all tested polynomial degrees. This outcome demonstrates that the proposed GA-IFE scheme robustly handles arbitrarily small cut subregions without resorting to cell aggregation, ghost-penalty stabilization, or any other special treatment for ill-conditioned tiny subcells. The key enabler is that both subcell regions within an interface element share the same set of element-wise degrees of freedom, so the contribution of a vanishingly small subregion is assembled consistently with the rest of the element.

#### 6.4. Example 4

This example considers a portion of a Route-C Fourier-sphere interface embedded in  $\Omega = [-1, 1]^3$ , as shown in Fig. 6. This interface is selected from the full Fourier sphere. It is smooth and non-polynomial. Its representative interface samples together with the outward unit normals are shown in Fig. 6.

Define the center and the base radius

$$\mathbf{c} = (c_x, c_y, c_z) = (-0.2, -0.2, 0)^T, \quad R_0 = 0.9, \quad \varepsilon = 0.35,$$

and the five amplitudes

$$(a_1, a_2, a_3, a_4, a_5) = (1.0, 2.0, 1.5, 0.3, 1.0).$$

For  $\mathbf{x} = (x, y, z)^T$ , set

$$\rho = \|\mathbf{x} - \mathbf{c}\|_2, \quad \theta = \arccos\left(\frac{z - c_z}{\rho}\right), \quad \varphi = \text{atan2}(y - c_y, x - c_x).$$

Define the auxiliary function

$$G(\theta, \varphi) = a_1 \cos \theta + a_2 \sin \theta \cos \varphi + a_3 \sin \theta \sin \varphi + a_4 \sin^2 \theta \cos(2\varphi) + a_5 \sin^2 \theta \sin(2\varphi),$$

and the radius function

$$r(\theta, \varphi) = \frac{R_0}{\sqrt{1 + \varepsilon G(\theta, \varphi)}}.$$

The interface is the zero level-set of

$$\phi(\mathbf{x}) = \frac{\rho^2}{r(\theta, \varphi)^2} - 1. \quad (107)$$

Equivalently, the interface is the surface  $\rho = r(\theta, \varphi)$ .

The manufactured exact solution is given by

$$u(\mathbf{x}) = \begin{cases} \frac{s(\mathbf{x})^{p/2}}{\beta^-}, & \mathbf{x} \in \Omega^-, \\ \frac{s(\mathbf{x})^{p/2}}{\beta^+} + \left( \frac{1}{\beta^-} - \frac{1}{\beta^+} \right), & \mathbf{x} \in \Omega^+, \end{cases} \quad (108)$$

where

$$s(\mathbf{x}) = \frac{\rho^2}{r(\theta, \varphi)^2},$$

and  $p$  is a fixed parameter. In the simulations below we set  $p = 5$ . The interface is  $s = 1$ , i.e.  $\Omega^- = \{s < 1\}$  and  $\Omega^+ = \{s > 1\}$ . Since  $r(\theta, \varphi)$  contains the non-polynomial Fourier-type angular dependence, both the interface  $s = 1$  and the manufactured solution (108) are non-polynomial. Thus, this example tests the approximation properties of the GA-IFE method for a smooth but geometrically complex non-polynomial interface and a non-polynomial exact solution.

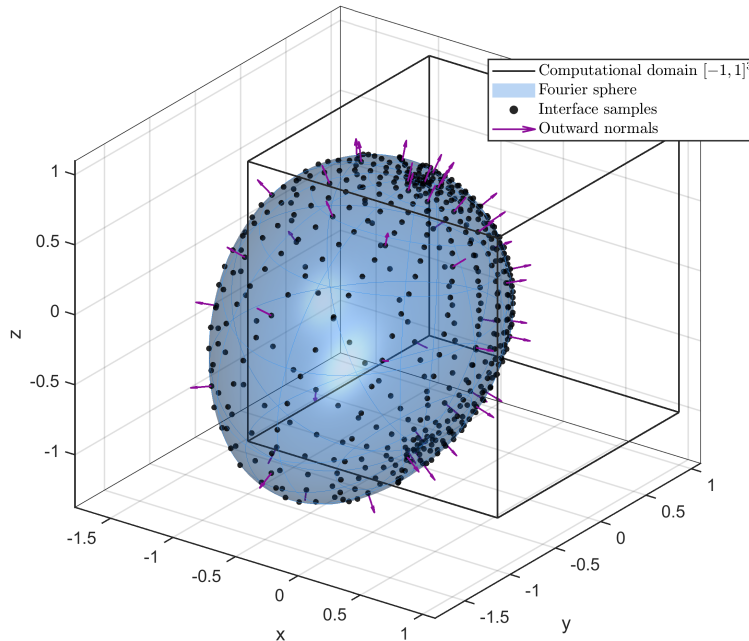


Figure 6: Route-C Fourier-sphere interface with representative interface samples and outward unit normals.

Source terms, interface jump data, and Dirichlet boundary conditions are constructed from the manufactured solution (108). Simulations use mesh resolutions  $N = 6, 9, 12, 15$ . Sampling is performed on a tensor-product grid in spherical coordinates: 240 points in  $\theta \in (0, \pi)$  and 480 points in  $\varphi \in [0, 2\pi)$ , together with the two poles. Duplicate points and samples near gradient singularities are discarded prior to reconstruction. Tables 8 and 9 collect the relative  $L^2$  and  $H^1$  errors. This test evaluates the GA-IFE approximation of a smooth, strongly undulated, non-polynomial interface and a non-polynomial exact solution.

**Table 8**Relative  $L^2$  errors and local convergence rates for the Route-C Fourier-sphere interface in Example 4.

| $N$ | $m = 1$   |        | $m = 2$   |        | $m = 3$   |        | $m = 4$   |        |
|-----|-----------|--------|-----------|--------|-----------|--------|-----------|--------|
|     | Error     | Rate   | Error     | Rate   | Error     | Rate   | Error     | Rate   |
| 6   | 6.4298e-1 | —      | 3.5737e-2 | —      | 3.0581e-3 | —      | 1.1343e-4 | —      |
| 9   | 2.7024e-1 | 2.1378 | 1.0245e-2 | 3.0815 | 5.6842e-4 | 4.1500 | 1.6853e-5 | 4.7024 |
| 12  | 1.4871e-1 | 2.0763 | 4.3284e-3 | 2.9948 | 1.6909e-4 | 4.2145 | 4.1928e-6 | 4.8358 |
| 15  | 9.2391e-2 | 2.1330 | 2.1360e-3 | 3.1651 | 6.4747e-5 | 4.3019 | 1.4102e-6 | 4.8831 |

**Table 9**Relative  $H^1$  errors and local convergence rates for the Route-C Fourier-sphere interface in Example 4.

| $N$ | $m = 1$  |        | $m = 2$   |        | $m = 3$   |        | $m = 4$   |        |
|-----|----------|--------|-----------|--------|-----------|--------|-----------|--------|
|     | Error    | Rate   | Error     | Rate   | Error     | Rate   | Error     | Rate   |
| 6   | 4.7651e0 | —      | 5.4725e-1 | —      | 5.9841e-2 | —      | 3.1847e-3 | —      |
| 9   | 3.1458e0 | 1.0241 | 2.3896e-1 | 2.0436 | 1.6042e-2 | 3.2468 | 7.0540e-4 | 3.7176 |
| 12  | 2.3338e0 | 1.0379 | 1.3426e-1 | 2.0042 | 6.3858e-3 | 3.2019 | 2.3008e-4 | 3.8944 |
| 15  | 1.8547e0 | 1.0295 | 8.4101e-2 | 2.0961 | 3.0908e-3 | 3.2519 | 9.8181e-5 | 3.8164 |

Numerical results demonstrate near-optimal convergence rates across all tested polynomial orders  $m = 1, 2, 3, 4$ , with  $L^2$  convergence approaching  $m + 1$  and  $H^1$  convergence near  $m$ . These results demonstrate that the GA-IFE method preserves near-optimal convergence for both a smooth, strongly undulated non-polynomial interface and a non-polynomial manufactured solution. Thus, this example complements the previous validations on spherical and ellipsoidal geometries by confirming the method's effectiveness for more complex non-polynomial interfaces.

## 7. Conclusion

This paper presents a high-order geometry-aligned immersed finite element (GA-IFE) method for 3D elliptic interface problems with curved interfaces. The proposed framework supports two complementary geometric pathways for interface description. For an interface surface with a sufficiently regular global parameterization, interface-related geometric quantities can be directly evaluated from the analytical parametric representation. For general interfaces lacking such global parameterization, local interface geometry is reconstructed via high-order Legendre-Monge patches, yielding local signed implicit level-set representations. The IFE basis functions are built upon the same Legendre polynomial space used for geometric reconstruction, following an isogeometric principle that unifies geometric approximation and physical field discretization within a single mathematical setting. In this sense, the construction shares a fundamental idea with IGA, namely, the use of exact geometry in finite element computations.

This consistent formulation ensures that geometric and discretization errors reside in the same Sobolev norm, with convergence behavior governed primarily by the polynomial approximation order. Within each interface element, a local Monge orthonormal frame parameterizes the interface neighborhood, and all required geometric derivatives are obtained analytically. An algebraic procedure evaluates high-order normal derivatives of the Laplace operator through the chain rule and metric-tensor variations, circumventing cumbersome explicit differential expansions. Numerical integration over cut-cells is realized by the dimension-reduction Saye-type quadrature scheme, which relies only on 1D root-finding and Gaussian quadrature to lower computational complexity.

Numerical examples are presented to validate the proposed GA-IFE method with a suite of test cases whose interfaces include a globally parameterized paraboloid, sphere, ellipsoid, and the non-polynomial Route-C Fourier sphere. For interfaces that are either analytically parameterized or described by polynomial or non-polynomial functions, the method robustly demonstrates the optimal  $L^2$  and  $H^1$  convergence, even for elements containing arbitrarily small cut subregions and without resorting to cell-aggregation or ghost-penalty stabilization techniques.

## A. Construction and Evaluation of IFE Basis Functions

---

### Algorithm 1 Construction and Evaluation of IFE Basis Functions

---

**Require:** Background grid  $\mathcal{T}_h$ , interface representation, diffusion coefficients  $\beta^\pm$ , evaluation point  $\mathbf{x}_p$

**Ensure:** Basis value  $\lambda_i^s(\mathbf{x}_p)$  and physical gradient  $\nabla_{\mathbf{x}} \lambda_i^s(\mathbf{x}_p)$

**Stage 1: Interface preparation and interface-element detection**

- 1: **if** globally parameterized surface **then**
- 2:     Detect interface elements  $\mathcal{T}_h^i$  via analytical surface intersection tests.
- 3: **else**
- 4:     Generate oriented interface samples, perform grid binning and collect  $\mathcal{T}_h^i$ .
- 5: **end if**

**Stage 2: Local geometric preprocessing per interface element**

- 6: **for** each  $K \in \mathcal{T}_h^i$  **do**
- 7:     **if** globally parameterized surface **then**
- 8:         Evaluate surface geometry and geometric derivatives analytically (Eqs. 15–26).
- 9:     **else**
- 10:         Build local orthonormal Monge frame  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{n}_0\}$  (Eq. 27).
- 11:         Collect and filter neighborhood samples; solve Legendre-Hermite least-squares fitting for height function  $\eta_h^K$  (Eq. 33).
- 12:         Construct local map  $\mathbf{S}_h$ , implicit representation  $\phi_h^K$ , and associated geometric quantities (Eqs. 35–49).
- 13:     **end if**
- 14:     Map element vertices to tubular coordinates and calculate elementwise scaling factors (Eqs. 52–58).
- 15: **end for**

**Stage 3: Assembly of interface-condition matrices**

- 16: **for** each  $K \in \mathcal{T}_h^i$  **do**
- 17:     Compute Jacobian-related normal derivatives, auxiliary curvature quantities and Laplacian derivatives (Eqs. 72–77).
- 18:     Evaluate high-order Laplacian normal derivatives and assemble interface blocks  $\mathbf{B}_0, \mathbf{B}_1, \tilde{\mathbf{A}}^{(n)}$  via  $L^2$  projection (Eq. 79 and Eqs. 68–70).
- 19:     Form global matrix  $\tilde{\mathbf{A}}$  and jump-scaling matrix  $\mathbf{J}_\beta$  (Eqs. 66, 67).
- 20:     Solve for coefficient matrices  $\mathbf{C}^\pm$  (Eq. 12) and perform orthogonalization [6].
- 21: **end for**

**Stage 4: Pointwise evaluation at  $\mathbf{x}_p$**

- 22: Determine phase sign  $s$ .
  - 23: Map  $\mathbf{x}_p$  to local coordinates  $\mathbf{u}_p$  and compute inverse Jacobian  $\mathbf{M}$ .
  - 24: Evaluate basis:  $\lambda_i^s(\mathbf{x}_p) = \sum_{k=1}^{N_b} (\mathbf{C}^s)_{k,i} R_k(\mathbf{u}_p)$ .
  - 25: Evaluate physical gradient:  $\nabla_{\mathbf{x}} \lambda_i^s(\mathbf{x}_p) = \mathbf{M}^\top \sum_{k=1}^{N_b} (\mathbf{C}^s)_{k,i} \nabla_{\mathbf{u}} R_k(\mathbf{u}_p)$ .
  - 26: **return**  $\lambda_i^s(\mathbf{x}_p), \nabla_{\mathbf{x}} \lambda_i^s(\mathbf{x}_p)$
-

## B. Saye-type quadrature for both phases

---

### Algorithm 2 Two-dimensional Saye-type quadrature

---

**Require:** Axis-aligned box  $D = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$ , all interface constraints on  $D$ , Gauss orders  $q_{\perp}$  and  $q_n$ , and maximum subdivision depth  $L$

**Ensure:** Phase quadrature rules  $Q^-$  and  $Q^+$

- 1: Initialize  $Q^-$  and  $Q^+$
  - 2: **if**  $D$  is not cut by any interface constraint **then**
  - 3:     Construct a tensor-product Gauss–Legendre rule on  $D$  and assign it to the corresponding phase
  - 4:     **return** ( $Q^-$ ,  $Q^+$ )
  - 5: **end if**
  - 6: Select a candidate height direction using one interface constraint on  $D$ , as described in Step 1 of the two-dimensional construction
  - 7: Check the candidate direction for admissibility with respect to every interface constraint on  $D$
  - 8: **if** the candidate direction is inadmissible **then**
  - 9:     **if** the subdivision depth is less than  $L$  **then**
  - 10:         Subdivide  $D$ , apply Algorithm 2 to each child box, and merge the resulting phase quadrature rules
  - 11:     **else**
  - 12:         Apply the prescribed low-order fallback rule and assign its points and weights to the corresponding phases
  - 13:     **end if**
  - 14:     **return** ( $Q^-$ ,  $Q^+$ )
  - 15: **end if**
  - 16: Without loss of generality, set  $n = 2$ , so that  $y$  is the height coordinate and  $x$  is the transverse coordinate
  - 17: Find all traces of all interface constraints on  $y = y_{\min}$  and  $y = y_{\max}$ , as in Step 2 of the two-dimensional construction
  - 18: Project the upper-boundary traces onto  $y = y_{\min}$  and collect their transverse coordinates together with those of the lower-boundary traces
  - 19: Sort the distinct transverse coordinates and add  $x_{\min}$  and  $x_{\max}$  to form the subdivision points  $\bar{x}_0, \dots, \bar{x}_{M+1}$
  - 20: **for all** transverse subintervals  $[\bar{x}_i, \bar{x}_{i+1}]$  **do**
  - 21:     Map the  $q_{\perp}$ -point Gauss–Legendre rule to the subinterval using Step 4 of the two-dimensional construction
  - 22:     Store the resulting points and weights as transverse pseudo-quadrature points  $(\tilde{x}_k, \tilde{\omega}_k^{\perp})$
  - 23: **end for**
  - 24: **for all** transverse pseudo-quadrature points  $(\tilde{x}_k, \tilde{\omega}_k^{\perp})$  **do**
  - 25:     Find all distinct intersections of all interface constraints with the height line
 
$$\mathbf{x}_k(t) = (\tilde{x}_k, t), \quad t \in [y_{\min}, y_{\max}]$$
  - 26:     Sort the intersection heights and append  $y_{\min}$  and  $y_{\max}$  to form the height intervals
  - 27:     **for all** resulting height intervals **do**
  - 28:         Classify the interval using an interior point and the prescribed phase-classification rule
  - 29:         Map the  $q_n$ -point Gauss–Legendre rule to the interval
  - 30:         Multiply each mapped height weight by  $\tilde{\omega}_k^{\perp}$
  - 31:         Append the resulting final points and weights to  $Q^-$  or  $Q^+$  according to the interval's phase
  - 32:     **end for**
  - 33: **end for**
  - 34: **return** ( $Q^-$ ,  $Q^+$ )
-

---

**Algorithm 3** Three-dimensional Saye-type quadrature
 

---

**Require:** Axis-aligned box  $\mathcal{D} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}]$ , all interface constraints on  $\mathcal{D}$ , Gauss orders  $q_{\perp}$  and  $q_n$ , and maximum subdivision depth  $L$

**Ensure:** Phase quadrature rules  $\mathcal{Q}^-$  and  $\mathcal{Q}^+$

- 1: Initialize  $\mathcal{Q}^-$  and  $\mathcal{Q}^+$
- 2: **if**  $\mathcal{D}$  is not cut by any interface constraint **then**
- 3:     Construct a tensor-product Gauss–Legendre rule on  $\mathcal{D}$  and assign it to the corresponding phase
- 4:     **return** ( $\mathcal{Q}^-, \mathcal{Q}^+$ )
- 5: **end if**
- 6: Select a candidate height direction using one interface constraint on  $\mathcal{D}$ , as described in Step 1 of the two-dimensional construction
- 7: Check the candidate direction for admissibility with respect to every interface constraint on  $\mathcal{D}$
- 8: **if** the candidate direction is inadmissible **then**
- 9:     **if** the subdivision depth is less than  $L$  **then**
- 10:         Subdivide  $\mathcal{D}$ , apply Algorithm 3 to each child box, and merge the resulting phase quadrature rules
- 11:     **else**
- 12:         Apply the prescribed low-order fallback rule and assign its points and weights to the corresponding phases
- 13:     **end if**
- 14:     **return** ( $\mathcal{Q}^-, \mathcal{Q}^+$ )
- 15: **end if**
- 16: Without loss of generality, set  $n = 3$ , so that  $z$  is the height coordinate and  $(x, y)$  are the transverse coordinates
- 17: Find all interface traces of all constraints on  $z = z_{\min}$  and  $z = z_{\max}$
- 18: Project the upper-face traces onto  $z = z_{\min}$  and combine them with the lower-face traces to partition the transverse rectangle

$$\mathcal{D}_{\perp n} = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$$

- 19: Apply Algorithm 2 to  $\mathcal{D}_{\perp n}$  using the transverse traces of all interface constraints. Retain the resulting two-dimensional quadrature nodes and weights as transverse pseudo-quadrature data

$$\{(\tilde{\mathbf{y}}_k, \tilde{\omega}_k^{\perp})\}_{k=1}^{N_{\perp}}, \quad \tilde{\mathbf{y}}_k = (\tilde{x}_k, \tilde{y}_k).$$

- 20: **for all** transverse pseudo-quadrature points  $(\tilde{\mathbf{y}}_k, \tilde{\omega}_k^{\perp})$  **do**
  - 21:     Form the height line  $\mathbf{x}_k(t) = (\tilde{x}_k, \tilde{y}_k, t)$ ,  $t \in [z_{\min}, z_{\max}]$ , and find and sort all its distinct intersections with the interface constraints, including the box endpoints.
  - 22:     Split the height line at these intersections. On each resulting interval, determine the phase from an interior point, map the one-dimensional reference Gauss rule to the interval, and multiply its weights by  $\tilde{\omega}_k^{\perp}$ .
  - 23:     Append the resulting points and weights to  $\mathcal{Q}^-$  or  $\mathcal{Q}^+$ , according to the phase.
  - 24: **end for**
  - 25: **return** ( $\mathcal{Q}^-, \mathcal{Q}^+$ )
-

## C. Assembly of local matrices and vectors for interface element

---

### Algorithm 4 Assembly of local matrices and vectors for interface element

---

**Require:** Interface element  $K$ , penalty parameter  $\sigma_e$ , boundary data  $g_D$ , source term  $f$

**Ensure:** Local matrices  $\mathbf{K}_{\text{vol}}^K$ ,  $\mathbf{K}_{\text{flux}}^{\partial K}$ ,  $\mathbf{K}_{\text{pen}}^{\partial K}$  and local load vectors  $\mathbf{f}^K$ ,  $\mathbf{f}^{\partial K}$

```

1: Initialize  $\mathbf{K}_{\text{vol}}^K = \mathbf{0}$ ,  $\mathbf{K}_{\text{flux}}^{\partial K} = \mathbf{0}$ ,  $\mathbf{K}_{\text{pen}}^{\partial K} = \mathbf{0}$ ,  $\mathbf{f}^K = \mathbf{0}$ ,  $\mathbf{f}^{\partial K} = \mathbf{0}$ 
2: Generate volume quadrature sets  $Q^-, Q^+$  using Algorithm ??
▷ 1. Volume Integration

3: for each  $s \in \{-, +\}$  and  $(\mathbf{x}_p, w_p) \in Q^s$  do
4:   Evaluate  $\lambda_i^s(\mathbf{x}_p)$  and  $\nabla \lambda_i^s(\mathbf{x}_p)$  for all  $i$  using Algorithm 1
5:    $(\mathbf{K}_{\text{vol}}^K)_{ij} \leftarrow (\mathbf{K}_{\text{vol}}^K)_{ij} + w_p \beta^s \nabla \lambda_i^s(\mathbf{x}_p) \cdot \nabla \lambda_j^s(\mathbf{x}_p)$ 
6:    $(\mathbf{f}^K)_i \leftarrow (\mathbf{f}^K)_i + w_p f(\mathbf{x}_p) \lambda_i^s(\mathbf{x}_p)$ 
7: end for

▷ 2. Face Integration (Interior and Boundary)

8: for each face  $e \subset \partial K$  do
9:   Initialize face-specific matrices:  $\mathbf{K}_{\text{flux}}^e = \mathbf{0}$ ,  $\mathbf{K}_{\text{pen}}^e = \mathbf{0}$ ,  $\mathbf{f}^e = \mathbf{0}$ 
10:  Determine face type:  $e \in \mathcal{E}_h^{ii}$  (interior) or  $e \in \mathcal{E}_h^{bi}$  (boundary)
11:  Generate face quadrature sets  $Q_e^s = \{\mathbf{s}_p, w_p\}$  for active subdomains  $s$  (using Algorithm ?? if cut)
12:  for active  $s \in \{-, +\}$  and  $(\mathbf{s}_p, w_p) \in Q_e^s$  do
13:    Evaluate  $\lambda_i^s(\mathbf{s}_p)$  and  $\nabla \lambda_i^s(\mathbf{s}_p)$  for all  $i$  using Algorithm 1
14:    if  $e \in \mathcal{E}_h^{ii}$  then
▷ Interior Face (Cut or Uncut)
15:      Accumulate  $(\mathbf{K}_{\text{flux}}^e)_{ij}$  and  $(\mathbf{K}_{\text{pen}}^e)_{ij}$  using Eqs. (88)–(89) (or uncut equivalents)
16:    else
▷ Boundary Face (Cut or Uncut)
17:       $(\mathbf{K}_{\text{flux}}^e)_{ij} \leftarrow (\mathbf{K}_{\text{flux}}^e)_{ij} - w_p \left( (\beta^s \nabla \lambda_i^s \cdot \mathbf{n}_e) \lambda_j^s + (\beta^s \nabla \lambda_j^s \cdot \mathbf{n}_e) \lambda_i^s \right)$ 
18:       $(\mathbf{K}_{\text{pen}}^e)_{ij} \leftarrow (\mathbf{K}_{\text{pen}}^e)_{ij} + w_p \frac{\sigma_e}{h_e} \lambda_i^s \lambda_j^s$ 
19:       $(\mathbf{f}^e)_i \leftarrow (\mathbf{f}^e)_i + w_p \left( -\beta^s \nabla \lambda_i^s \cdot \mathbf{n}_e + \frac{\sigma_e}{h_e} \lambda_i^s \right) g_D(\mathbf{s}_p)$ 
20:    end if
21:  end for
22:   $\mathbf{K}_{\text{flux}}^{\partial K} \leftarrow \mathbf{K}_{\text{flux}}^{\partial K} + \mathbf{K}_{\text{flux}}^e$ ,  $\mathbf{K}_{\text{pen}}^{\partial K} \leftarrow \mathbf{K}_{\text{pen}}^{\partial K} + \mathbf{K}_{\text{pen}}^e$ ,  $\mathbf{f}^{\partial K} \leftarrow \mathbf{f}^{\partial K} + \mathbf{f}^e$ 
23: end for
24: return  $\mathbf{K}_{\text{vol}}^K$ ,  $\mathbf{K}_{\text{flux}}^{\partial K}$ ,  $\mathbf{K}_{\text{pen}}^{\partial K}$ ,  $\mathbf{f}^K$ ,  $\mathbf{f}^{\partial K}$ 

```

---

## References

- [1] Abbena, E., Salamon, S., Gray, A., 2006. Modern Differential Geometry of Curves and Surfaces with Mathematica. 3rd ed., Chapman and Hall/CRC.
- [2] Adjerd, S., Ben-Romdhane, M., Lin, T., 2014. Higher degree immersed finite element methods for second-order elliptic interface problems. International Journal of Numerical Analysis and Modeling 11, 541–566.
- [3] Adjerd, S., Ben-Romdhane, M., Lin, T., 2018. Higher degree immersed finite element spaces constructed according to the actual interface. Computers & Mathematics with Applications 75, 1868–1881.
- [4] Adjerd, S., Guo, R., Lin, T., 2017. High degree immersed finite element spaces by a least squares method. International Journal of Numerical Analysis and Modeling 14, 604–625.
- [5] Adjerd, S., Lin, T., Meghaichi, H., 2024. A high order geometry conforming immersed finite element for elliptic interface problems. Computer Methods in Applied Mechanics and Engineering 420, 116703. doi:10.1016/j.cma.2023.116703.
- [6] Adjerd, S., Lin, T., Meghaichi, H., 2025a. Construction of basis functions for the geometry conforming immersed finite element method. URL: <https://arxiv.org/abs/2510.12018>, arXiv:2510.12018.
- [7] Adjerd, S., Lin, T., Meghaichi, H., 2025b. The Frenet immersed finite element method for elliptic interface problems: An error analysis. Computer Methods in Applied Mechanics and Engineering 438, 117829. doi:10.1016/j.cma.2025.117829.
- [8] Adjerd, S., Lin, T., Meghaichi, H., 2026a. Construction of basis functions for the geometry conforming immersed finite element method. Numerical Algorithms doi:<https://doi.org/10.1007/s11075-026-02424-6>.

- [9] Adjerid, S., Lin, T., Meghaichi, H., 2026b. A MATLAB implementation of the geometry-conforming immersed finite element space. Zenodo. doi:10.5281/ZENODO.17314849. version 1.0.
- [10] BAZILEVS, Y., BEIRÃO DA VEIGA, L., COTTRELL, J.A., HUGHES, T.J.R., SANGALLI, G., 2006. Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences* 16, 1031–1090. doi:10.1142/S0218202506001455.
- [11] Belytschko, T., Black, T., 1999. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 45, 601–620.
- [12] Burman, E., Claus, S., Hansbo, P., Larson, M.G., Massing, A., 2015. Cutfem: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering* 104, 472–501.
- [13] Cao, H., Cao, Y., Chu, Y., He, X.M., Lin, T., 2018. A Huygens immersed-finite-element particle-in-cell method for modeling plasma-surface interactions with moving interface. *Communications in Nonlinear Science and Numerical Simulation* 59, 132–148.
- [14] Chorin, A.J., 1968. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation* 22, 745–762.
- [15] Ciarlet, P.G., 1978. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam.
- [16] Clough, R., Tocher, J., 1966. Finite element stiffness matrices for analysis of plates in bending, in: *Proceedings of the Conference on Matrix Methods in Structural Mechanics*, Wright-Patterson Air Force Base. pp. 515–545.
- [17] Depew, D., Han, D., Wang, J., He, X.M., Lin, T., 2014. Immersed-finite-element particle-in-cell simulations of lunar surface charging, in: *Proceedings of the 13th Spacecraft Charging Technology Conference*, Pasadena, California. p. 199.
- [18] Ewing, R.E., 1983. *The Mathematics of Reservoir Simulation*. Society for Industrial and Applied Mathematics.
- [19] Farnell, C.C., 2007. *Plasma Flow Field Measurements Downstream of a Hollow Cathode*. Ph.D. thesis. Colorado State University. Fort Collins, CO, USA.
- [20] Fries, T., 2018. Higher-order conformal decomposition fem (cdfem). *Computer Methods in Applied Mechanics and Engineering* 328, 75–98.
- [21] Fries, T., Schöllhammer, D., 2017. Higher-order meshing of implicit geometries, part ii: approximations on manifolds. *Computer Methods in Applied Mechanics and Engineering* 326, 270–297.
- [22] Fries, T.P., Omerović, S., 2015. Higher-order accurate integration of implicit geometries. *International Journal for Numerical Methods in Engineering* 106, 323–371.
- [23] Gersborg-Hansen, A., Bendsøe, M.P., Sigmund, O., 2006. Topology optimization of heat conduction problems using the finite volume method. *Structural and Multidisciplinary Optimization* 31, 251–259.
- [24] Guo, R., Lin, T., 2019. A higher degree immersed finite element method based on a Cauchy extension. *SIAM Journal on Numerical Analysis* 57, 1545–1573.
- [25] Hale, N., Townsend, A., 2013. Fast and accurate computation of Gauss–Legendre and Gauss–Jacobi quadrature nodes and weights. *SIAM Journal on Scientific Computing* 35, A652–A674.
- [26] Han, D., He, X.M., Lund, D., Zhang, X., 2021. Pife-pic: parallel immersed-finite-element particle-in-cell for 3-d kinetic simulations of plasma-material interactions. *SIAM Journal on Scientific Computing* 43, C235–C257.
- [27] Han, D., Wang, J., He, X.M., 2016a. A non-homogeneous immersed-finite-element particle-in-cell method for modeling dielectric surface charging in plasmas. *IEEE Transactions on Plasma Science* 44, 1326–1332.
- [28] Han, D., Wang, J., He, X.M., 2018. Immersed-finite-element particle-in-cell simulations of plasma charging at lunar terminator. *Journal of Spacecraft and Rockets* 55, 1490–1497.
- [29] Han, D., Wang, P., He, X.M., Lin, T., Wang, J., 2016b. A 3d immersed finite element method with non-homogeneous interface flux jump for applications in particle-in-cell simulations of plasma-lunar surface interactions. *Journal of Computational Physics* 321, 965–980.
- [30] Han, Y., Xia, G., Kang, H., Lu, C., Chen, C., Saetchnikov, V., 2025. Three-dimensional direct-implicit particle-in-cell model using trilinear anisotropic immersed-finite-element for plasma propulsion. *AIAA Journal* 63, 323–335. doi:10.2514/1.J064305.
- [31] Han, Y., Xia, G., Lu, C., He, X.M., 2023. Trilinear immersed finite element method for 3d anisotropic interface problems with applications to plasma thrusters. *AIAA Journal* 61.
- [32] Hansbo, A., Hansbo, P., 2002. An unfitted finite element method, based on Nitsche’s method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering* 191, 5537–5552.
- [33] He, X., Lin, T., Lin, Y., 2008. Approximation capability of a bilinear immersed finite element space. *Numerical Methods for Partial Differential Equations* 24, 1265–1300.
- [34] Hesthaven, J.S., Warburton, T., 2008. *Nodal Discontinuous Galerkin Methods*. volume 54 of *Texts in Applied Mathematics*. Springer, New York, NY.
- [35] Hughes, T., Cottrell, J., Bazilevs, Y., 2005. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194, 4135–4195. URL: <https://www.sciencedirect.com/science/article/pii/S0045782504005171>, doi:<https://doi.org/10.1016/j.cma.2004.10.008>.
- [36] Kafafy, R., 2005. Immersed finite element Particle-In-Cell simulations of ion propulsion. Ph.D. thesis. Virginia Polytechnic Institute and State University.
- [37] Kafafy, R., Wang, J., Lin, T., 2005. A hybrid-grid immersed-finite-element particle-in-cell simulation model of ion optics plasma dynamics. *Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Applications and Algorithms* 12, 1–16.
- [38] Koornwinder, T., 1975. Two-variable analogues of the classical orthogonal polynomials, in: *Theory and Application of Special Functions*. Academic Press, pp. 435–495.
- [39] Kopriva, D.A., 2009. *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Scientific Computation, Springer Netherlands, Dordrecht.
- [40] Legrain, G., Moës, N., 2018. Adaptive anisotropic integration scheme for high-order fictitious domain methods: application to thin structures. *International Journal for Numerical Methods in Engineering* 114, 882–904.

- [41] Lehrenfeld, C., 2016. High order unfitted finite element methods on level set domains using isoparametric mappings. *Computer Methods in Applied Mechanics and Engineering* 300, 716–733.
- [42] LeVeque, R.J., Li, Z., 1994. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis* 31, 1019–1044.
- [43] Li, J., Xia, G., Han, Y., Wang, Z., Lu, C., He, X., 2026. Partially penalized anisotropic trilinear ife-pic methods for dc plasma transport problems. *AIAA Journal* 64, 280–302. doi:10.2514/1.J065244.
- [44] Li, Z., Lin, T., Lin, Y., Rogers, C., 2004. An immersed finite element space and its approximation capability. *Numerical Methods for Partial Differential Equations* 20, 338–367.
- [45] Li, Z., Lin, T., Wu, X., 2003. New cartesian grid methods for interface problems using finite element formulation. *Numerische Mathematik* 96, 61–98.
- [46] Lin, T., Lin, Y., Rogers, R.C., Ryan, L.M., 2001. A rectangular immersed finite element method for interface problems, in: *Scientific Computing and Applications*. Nova Science Publishers, Commack, NY, USA. volume 7, pp. 107–114.
- [47] Lu, C., Wan, J., Cao, Y., He, X.M., 2020. A fully decoupled iterative method with three-dimensional anisotropic immersed finite elements for kaufman-type discharge problems. *Computer Methods in Applied Mechanics and Engineering* 372, 113345.
- [48] Lund, D., He, X.M., Han, D., 2023. Kinetic particle simulations of plasma charging at lunar craters under severe conditions. *Journal of Spacecraft and Rockets* 60, 1176–1187.
- [49] Lund, D., He, X.M., Zhang, X., Han, D., 2022. Weak scaling of the parallel immersed finite element particle-in-cell (pife-pic) framework with lunar plasma charging simulations. *Computational Particle Mechanics*.
- [50] Lyness, J., Jespersen, D., 1975. Moderate degree symmetric quadrature rules for the triangle. *IMA Journal of Applied Mathematics* 15, 19–32.
- [51] Meghaichi, H., 2024. Higher order immersed finite element methods for interface problems. Ph.D. thesis. Virginia Polytechnic Institute and State University.
- [52] Moës, N., Dolbow, J., Belytschko, T., 1999. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering* 46, 131–150.
- [53] Proriot, J., 1957. Sur une famille de polynômes à deux variables orthogonaux dans un triangle. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences* 245, 2459–2461.
- [54] Saye, R.I., 2015. High-order quadrature methods for implicitly defined surfaces and volumes in hyperrectangles. *SIAM Journal on Scientific Computing* 37, A993–A1019.
- [55] Stanford, J., Fries, T., 2019a. A higher-order conformal decomposition finite element method for plane b-rep geometries. *Computers & Structures* 214, 15–27.
- [56] Stanford, J.W., Fries, T.P., 2019b. Higher-order accurate meshing of nonsmooth implicitly defined surfaces and intersection curves. *Computational Mathematics and Mathematical Physics* 59, 2093–2107.
- [57] Stroud, A.H., 1971. *Approximate Calculation of Multiple Integrals*. Prentice-Hall.
- [58] Wang, J., He, X.M., Cao, Y., 2007. Modeling spacecraft charging and charged dust particle interactions on lunar surface, in: *Proceedings of the 10th Spacecraft Charging Technology Conference*, Biarritz, France.
- [59] Wang, J., He, X.M., Cao, Y., 2008. Modeling electrostatic levitation of dusts on lunar surface. *IEEE Transactions on Plasma Science* 36, 2459–2466.
- [60] Wang, Z., Xia, G., Han, Y., Lu, C., Zhang, L., Xu, G., 2025. An innovative galerkin scheme based on anisotropic trilinear immersed finite elements for the magnetized plasma diffusion problem with plasma sheath interface. *Computers & Mathematics with Applications* 192, 155–171.
- [61] Xia, G., Li, J., Lu, C., Geng, H., Saetchnikov, V.A., Li, J., Wang, Y., 2023. Ion thruster accelerator grid erosion mechanism under extreme conditions of cylindrical erosion and chamfer erosion. *Acta Astronautica* Available online 22 March 2023.
- [62] Zlámal, M., 1973. Curved elements in the finite element method. I. *SIAM Journal on Numerical Analysis* 10, 229–240.