Computational Mechanics
https://doi.org/10.1007/s00466-024-02506-z

ORIGINAL PAPER

Interpolation-based immersogeometric analysis methods for multi-material and multi-physics problems

Jennifer E. Fromm[1] · Nils Wunsch[2] · Kurt Maute[2] · John A. Evans[2] · Jiun-Shyan Chen[3]

Received: 24 February 2024 / Accepted: 20 May 2024

Abstract

Immersed boundary methods are high-order accurate computational tools used to model geometrically complex problems in computational mechanics. While traditional finite element methods require the construction of high-quality boundary-fitted meshes, immersed boundary methods instead embed the computational domain in a structured background grid. Interpolation-based immersed boundary methods augment existing finite element software to non-invasively implement immersed boundary capabilities through extraction. Extraction interpolates the structured background basis as a linear combination of Lagrange polynomials defined on a foreground mesh, creating an interpolated basis that can be easily integrated by existing methods. This work extends the interpolation-based immersed isogeometric method to multi-material and multi-physics problems. Beginning from level-set descriptions of domain geometries, Heaviside enrichment is implemented to accommodate discontinuities in state variable fields across material interfaces. Adaptive refinement with truncated hierarchically refined B-splines (THB-splines) is used to both improve interface geometry representations and to resolve large solution gradients near interfaces. Multi-physics problems typically involve coupled fields where each field has unique discretization requirements. This work presents a novel discretization method for coupled problems through the application of extraction, using a single foreground mesh for all fields. Numerical examples illustrate optimal convergence rates for this method in both 2D and 3D, for partial differential equations representing heat conduction, linear elasticity, and a coupled thermo-mechanical problem. The utility of this method is demonstrated through image-based analysis of a composite sample, where in addition to circumventing typical meshing difficulties, this method reduces the required degrees of freedom when compared to classical boundary-fitted finite element methods.

Keywords Immersed finite element method · Lagrange extraction · XIGA · Multi-physics problems · Multi-material problems

1 Introduction

Computational modeling has become an integral part of engineering of all types, and developments in manufacturing and design have increased the complexity of the models required. Multi-material problems are now ubiquitous, appearing in the design and analysis of composites [56], advanced additive manufacturing products [50], and multi-phase system analysis [72]. These new design spaces present challenges to modeling methods, primarily in the discretization of intricate geometries and domain interfaces, and the unique discontinuities in solutions that result from material interactions. Finite element methods (FEMs) are among the most widely used computational tools for structural analysis. Classical FEM relies on sufficiently refined boundary-fitted meshes to discretize both the geometric domain of a material and its solution's function space. The accuracy of these methods is closely tied to mesh quality [12], especially when high-order

✉ Jiun-Shyan Chen
js-chen@ucsd.edu

Jennifer E. Fromm
jefromm@ucsd.edu

Nils Wunsch
nils.wunsch@colorado.edu

Kurt Maute
kurt.maute@colorado.edu

John A. Evans
john.a.evans@colorado.edu

[1] Mechanical and Aerospace Engineering, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, USA

[2] Aerospace Engineering, University of Colorado, Boulder, 3775 Discovery Drive, Boulder, CO 80303, USA

[3] Structural Engineering, University of California San Diego, 9500 Gilman Dr, La Jolla, CA 92093, USA

Published online: 17 June 2024

methods are employed [22]. Thus, considerable care must be taken in mesh construction before analysis can be performed [41]. As the complexity of multi-material problems increases, generating these high-quality conforming meshes required by FEM becomes increasingly challenging, especially in three dimensions. Even for single material problems, mesh generation and refinement can consume up to 80% of the design-through-analysis time of engineers [10]. More complicated PDEs involving multiple state variables also present unique modeling challenges. In such multi-physics problems, the various fields may have distinct discretization needs and the fields must be coupled.

Immersed boundary methods circumvent conforming mesh generation by embedding the geometric problem domain into a background grid constructed on a geometrically simple domain. Initially proposed to track fluid–structure interfaces in [55], similar classes of immersed methods were likewise developed by the solid mechanics community to accommodate discontinuities in solution fields without remeshing to create boundary-fitted meshes. The partition of unity method (PUM), introduced in [4], leverages the concept of enriching solution functions using *a priori* knowledge of the location of discontinuities. This was combined with classical FEM in [65] and [64] to introduce the generalized finite element method (GFEM). A similar enriched method, characterized by adaptive enrichment schemes, known as the “eXtended" finite element method (XFEM) [8, 9, 48], was also introduced to model crack propagation and other discontinuous problems. Immersed boundary methods have also been extended to include high-order methods. The finite cell method [54, 57] utilizes p refinement of Lagrange polynomial basis functions to increase convergence rates within an immersed framework. In the field of meshfree methods, the concepts of immersed or embedded methods have been used to model heterogeneous materials [60] and to enhance solution accuracy and stability near material interfaces in fluid–structure interaction problems [32].

Isogeometric analysis (IGA) directly utilizes the geometric representation used in most computer-aided design (CAD) software in analysis [33] and was initially proposed to address the issue of generating high-quality boundary-fitted meshes. The B-spline basis functions used in IGA offer additional advantages over classical FEM including improved geometric representation, higher levels of continuity, and improved per-degree-of-freedom accuracy compared to more common nodal finite element basis functions [34, 35]. The combinations of IGA and immersed boundary methods are often called immersogeometric methods [39], and have been validated for use with hierarchically refined T-spline CAD models [58] and for ‘trimmed’ CAD geometries [14]. The XFEM class of enriched methods was applied to IGA to create the eXtended isogeometric method (XIGA) in [52], exploiting level-set geometric descriptions and sophisticated integration algorithms to solve complicated PDEs involving multiple materials. XIGA was extended to use truncated hierarchically refined B-splines (THB-splines) for multi-physics problems in [61].

While offering elegant solutions to the problem of modeling complex geometries, existing immersed boundary software is currently limited to custom research codes. This is in part due to the complexity of generating custom quadrature rules on each cut background element [19]. These custom quadrature rules require implementations that depart significantly from the highly optimized integration algorithms used in classical FEM codes. For example, an immersed boundary functionality called MultiMesh [36] was developed for the popular open-source FEM software FEniCS [1], however it proved too difficult to maintain and was not ported to the more recent FEniCSx [6].

Some classes of immersed boundary methods are specially formulated to reduce the difficulty of integrating over cut cells, including approximate domain methods and specifically the shifted boundary method [42, 43]. The shifted boundary method maps the boundaries of a computational domain to a mesh conforming surrogate domain and has been extended to high-order methods in [3]. Similar to the shifted boundary method, interpolation-based immersed methods like the one presented in this work provide the benefits of immersed methods without requiring invasive implementation of custom quadrature methods.

This work utilizes approximate extraction to retrofit classical FEM codes to perform immersed analysis. Extraction was introduced to represent B-spline basis functions with Bézier polynomials for implementation of IGA [11], and was generalized to Lagrange extraction in [59]. With Lagrange extraction, each spline in a B-spline function space is represented as a linear combination of Lagrange polynomials belonging to an interpolatory function basis $\{N_i\}$, which satisfies the Kronecker delta property $\delta_{ij} = N_i(\boldsymbol{x}_j)$, where $\boldsymbol{x}_j$ are the nodal coordinates. Because of this interpolatory property, Lagrange extraction-based methods are also called interpolation-based methods. IGA and other partition of unity methods are challenging to implement as shape functions are not the same from element to element [47] making interpolation-based methods attractive. Lagrange extraction has been utilized to implement IGA within existing finite element software in [37, 38], which uses the software FEniCS, and in [68], which uses the software Code_Aster.

In [26], approximate extraction was applied to immersed boundary methods. In interpolation-based immersed boundary methods, a domain is embedded in a structured background mesh, which is used to define a background basis. A boundary-fitted foreground mesh is quickly generated by decomposing the cut elements of the background mesh. The foreground mesh, which does not need to adhere to the usual mesh quality metrics [41], is used to define a boundary-fitted Lagrange polynomial foreground basis. The background basis is interpolated with the foreground basis, and the interpolated basis can then be used to solve PDEs with Galerkin-type methods. As the interpolated basis is represented as a linear combination of Lagrange polynomials defined upon a boundary-fitted mesh, the integration can be performed with classical FE software, as illustrated with FEniCS in [26]. This work also employed B-spline background bases, combining immersed and isogeometric methods for an immersogeometric method. The interpolation-based immersogeometric method differs from previous extraction-based IGA methods in its use of approximate instead of exact extraction. With approximate extraction, the interpolated background basis is not everywhere equivalent to the actual background basis, resulting in additional interpolation error. The additional interpolation error is demonstrated to be bounded by the optimal method error, thus the method still yields optimal convergence rates. Easing exact interpolation constraints to permit approximate extraction allows for easier and more efficient implementation.

Previous work using interpolation-based immersed-boundary methods used a single uniform background basis for single material and single physics problems [26]. This work expands the applications of this method to multi-material and multi-physics problems with the following contributions:

- Discontinuous state variable fields are approximated by background bases with Heaviside enrichment applied at material interfaces.
- Material interfaces are described by level sets, and hierarchical refinement allows for local refinement.
- Local foreground refinement is applied independently of background refinement, improving geometric approximation of material interfaces without increasing the number of system degrees of freedom.
- Separate background bases are used to approximate different fields in multi-physics applications, which are interpolated using a single foreground basis, allowing for easy coupling.

An interpolation-based immersed boundary framework utilizing these developments is implemented in the next generation open-source software library FEniCSx [6], with code available at [25]. Multi-material heat conduction and linear elasticity are modeled to demonstrate the convergence rates of the proposed workflow. A coupled thermo-mechanical problem illustrates the combined multi-material and multi-physics capabilities of the interpolation-based immersed boundary framework.

The outline of this paper is as follows: Sect. 2 provides an overview of hierarchical B-splines and the generation of the THB-splines used for local refinement. Section 3 describes this method's treatment of the geometric description of material interfaces and the Heaviside enrichment of solution spaces with discontinuities at material interfaces. Section 4 details the novel interpolation-based immersed boundary method's application to multi-material and multi-physics problems and its implementation workflow within existing FEM codes. Section 5 provides numerical results validating and expanding upon this method, and finally Sect. 6 draws conclusions and suggests future work that can be done with this method.

2 Hierarchical B-splines

Immersogeometric analysis combines aspects of two classes of methods, immersed methods and isogeometric methods. As in isogeometric methods, this work employs splines to represent state variable fields. A review of multivariate B-spline functions spaces is provided in "Appendix A". The following is a discussion of hierarchical refinement of B-spline functions spaces and an overview of the THB-spline spaces used in the application of interpolation-based immersogeometric methods to multi-material problems.

2.1 Hierarchically refined B-spline function spaces

Hierarchiacally refined B-splines (HB-splines) are constructed using nested sequences of spline spaces created by repeated knot insertion. Following the algorithms presented in [27], an HB-spline basis begins with the construction of a sequence of r tensor-product spline spaces $\mathcal{V}^l$,

$$\mathcal{V}^0 \subset \mathcal{V}^1 \subset \cdots \subset \mathcal{V}^{r-1}, \tag{1}$$

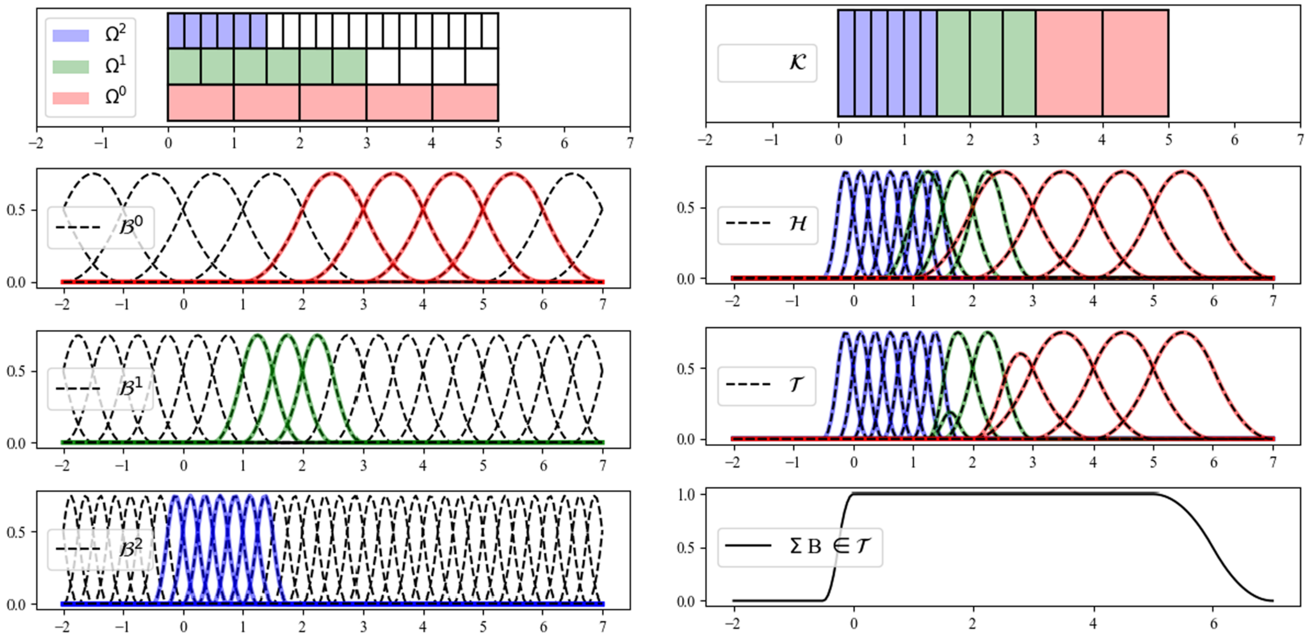
each of which has an accompanying B-spline basis $\mathcal{B}^l = \{B_i^l\}$, and tensor-product Cartesian mesh $\mathcal{K}^l$, where elements are denoted by K. A sequence of subdomains Ω^l are chosen, such that Ω^{l+1} is a subregion of Ω^l,

$$\Omega^0 \supseteq \Omega^1 \supseteq \cdots \supseteq \Omega^{r-1}, \tag{2}$$

and each Ω^l can be discretized with mesh elements $K \in \mathcal{K}^l$. Here r is the depth of refinement.

The HB-spline basis $\mathcal{H} := \mathcal{H}^{r-1}$ is constructed recursively by the algorithm:

$$\begin{cases} \mathcal{H}^0 & := \mathcal{B}^0, \\ \mathcal{H}^{l+1} & := \{B \in \mathcal{H}^l \mid \operatorname{supp}(B) \not\subset \Omega^{l+1}\} \cup \\ & \quad\ \{B \in \mathcal{B}^{l+1} \mid \operatorname{supp}(B) \subset \Omega^{l+1}\}, \\ & l \in \{0, \ldots, r-2\}. \end{cases} \tag{3}$$



(a) A 1D illustration of a sequence of meshed subregions Ω^l and quadratic B-spline bases $\mathcal{B}^l$.

(b) The hierarchically refined mesh $\mathcal{K}$, the HB-spline basis $\mathcal{H}$, the THB-spline basis $\mathcal{T}$, and the verification of the partition of unity. The colors of the elements of $\mathcal{K}$ correspond to the refinement level.

Fig. 1 Local refinement is applied through truncated hierarchically-refined B-splines (THB)

In essence, each subsequent level's basis $\mathcal{H}^{l+1}$ is formed from the union of the set of basis functions from the previous level whose support is not in the new level's subdomain Ω^{l+1}, and the set of functions from the new basis $\mathcal{B}^{l+1}$ whose support is within Ω^{l+1}. This is illustrated with a 1D mesh in Fig. 1.

The hierarchically refined basis $\mathcal{H}$ is associated with a hierarchically refined mesh $\mathcal{K}$, defined as

$$\mathcal{K} := \bigcup_{l=0}^{r-1} \{K \in \mathcal{K}^l \mid K \in \Omega^l \text{ and } K \notin \Omega^{l+1}\}. \tag{4}$$

2.2 Enforcing the partition of unity property through truncation

While a useful tool for applying adaptive refinement to IGA, HB-spline bases violate the partition of unity (PU) property. To regain the this property, the hierarchically refined bases are truncated as in [27] and [28]. In addition to forming a partition of unity, truncation reduces the size of some basis functions' supports, thereby reducing the bandwidth of the resulting system of equations when compared to a non-truncated HB-spline basis.

A given multivariate basis function $B^l \in \mathcal{B}^l$ can be represented as a linear combination of the more refined functions of level $\mathcal{B}^{l+1}$:

$$B^l = \sum_{B^{l+1} \in \mathcal{B}^{l+1}} c^{B^{l+1}}\left(B^l\right) B^{l+1}, \tag{5}$$

where $c^{B^{l+1}}\left(B^l\right)$ are coefficients relating the coarse basis function B^l to the finer function B^{l+1}.

The truncation of B^l removes the contributions from $B^{l+1} \in \mathcal{B}^{l+1}$ with support contained within Ω^{l+1}, such that

$$\text{trun}^{l+1}(B^l) = B^l - \sum_{\substack{B^{l+1} \in \mathcal{B}^{l+1},\\ \text{supp}(B^{l+1}) \subseteq \Omega^{l+1}}} c^{B^{l+1}}\left(B^l\right) B^{l+1}. \tag{6}$$

Using a similar algorithm to that given in Eq. (3), the truncated basis $\mathcal{T} := \mathcal{T}^{r-1}$ can be constructed by

$$\begin{cases} \mathcal{T}^0 & := \mathcal{B}^0, \\ \mathcal{T}^{l+1} & := \{\text{trun}^{l+1}(B) \mid B \in \mathcal{T}^l,\ \text{supp}(B) \not\subset \Omega_{l+1}\} \\ & \quad \cup \{B \in \mathcal{B}^{l+1} \mid \text{supp}(B) \subset \Omega_{l+1}\}, \\ & l \in \{0, \ldots, r-2\}, \end{cases} \tag{7}$$

as illustrated in Fig. 1b. This truncated basis forms a partition of unity, as is proven in Theorem 10 of [28].

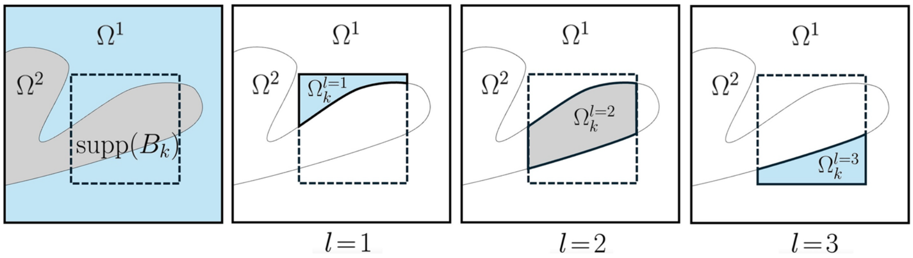


Fig. 2 The function-wise enrichment strategy considers the material connectivity within a basis functions support. The function B_k spans two material subdomains, denoted Ω^1 and Ω^2, forming three disconnected material subregions $\Omega_k^{l=1}$, $\Omega_k^{l=2}$, and $\Omega_k^{l=3}$

3 Immersed material interfaces

Workflows to solve multi-material PDEs require functionalities to both describe the geometry of material interfaces and to represent the associated discontinuities in the state variable fields. In this work, level set functions (LSFs) are utilized to implicitly describe the geometry of material interfaces, and a generalized Heaviside enrichment strategy in conjunction with a set of interface terms is employed to represent the required discontinuities at material interfaces.

3.1 Representing interface geometry through level set functions

The level set method, developed in [53], has been used to describe interfaces in the extended finite element method (XFEM) [8, 48] and extended isogeometric analysis (XIGA) [52, 61]. Following these works, the domain geometry is implicitly represented using LSFs $\phi_i(\boldsymbol{x})$. An iso-level ϕ_t of the LSF describes the interface $\Gamma_\pm$ between two subdomains Ω_+ and Ω_- such that

$$\begin{aligned}
&\phi(\boldsymbol{x}) < \phi_t, \boldsymbol{x} \in \Omega_+,\\
&\phi(\boldsymbol{x}) > \phi_t, \boldsymbol{x} \in \Omega_-,\\
&\phi(\boldsymbol{x}) = \phi_t, \boldsymbol{x} \in \Gamma_\pm.
\end{aligned} \tag{8}$$

With n LSFs, this method can represent up to 2^n subdomains. Materials are then associated with these subdomains using a multi-phase level set model as in [69], where phases are identified by phase indices $\mathcal{P}$. Phase indices $\mathcal{P}$ are assigned with characteristic functions f_i,

$$f_i(\boldsymbol{x}) = \begin{cases} 0, & \phi_i(\boldsymbol{x}) < \phi_t, \\ 1, & \phi_i(\boldsymbol{x}) \geq \phi_t, \end{cases} \tag{9}$$

such that

$$\mathcal{P}(\boldsymbol{x}) = \sum_{j=1}^{n} 2^{j-1} f_j(\boldsymbol{x}). \tag{10}$$

Phases are then mapped onto material subregions.

In this work LSFs are discretized using linear basis functions from a THB-spline basis $B_k \in \mathcal{T}$,

$$\phi_i^h(\boldsymbol{x}) = \sum_k B_k(\boldsymbol{x}) \phi_i^k, \tag{11}$$

where ϕ_i^k are the coefficients associated with LSF ϕ_i. The LSF are linearly interpolated such that the coefficients are the nodal values $\phi_i^k \equiv \phi_i(\boldsymbol{x}_k)$. This discretization is used to construct material characteristic functions and to enrich background basis functions.

3.2 Heaviside enrichment of basis functions at material interfaces

Heaviside enrichment have been widely used in PUM [4], GFEM [65], and XFEM [9] as a means to represent strong discontinuities within elements. The enrichment strategies presented in most existing literature, such as in [31, 67], add enriched basis functions for each material domain.

While effective, these global enrichment strategies can lead to artificial numerical stiffening around small geometric features. This stiffening is caused by interpolation of a state variable field in locally disconnected domains of the same material by the same basis function as shown in Fig. 2. The high-order, higher-continuity B-spline basis functions with large supports employed in this work, alongside the complex material layouts presented in Sect. 5.3, would exacerbate local stiffening effects and lead to an increased solution error. Typically, h-refinement is used to avoid locally disconnected same-material domains within the region of support of any given basis function, increasing overall system size. This work instead adopts the enrichment strategy presented by [52] which considers the material connectivity in the individual basis functions' supports.

As shown in Fig. 2, for a given function B_k with support supp(B_k), the phase IDs, defined in Eq. (10), are used to identify the L_k distinct but connected material subregions Ω_k^m, such that supp$(B_k) = \cup_{m=1}^{L_k} \Omega_k^m$. For L_k distinct subregions, the basis function B_k requires L_k enrichment levels.

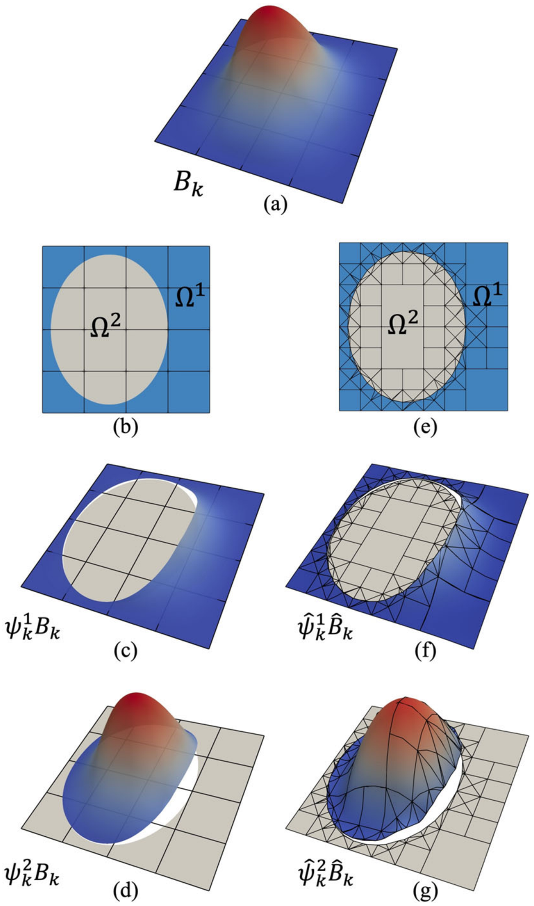


Fig. 3 A B-spline basis is defined on a structured background mesh and an example function B_k is depicted in (**a**). Using the geometry description of the material subdomains in (**b**), Heaviside enrichment is applied to form the discontinuous functions $\psi_k^1 B_k$ and $\psi_k^2 B_k$, depicted in (**c**) and (**d**) respectively. A Lagrange foreground function space is defined on the boundary-fitted mesh in (**e**). The function space is used to interpolate the enriched background functions $\hat{\psi}_k^1 \hat{B}_k$ and $\hat{\psi}_k^2 \hat{B}_k$, depicted in (**f**) and (**g**), respectively

This enrichment is then achieved through characteristic functions ψ_k^m,

$$\psi_k^m(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in \Omega_k^m \\ 0, & \text{else,} \end{cases} \tag{12}$$

such that the enriched basis functions can be expressed as

$$B_k^m(\boldsymbol{x}) = \psi_k^m(\boldsymbol{x}) B_k(\boldsymbol{x}), \quad \forall\, m \in \{1, \ldots, L_k\}. \tag{13}$$

The enriched basis functions constructed from a single non-enriched basis function, are shown in Fig. 3 for a two-material configuration. The bi-quadratic B-spline B_k depicted in Fig. 3a is enriched assigning one material to the inside and one material to the outside of the ellipse shown in Fig. 3(b). The basis function B_k is split into two enriched functions, $B_k^1(\boldsymbol{x}) = \psi_1^m(\boldsymbol{x}) B_k(\boldsymbol{x})$ in Fig. 3c and $B_k^2(\boldsymbol{x}) = \psi_k^2(\boldsymbol{x}) B_k(\boldsymbol{x})$ in Fig. 3d allowing for the representation of strong discontinuities at the material interface. Interface conditions are enforced weakly; for example C^0 continuity can be enforced at the interface using Nitsche's method [2].

4 The interpolation-based immersed boundary method

One of the core challenges associated with classic immersed methods is the construction of custom quadrature rules for the various material regions within each intersected background element. Numerous solutions exist and have been used in custom research codes, such as octree refinement [20, 39, 57], interface reconstruction and tessellation [15, 24, 46], moment-fitting [49, 66], and quadrature schemes using generalized Stokes theorem [29, 30, 63]. However, the generation of such quadrature rules can generally not be implemented within existing finite element software without major changes to the software itself.

The interpolation-based immersed approach is instead designed to utilize the integration subroutines of existing FEM codes. To this end, a boundary-fitted foreground mesh is constructed with only minimal requirements on mesh quality. Element formation is then performed on the poor quality boundary-fitted mesh using existing standard finite element routines. Using Lagrange extraction operators [59] the resulting tangent matrix and force vector are projected into the enriched THB-spline space. This can be done either on an elemental level during assembly, or globally afterwards. The resulting final problem uses an approximation of the enriched function space of the background mesh which is interpolated by the basis functions of the foreground mesh.

The following Sect. 4.1 will first introduce the thermo-elastic model problem. Section 4.2 will provide an overview of the interpolation-based approach specific to the multi-material, multi-physics problems presented in this paper. For a more general and comprehensive introduction to the approach, we refer the reader to the authors' previous work on the topic [26]. The generation of the boundary-fitted foreground mesh is discussed in Sect. 4.3.

4.1 Multi-material and multi-physics model problem

To illustrate the application of interpolation-based immersed boundary methods to multi-material and multi-physics problems, a thermo-elastic problem is introduced. This problem can be broken into a thermal subproblem and a structural subproblem.

Let a domain of interest Ω with closure denoted $\overline{\Omega}$ be composed of n material subdomains $\mathcal{M} = \{1, \cdots, n\}$

$$\overline{\Omega} = \bigcup_{m\in\mathcal{M}} \overline{\Omega}^m \quad \subset \mathbb{R}^{d_p}. \tag{14}$$

A different thermal conductivity κ^m may be associated with each material m

$$\kappa(\boldsymbol{x}) = \kappa^m, \ \boldsymbol{x} \in \Omega^m, \tag{15}$$

for $m \in \mathcal{M}$. A source term $f : \Omega \to \mathbb{R}$, a boundary heat flux term $\overline{q} : \partial\overline{\Omega} \to \mathbb{R}$ on $\Gamma_{\bar{q}} \subset \partial\overline{\Omega}$, and Dirichlet boundary data $\overline{T} : \partial\overline{\Omega} \to \mathbb{R}$ on $\Gamma_{\overline{T}} \subset \partial\overline{\Omega}$ are ascribed. The strong form for the thermal problem then reads as: Find $T : \Omega \to \mathbb{R}$ such that $\forall\, m \in \mathcal{M}$

$$\begin{aligned} -\nabla\cdot(\kappa(\boldsymbol{x})\nabla T) &= f \quad \text{in } \Omega^m,\\ \llbracket T \rrbracket &= 0 \quad \text{on all } \Gamma_{km},\\ \llbracket \boldsymbol{q} \rrbracket &= 0 \quad \text{on all } \Gamma_{km},\\ -\kappa(\boldsymbol{x})\nabla T\cdot\boldsymbol{n} &= \overline{q} \quad \text{on } \Gamma^m_{\bar{q}},\\ T &= \overline{T} \quad \text{on } \Gamma^m_{\overline{T}}, \end{aligned} \tag{16}$$

where $\Gamma_{km} = \overline{\Omega}^k \cap \overline{\Omega}^m \neq \emptyset$, with $k \in \mathcal{M}$ and $k \neq m$, are the material interfaces, and $\llbracket\cdot\rrbracket = (\cdot)^k - (\cdot)^m$ is the jump of a given quantity over an interface Γ_{km}. The material fields are defined $T^m = T(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega^m$, and $\boldsymbol{q}^m = -\kappa^m\nabla T^m$. The domains $\Gamma^m_{\bar{q}} = \Gamma_{\bar{q}} \cup \partial\overline{\Omega}^m$, and $\Gamma^m_{\overline{T}} = \Gamma_{\overline{T}} \cup \partial\overline{\Omega}^m$ are the intersections of the domain boundaries with the material subdomain boundaries. $\boldsymbol{n}$ denotes the surface normal.

The domain of interest $\overline{\Omega}$ is embedded into a hierarchically refined background mesh $\mathcal{K}_T$, generated using the sequence of refined meshes $\mathcal{K}^l$ and subdomains Ω^l_T.[1] Note that the largest subdomain Ω^0_T must be chosen such that the closure of the domain of interest is a subset, $\overline{\Omega} \subset \Omega^0_T$. The mesh $\mathcal{K}_T$ is associated with the enriched THB-spline basis $\mathcal{T}_T = \{B^T_i\}$. The temperature field is discretized using the function space

$$\mathcal{V}^h_T = \text{span}\{B^T_i \mid \text{supp}(B^T_i) \cap \overline{\Omega} \neq \emptyset\}. \tag{17}$$

The discrete form can then be defined as: Find $T^h \in \mathcal{V}^h_T$ such that $\forall\, \theta^h \in \mathcal{V}^h_T$,

$$\begin{aligned} &\sum_{m=1}^{n}\left[\int_{\Omega^m}\kappa\nabla T^h\cdot\nabla\theta^h d\Omega\right] - \int_{\Omega} f\theta^h d\Omega - \int_{\Gamma_{\bar{q}}}\overline{q}\theta^h d\Gamma\\ &\quad = \mathcal{R}^D_T + \mathcal{R}^I_T\ , \end{aligned} \tag{18}$$

where $\mathcal{R}^D_T$ and $\mathcal{R}^I_T$ are Dirichlet and interface residual terms. The temperature Dirichlet residual is the result of Nitsche's method [51] enforcement of the Dirichlet boundary condition,

$$\begin{aligned} \mathcal{R}^D_T = \sum_{m=1}^{n}\Bigg[&\mp\int_{\Gamma^m_{\overline{T}}}\kappa(T^h - \overline{T})(\nabla\theta^h\cdot\boldsymbol{n})\,d\Gamma\\ &-\int_{\Gamma^m_{\overline{T}}}\kappa\theta^h(\nabla T^h\cdot\boldsymbol{n})\,d\Gamma\\ &+\int_{\Gamma^m_{\overline{T}}}\frac{\beta^D_T\kappa}{h}(T^h - \overline{T})\theta^h\,d\Gamma\Bigg], \end{aligned} \tag{19}$$

where $\beta^D_T \geq 0$ is a user defined constant. The first integral of Eq. (19) will be negative for the symmetric version of Nitsche's method (which is employed in numerical examples in this work) or positive for the non-symmetric version. h is taken as the characteristic element size on the foreground mesh, differing from the usual implementation where h is the element size on the background mesh.

The temperature interface conditions, lines 2 and 3 of Eq. (16), are also enforced through a Nitsche-like method, resulting in the temperature interface residual

$$\begin{aligned} \mathcal{R}^I_T = \sum_{i=1}^{n}\sum_{j=i+1}^{n}\Bigg[&-\int_{\Gamma_{ij}}\llbracket T^h\rrbracket\{\kappa\nabla\theta^h\}\cdot\boldsymbol{n})\,d\Gamma\\ &-\int_{\Gamma_{ij}}\llbracket\theta^h\rrbracket\{\kappa\nabla T^h\}\cdot\boldsymbol{n})\,d\Gamma\\ &+\int_{\Gamma_{ij}}\gamma^{ij}_T\llbracket T^h\rrbracket\llbracket\theta^h\rrbracket\,d\Gamma\Bigg], \end{aligned} \tag{20}$$

where $\{\cdot\} = w^i(\cdot)^i - w^j(\cdot)^j$ is the weighted average of a given quantity. Motivated by the formulation in [2], these weights are defined as

$$\begin{aligned} w^i &= \frac{(h^i)^{d_p}/\omega^i}{(h^i)^{d_p}/\omega^i + (h^j)^{d_p}/\omega^j} \quad \text{and}\\ w^j &= \frac{(h^j)^{d_p}/\omega^j}{(h^i)^{d_p}/\omega^i + (h^j)^{d_p}/\omega^j}, \end{aligned} \tag{21}$$

where h^m is the characteristic size of the foreground element in domain Ω^m bordering the interface facet, ω^m is the characteristic material parameter, which for the thermal subproblem is κ^m, and d_p is the domain dimension. The penalty parameter γ^{ij}_T is defined as

$$\gamma^{ij}_T = 2\beta^I_T\frac{(h^i)^{d_p-1} + (h^j)^{d_p-1}}{(h^i)^{d_p}/\omega^i + (h^j)^{d_p}/\omega^j}, \tag{22}$$

where $\beta^I_T \geq 0$ is a user specified constant.

[1] Here the subscript $(\cdot)_T$ refers to entities associated with the thermal subproblem. The subscript $(\cdot)_u$ will refer to entities associated with the structural subproblem. T and u are used as superscripts when referring to entities that require subscripts for indices, such as B-spline basis functions (B^T_i and B^u_i).

The thermal subproblem can be stated compactly as the variational problem: Find $T^h \in \mathcal{V}_T^h$ such that $\forall\, \theta^h \in \mathcal{V}_T^h$

$$a_T(T^h, \theta^h) = L_T(\theta^h), \tag{23}$$

where $a_T(T, \theta)$ and $L_T(\theta)$ can be computed from Eq. (18).

The structural subproblem may utilize a differently refined background discretization. In this case a separate sequence of refined domains Ω_u^l may be selected. The structural subproblem's background mesh $\mathcal{K}_u$ is constructed with this sequence of subdomains and the same sequence of refined tensor-product Cartesian grids $\mathcal{K}^l$. The basis $\mathcal{T}_u = \{B_i^u\}$ associated with the mesh is used for the components of the displacement field $\boldsymbol{u}$. Each displacement component is discretized using the function space

$$\mathcal{V}_u^h = \operatorname{span}\{B_i^u \mid \operatorname{supp}(B_i^u) \cap \Omega \neq \emptyset\}. \tag{24}$$

Using a similar derivation (given in full in "Appendix B") as applied to the temperature subproblem the mechanical variation problem is compactly written as: Find $\boldsymbol{u}^h \in \boldsymbol{\mathcal{V}}_u^h = [\mathcal{V}_u^h, \mathcal{V}_u^h]$ such that, $\forall \boldsymbol{v}^h \in \boldsymbol{\mathcal{V}}_u^h$

$$a_u(\boldsymbol{u}^h, \boldsymbol{v}^h) = L_u(\boldsymbol{v}^h). \tag{25}$$

The two subproblems are coupled through a constitutive model accounting for the thermal expansion by computing the mechanical strain $\boldsymbol{\varepsilon}_m$ as

$$\begin{aligned}\boldsymbol{\varepsilon}_m(\boldsymbol{u}, T) &= \boldsymbol{\varepsilon}_u(\boldsymbol{u}) - \boldsymbol{\varepsilon}_T(T) \\ &= \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\mathrm{T}}\right) - \alpha(T - T_0)\boldsymbol{I},\end{aligned} \tag{26}$$

where α is the thermal expansion coefficient and T_0 is the temperature in the reference configuration. $\boldsymbol{I}$ is the identity matrix.

The fully coupled system is then given by the variational problem: Find $(\boldsymbol{u}^h, T^h) \in [\boldsymbol{\mathcal{V}}_u^h, \mathcal{V}_T^h]$ such that, $\forall\, (\boldsymbol{v}^h, \theta^h) \in [\boldsymbol{\mathcal{V}}_u^h, \mathcal{V}_T^h]$

$$\begin{aligned} a_T(T^h, \theta^h) = L_T(\theta^h) \text{ and} \\ a_u(\boldsymbol{u}^h, \boldsymbol{v}^h) - b(T^h, \boldsymbol{v}^h) = L_u(\boldsymbol{v}^h),\end{aligned} \tag{27}$$

where the form $b(T, \boldsymbol{v})$ is a result of the coupling conditions and is given by

$$\begin{aligned} b(T, \boldsymbol{v}) = &\sum_{m=1}^{n}\left[\int_{\Omega^m} \boldsymbol{\varepsilon}^u(\boldsymbol{v}) : \boldsymbol{\varepsilon}^T(T)\mathrm{d}\Omega\right] \\ &+ \int_{\Gamma_{\overline{u}}} \boldsymbol{C} : \boldsymbol{\varepsilon}^T(T) \cdot \boldsymbol{n} \cdot \boldsymbol{v}^h \mathrm{d}\Gamma \\ &- \sum_{i=1}^{n}\sum_{j=i+1}^{n}\left[\int_{\Gamma_{ij}} [\![\boldsymbol{v}]\!] \cdot \{\boldsymbol{C} : \boldsymbol{\varepsilon}^T(T)\} \cdot \boldsymbol{n}\, d\Gamma\right].\end{aligned} \tag{28}$$

4.2 Interpolated basis functions

In traditional immersed boundary methods custom quadrature rules would be used to evaluate the integral in the weak form of the coupled problem, given in Eq. (27). The main idea of the interpolation-based immersed paradigm is to interpolate the background basis functions using a space of Lagrange functions defined on a foreground mesh, which can be integrated with classical quadrature methods. This workflow thus introduces an interpolated background function space for the thermal subproblem

$$\widehat{\mathcal{V}}_T^h = \operatorname{span}\{\widehat{B}_i^T \mid \operatorname{supp}(\widehat{B}_i^T) \cap \Omega \neq \emptyset\}, \tag{29}$$

where the interpolated basis functions are defined as

$$\widehat{B}_i^T := \sum_{j=1}^{\nu} M_{ij}^T N_j \tag{30}$$

where

$$M_{ij}^T := B_i^T(\boldsymbol{x}_j) \tag{31}$$

is the Lagrange extraction operator. $\{N_j\}_{j=1}^{\nu}$ is the basis of a Lagrange FE space with nodal points $\boldsymbol{x}_j$ such that $N_i(\boldsymbol{x}_j) = \delta_{ij}$. Here ν is the number of foreground basis functions. The same foreground space is used to interpolate the background bases for both the temperature and the displacement state variables.

The approximations of the temperature field becomes

$$T^h = \sum_{i=1}^{n_T} \widehat{B}_i^T d_i^T = \sum_{i=1}^{n_T}\sum_{j=1}^{\nu} M_{ij}^T N_j d_i^T, \tag{32}$$

where $\{d_i^T\}_{i=1}^{n_T}$ are the unknown coefficients and n_T is the number of basis functions in the temperature field's interpolated background B-spline basis. The displacement is similarly discretized with vector valued function spaces as

$$\boldsymbol{u}^h = \sum_{I=1}^{(d_p \cdot n_u)} \widehat{\boldsymbol{B}}_I^u d_I^{\boldsymbol{u}} = \sum_{I=1}^{(d_p \cdot n_u)} \sum_{J=1}^{(d_p \cdot \nu)} M_{IJ}^{\boldsymbol{u}} \boldsymbol{N}_J d_I^{\boldsymbol{u}}. \tag{33}$$

Details regarding this discretization are given in "Appendix B".

The variational problem in Eq. (27) can be assembled using the interpolated bases in Eqs. (32) and (33) to form the linear system

$$\begin{bmatrix} \boldsymbol{K}^{\theta\theta} & \boldsymbol{0} \\ \boldsymbol{K}^{\theta\boldsymbol{v}} & \boldsymbol{K}^{\boldsymbol{vv}} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}^{T} \\ \boldsymbol{d}^{\boldsymbol{u}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}^{\theta} \\ \boldsymbol{f}^{\boldsymbol{v}} \end{bmatrix}, \tag{34}$$

where

$$K_{ij}^{\theta\theta} = a_T(\widehat{B}_i^T, \widehat{B}_j^T), \tag{35}$$

$$K_{iJ}^{\theta\boldsymbol{v}} = b(\widehat{B}_i^T, \widehat{\boldsymbol{B}}^{\boldsymbol{u}}_J), \tag{36}$$

$$K_{IJ}^{\boldsymbol{vv}} = a_{\boldsymbol{u}}(\widehat{\boldsymbol{B}}^{\boldsymbol{u}}_I, \widehat{\boldsymbol{B}}^{\boldsymbol{u}}_J), \tag{37}$$

$$f_i^{\theta} = L_T(\widehat{B}_i^T), \quad \text{and} \tag{38}$$

$$f_i^{\boldsymbol{v}} = L_u(\widehat{\boldsymbol{B}}^{\boldsymbol{u}}_I) \tag{39}$$

are the matrix entries.

By consolidating the extraction operators for all components in a single matrix

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}^{T} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}^{\boldsymbol{u}} \end{bmatrix} \tag{40}$$

the linear system can be rewritten as

$$\boldsymbol{M}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{A}^{\theta\theta} & \boldsymbol{0} \\ \boldsymbol{A}^{\theta\boldsymbol{v}} & \boldsymbol{A}^{\boldsymbol{vv}} \end{bmatrix} \boldsymbol{M} \begin{bmatrix} \boldsymbol{d}^{T} \\ \boldsymbol{d}^{\boldsymbol{u}} \end{bmatrix} = \boldsymbol{M}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{b}^{\theta} \\ \boldsymbol{b}^{\boldsymbol{v}} \end{bmatrix}, \tag{41}$$

where

$$A_{ij}^{\theta\theta} = a_T(N_i, N_j), \tag{42}$$

$$A_{iJ}^{\theta\boldsymbol{v}} = b(N_i, \boldsymbol{N}_J), \tag{43}$$

$$A_{IJ}^{\boldsymbol{vv}} = a_{\boldsymbol{u}}(\boldsymbol{N}_I, \boldsymbol{N}_J), \tag{44}$$

$$b_i^{\theta} = L_T(N_i), \quad \text{and} \tag{45}$$

$$b_I^{\boldsymbol{v}} = L_u(\boldsymbol{N}_I). \tag{46}$$

The quantities $\boldsymbol{A}^{\theta\theta}$, $\boldsymbol{A}^{\theta\boldsymbol{v}}$, $\boldsymbol{A}^{\boldsymbol{vv}}$, $\boldsymbol{b}^{\theta}$, and $\boldsymbol{b}^{\boldsymbol{v}}$ are evaluated and assembled on the boundary-fitted foreground mesh. As the foreground basis is the typical conforming Lagrange polynomial basis, existing commercial or open-source FE software applying standard quadrature rules may be utilized.

The extraction operators can either be computed globally, as suggested in Eq. (42), or on an element level for more efficient implementation. However, the latter option requires a modification of the assembly maps in existing FE software. After solving the linear system in Eq. (41) with the interpolated basis, the solution is post-processed by projecting the solution for each field onto the foreground basis.

A visual example of the interpolation of an enriched B-spline basis function is given in Fig. 3. The boundary-fitted foreground mesh in Fig. 3(e) is used to define a discontinuous Lagrange polynomial foreground function space. The enriched background functions B_k^1 and B_k^2 are interpolated with this function space, as shown in Fig. 3(f) and (g).

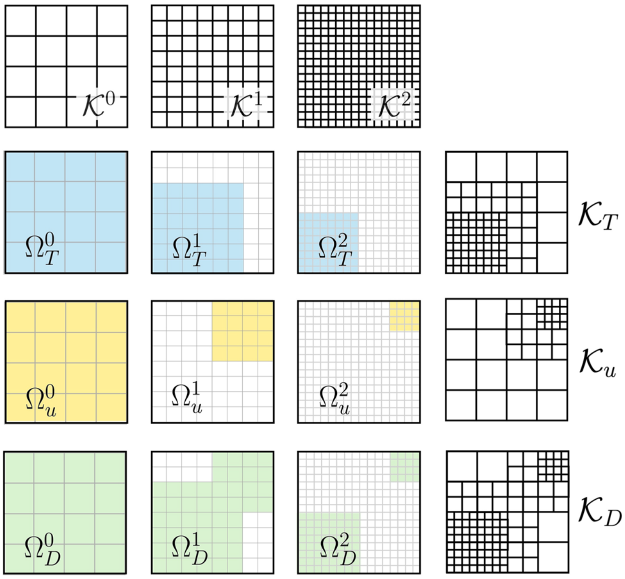


Fig. 4 A series of uniformly refined meshes $\mathcal{K}^l$ is shown in the top row. Series of nested subdomains, $\Omega_T^{l+1} \subseteq \Omega_T^l$, shown in the second row, and $\Omega_u^{l+1} \subseteq \Omega_u^l$, shown in the third row, are defined for each state variable. For the decomposition mesh the series of subdomains Ω_D^l, shown in the last row, is defined such that the domain on each level l contains the union of the l^{th} level domains for both variable fields. The hierarchically refined meshes $\mathcal{K}_T$, $\mathcal{K}_u$, and $\mathcal{K}_D$, shown in the rightmost column, are constructed with the mesh series $\mathcal{K}^l$ in the top row and their respective subdomain series

In this work, B-spline background spaces are utilized to exploit their superior approximation properties as compared to traditional finite element function spaces [23] and to build upon previous work done with enriched THB-splines in XIGA implementations [61]. The interpolation scheme presented here can be utilized with other background basis functions as in the previous work by the authors [26], where both B-splines and classic Lagrange polynomial bases were used. The method can be applied to multi-material problems provided the background basis used is sufficiently enriched at material interfaces.

4.3 Foreground mesh generation

Both state variable fields are interpolated using the same Lagrange FE space $\{N_j\}_{j=1}^{\nu}$ which is constructed on a boundary-fitted foreground mesh. To generate this foreground mesh, the elements of a background mesh intersected by interfaces are decomposed into triangles and tetrahedrons whose facets approximately reconstruct the interfaces.

To ensure the foreground mesh is sufficiently refined, a background mesh for decomposition $\mathcal{K}_D$ is constructed from

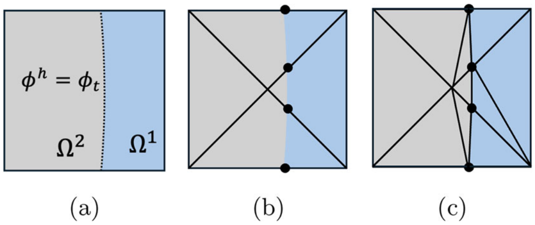


Fig. 5 Foreground integration meshes are formed by triangulating cut elements of the decomposition mesh $\mathcal{K}_D$. **a** A cell is intersected by the isocontour of the discretized level set function $\phi^h = \phi_t$ defining a material interface. **b** The cell is subdivided into triangular cells and isocontour-edge intersections (indicated by black cicles) are computed. **c** Using the intersection points as nodal points, the cell is further subdivided

the series of hierarchically refined meshes $\mathcal{K}^l$,

$$\mathcal{K}_D := \bigcup_{l=0}^{r-1} \{K \in \mathcal{K}^l | K \in \Omega^l_D \text{ and } K \notin \Omega^{l+1}_D\}, \tag{47}$$

where $\Omega^l_D \supseteq \Omega^l_T \cup \Omega^l_u$ is a series of subdomains containing the union of the subdomains used for each subproblem's discretization. This ensures that $\mathcal{K}_D$ is at least as refined as the meshes used for each subproblem's discretization and allows for additional refinement of the foreground to improve geometric resolution. The construction of a decomposition mesh $\mathcal{K}_D$ is illustrated in Fig. 4.

The decomposition mesh $\mathcal{K}_D$ is triangulated by first applying a pre-defined triangulation to the intersected background elements, as shown in Fig. 5a, b. This pre-defined triangulation forms 4 triangular elements in 2D domains or 24 hexahedral elements in 3D. Through root finding along elemental edges, the location of the isocontour is found, indicated by the black dots in Fig. 5b. A subdivision template is then applied to each intersected triangle/tetrahedron, as shown in Fig. 5c, to further subdivide the triangles/tetrahedrons into a set of triangles and tetrahedrons whose facets follow the LS isocontour. The last step is repeated recursively for each individual LSF ϕ^h_i which enables sharp geometric corners and edges to be captured where multiple interfaces meet. This approach has been used previously by, e.g., [62]. The resulting approximation of the interface is piecewise linear and depends on the resolution of the decomposition mesh $\mathcal{K}_D$.

Note that the sliver elements and poor aspect ratios in meshes produced by this method are still suitable for the interpolation of the background basis functions which is not bound by the typical mesh quality constraints of traditional FEM [40]. The resulting foreground mesh is of mixed element type (triangles and quadrilaterals in 2D, or hexahedrons and tetrahedrons in 3D) and contains hanging nodes. To accommodate the hanging nodes on the foreground mesh and to adequately interpolate the discontinuous enriched background basis functions described in Sect. 3.2, this method employs discontinuous Galerkin type elements for foreground function spaces. As the original THB-spline background basis maintains at minimum C^{p-1} continuity within each material domain, the continuity of the interpolated basis is likewise C^{p-1} continuous where interpolation is exact [59]. This work expands upon previous results using approximate extraction methods [26], where the constraints placed on the foreground Lagrange basis are lessened while numerical accuracy is maintained.

The sliver elements resulting from this procedure do not themselves present problems with the interpolation-based workflow. However, due to the arbitrary location of material interfaces with respect to cell boundaries, it is possible for cells to be cut such that only a small portion of a basis function's support resides inside a given material domain. These small cell cuts result in sparsely supported basis functions, which can present issues with stability and linear conditioning.

Numerous strategies exist to mitigate these issues and were recently reviewed in [18]. Strategies include basis function removal [21], ghost stabilization [13], and basis function agglomeration [5] or extension [14], and have the potential to be implemented within the presented interpolation-based framework. The benchmark problems presented in this work do not require special treatment of sparsely support basis functions and the authors leave the implementation of these stabilization strategies to future work.

5 Numerical results

The accuracy of the proposed method is demonstrated through the study of several benchmark problems. Problems were defined in the Python-based open-source FE code FEniCSx, using foreground meshes and extraction operators generated by the open-source XIGA code MORIS, available at github.com/kkmaute/moris [45]. The source code with which the results were generated is also available at github.com/jefromm/EXHUME_dolfinX [25].

5.1 Resolving discontinuities in solution fields through Heaviside enrichment

In this Subsection, a multi-material beam undergoing a spatially varying heat load presents a weakly discontinuous temperature solution field. In this work, weak discontinuities refer to discontinuities in the gradients of solution fields. The solution is approximated with an interpolated Heaviside-enriched background basis, using the interpolation-based immersed boundary workflow. The convergence results from this study demonstrate the accuracy of this method's enrichment scheme for multi-material problems. Beams in both 2D and 3D domains are considered.

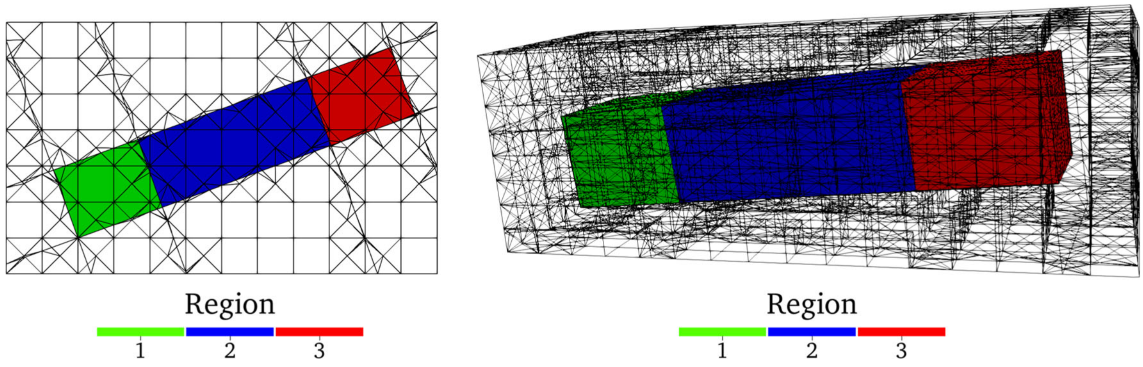


Fig. 6 The geometric configurations for the 2D (left) and 3D (right) domains. A three-material beam is embedded in a structured background grid and rotated such that material interfaces do not align with element edges (in 2D) or facets (in 3D). Elements intersected by the level set functions defining the beam geometry are triangulated to form a boundary-fitted foreground mesh

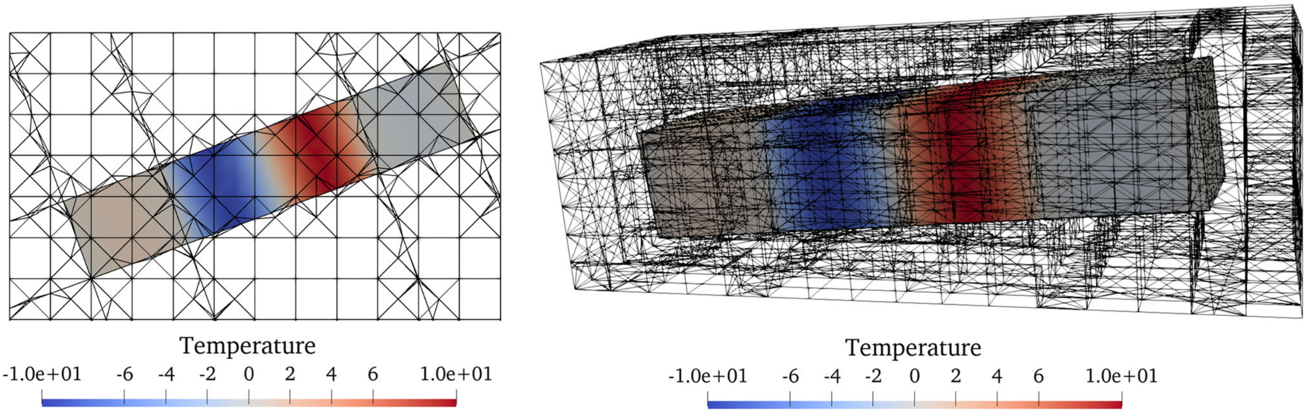


Fig. 7 The temperature solution field plotted for both domain geometries, using a bi(tri)-quadratic B-spline basis interpolated with a bi(tri)-quadratic foreground Lagrange space. The solution is weakly discontinuous at material interfaces

The beam is initially defined with corner coordinates (0, 0) and (L, H) in 2D, and (0, 0, 0) and (L, H, H) in 3D, with $L = 5$ and $H = 1$. The beam is divided into 3 sections, with interfaces at $x = L/4$ and $x = 3L/4$. Each section of the beam is assigned a thermal conductivity $\kappa_1 = 1.0$, $\kappa_2 = 0.1$, and $\kappa_3 = 1.0$. To ensure the non-conformity of the material interfaces with respect to the background mesh facets, the 2D beam is rotated about the origin by angle $\phi = 20°$, while the 3D beam is rotated about y- and z-axes by angles $\phi_y = -5°$ and $\phi_z = 5°$, respectively. The 2D beam is embedded into rectangle with corner coordinates $(-1.0, -0.5)$ and (5,3) and the 3D beam is embedded into a rectangular prism with corner coordinates $(-0.5, -0.25, -0.25)$ and (5.5,1.75,1.75). The geometric configurations are shown in Fig. 6. Each edge (in 2D) or plane (in 3D) of the beam is defined by a level set function which extends beyond the beam domain in the mesh. The functions are extended through the entire mesh to fully resolve the corners (in 2D) or edges (in 3D) of the beam.

Thermal diffusion is governed by the Poisson equation. The strong and weak forms of this problem are given by Eqs. (16) and (18) in Sect. 4.1. The source term

$$q_B = -\nabla \cdot (\kappa \nabla T_{ex}) \tag{48}$$

is constructed from the exact solution

$$T_{ex}(x') = \frac{1}{\kappa} \sin\left(\frac{4\pi}{L} x'\right). \tag{49}$$

In 2D the beam-aligned coordinate is expressed in global coordinates as

$$x' = x \cos(-\phi) - y \sin(-\phi), \tag{50}$$

and in 3D

$$x' = z \sin(-\phi_y) + \big(x \cos(-\phi_z) - y \sin(-\phi_z)\big) \cos(-\phi_y), \tag{51}$$

where $\boldsymbol{x} = [x, y, z]$ are the mesh coordinates. The exact solution is imposed as Dirichlet boundary data on the ends of the bar, $x' = 0$ and $x' = L$.

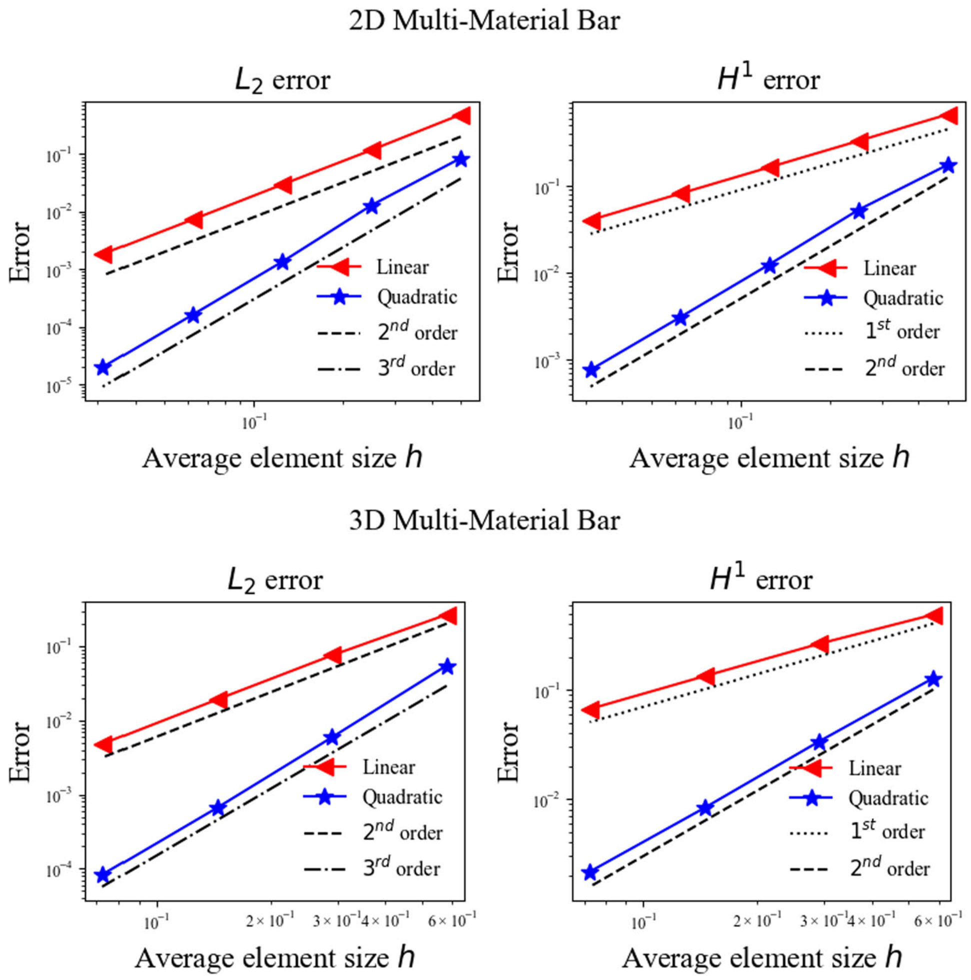


Fig. 8 Ideal convergence rates are seen for both bi(tri)-linear and quadratic B-spline basis functions, which were interpolated with equal order foreground Lagrange function spaces. The convergence rates are ideal for both the 2D domain (top) and 3D domain (bottom)

For the 2D domains a suite of meshes with average background element sizes $h \in 0.5 \times [1, 0.5, 0.25, 0.125, 0.0625]$ was used. In 3D, the background element sizes $h \in 0.58 \times [1, 0.5, 0.25, 0.125]$ were used. The foreground meshes were constructed with the workflow described in Sect. 4.3. Results from the 2D mesh with background element size $h = 0.5$ and from the 3D mesh with background element size $h = 0.29$ using bi(tri)-quadratic Lagrange foreground bases to interpolate bi(tri)-quadratic enriched B-spline bases are shown in Fig. 7. Convergence rates are plotted in Fig. 8 for both bi(tri)-linear and quadratic enriched background spline spaces interpolated with equal-order foreground bases. Ideal convergence rates validate the interpolation-based immersed boundary workflow for multi-material problems.

5.2 Approximating curved geometries through local foreground refinement

A major challenge in the modeling of multi-material problems is the discretization of material interfaces. In this work, local refinement of the foreground mesh is performed to increase geometric resolution without affecting the number of degrees of freedom in the solution space. In this example the linear elastic behavior of an infinite plate with an embedded circular inclusion of radius $R = 0.5$ is modeled, with local refinement employed to improve the approximation of the inclusion geometry.

The inclusion is comprised of Material 1 with Lamé constants $\lambda_1 = 497.16$ and $\mu_1 = 390.63$, while the exterior plate is made of Material 2 with $\lambda_2 = 656.79$, and $\mu_2 = 338.35$. A uniform isotropic eigenstrain of $\varepsilon_0 = 0.1$ is imposed on the inclusion. The weakly discontinuous analytic solution for the radial displacement is given in [70] as

$$u_r = \begin{cases} C_1 r, & r \leq R, \\ C_1 \dfrac{R^2}{r}, & r \geq R, \end{cases} \tag{52}$$

where

$$C_1 = \frac{(\lambda_1 + \mu_1)\varepsilon_0}{\lambda_1 + \mu_1 + \mu_2}. \tag{53}$$

Exploiting symmetry, only the upper right quadrant of the plate is modeled as shown in Fig. 9. Symmetry conditions are

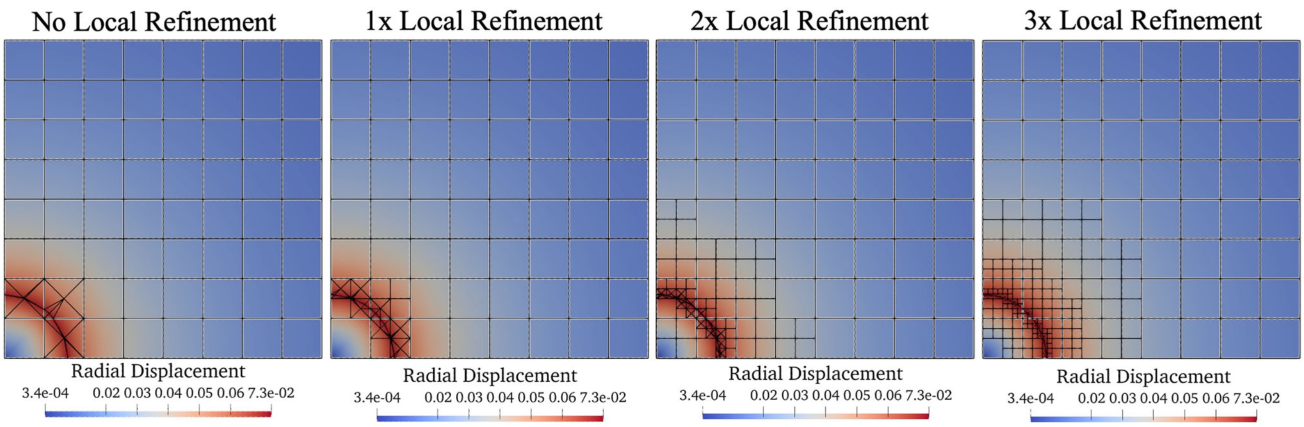


Fig. 9 Approximated radial displacement of the eigenstrain problem. The background mesh is shown in white, while the foreground integration mesh is overload in black. The background mesh remains the same as local refinement is applied to the foreground for improved geometric resolution. The images correspond to the coarsest refinement level with background element size $h = 0.625$

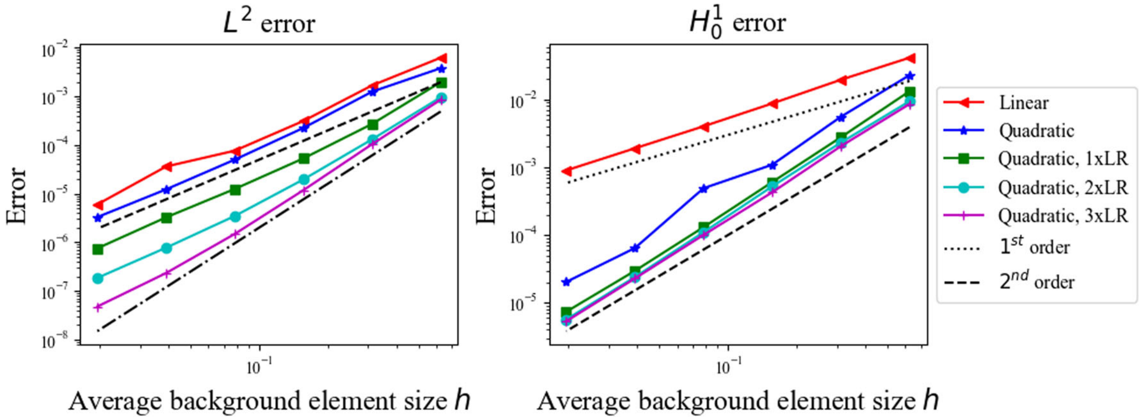


Fig. 10 Error convergence data for the eigenstrain problem, illustrating the efficacy of foreground refinement. With no foreground refinement, the L^2 convergence rate is limited by the geometric error. With sufficient foreground refinement (3x LR), the convergence rate approaches the ideal of 3

enforced on the left and bottom edges of the domain, and the exact displacement is prescribed on the right and top edges. The solution domain is a 5 × 5 square, with a quarter circle at the lower left corner.

The strong and discrete forms of the multi-material linear elasticity PDE are detailed in "Appendix B". For this example the mechanical strain is computed by

$$\boldsymbol{\varepsilon}_m(\boldsymbol{u}) = \begin{cases} \boldsymbol{\varepsilon}_u(\boldsymbol{u}) - \varepsilon_0 \boldsymbol{I}, & \boldsymbol{x} \in \Omega_1 \\ \boldsymbol{\varepsilon}_u(\boldsymbol{u}), & \boldsymbol{x} \in \Omega_2 \end{cases} \tag{54}$$

where the total strain $\boldsymbol{\varepsilon}_u(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\mathrm{T}})$.

The domain Ω is immersed into an axis-aligned 5 × 5 square on which the enriched background B-spline spaces are constructed. A suite of background meshes with characteristic element lengths $h \in 0.625 \times [1, 0.5, 0.25, 0.125, 0.0625, 0.03125]$ are used to generate convergence data.

The foreground meshes used here are locally refined about the material interfaces, as seen in Fig. 9. The background basis functions remain constant, meaning that there is no increase to the number of solution degrees of freedom with foreground refinement.

The convergence plots in Fig. 10 show the expected convergence rates for the bi-linear B-spline function space. With the bi-quadratic function space and no foreground refinement, the dominating effects of geometric error are shown with degraded L^2 convergence rates. Geometric error is reduced through local foreground refinement, allowing the function approximation error to dominate the overall convergence. With local foreground refinement the convergence rates approach the ideal rates exhibited by the method for problems without geometric approximation error, as in Sect. 5.1.

The foreground meshes generated using this octree local refinement strategy include hanging nodes, which are difficult for most commercial or open-source finite element software to handle. To accommodate these hanging nodes, this interpolation-based workflow employs C^{-1} discontinuous foreground Lagrange polynomial basis functions,. Classical discontinuous Galerkin methods require augmentation of the variational form to enforce continuity at cell interfaces [16], but this augmentation is not done within this workflow. The results in this work suggest that for a problem where conforming FE methods would require C^0 continuity, a C^{-1} interpolated basis can be employed provided the background basis is at least C^0 continuous. More generally, so long as the background basis is sufficiently continuous it may be interpolated with a less-continuous foreground basis. This attribute of interpolated-based methods was first exploited in [26] to approximate fourth-order PDEs with a background basis of quadratic B-spline bases, which are C^1 continuous, interpolated with a foreground basis of quadratic Lagrange polynomials, which are only C^0 continuous.

5.3 Image-based thermo-mechanical analysis of composite materials, utilizing multiple levels of local refinement

The capability of this method to tackle distinct discretization requirements of state variable fields within a multi-physics problem is illustrated here in a coupled thermo-elastic problem. This problem is posed on an alumina-epoxy composite sample undergoing simultaneous heating and loading conditions. Separate background discretizations are interpolated with a single foreground discretization for the two fields.

The geometry of the composite sample is taken from a real micro-CT image converted to an implicit level-set description. The micro-CT image, from [71] and shown in Fig. 11a, is made up of 200 × 200 pixels, with a pixel size of 8μm, and the specimen is 1.6mm by 1.6mm. The epoxy is represented by the grey background while the alumina particles appear white. The image was then manually processed to generate the smoothed image shown in 11b, which was then converted to the implicit level-set description. The following material properties, from [7, 44], were used: Poisson ratios $\nu_{\text{Al}} = 0.23$ and $\nu_{\text{Ep}} = 0.358$, elastic moduli $E_{\text{Al}} = 320e^9$ Pa and $E_{\text{Ep}} = 3.66e^9$ Pa, thermal conductivities $\kappa_{\text{Al}} = 25.0$ W/mK and $\kappa_{\text{Ep}} = 0.14$ W/mK, and thermal expansion coefficients $\alpha_{\text{Al}} = 15e^{-6}$ 1/C^o and $\alpha_{\text{Ep}} = 65e^{-6}$ 1/C^o.

The process described in Sect. 4.3 was used to construct a foreground integration mesh, beginning with a uniform 80 element by 80 element axis aligned decomposition mesh. Two levels of local refinement were applied about the material interfaces, and the refined decomposition mesh was triangulated. The resulting foreground mesh, shown in 12a, contains 81,809 cells.

For comparison purposes, a similar workflow was used to generate a boundary-fitted mesh for use in classical FEM. Classical FEM with FEniCSx requires a single element type mesh without hanging nodes. To avoid hanging nodes and to sufficiently resolve the gradients of the state variable fields the decomposition mesh was uniformly refined three times forming a 640 element by 640 element square mesh. The cut cells were triangulated to create a boundary-fitted mesh, and then the remaining quadrilateral elements were triangulated. The mesh was then modified with the software package Coreform Cubit 2023.11 to improve mesh quality metrics. The resulting mesh, shown in Fig. 12b, contains 1,675,860 cells. The method described here was used to ensure the level set descriptions of the material interfaces matched between this mesh and the foreground mesh used with the interpolation-based immersed boundary method.

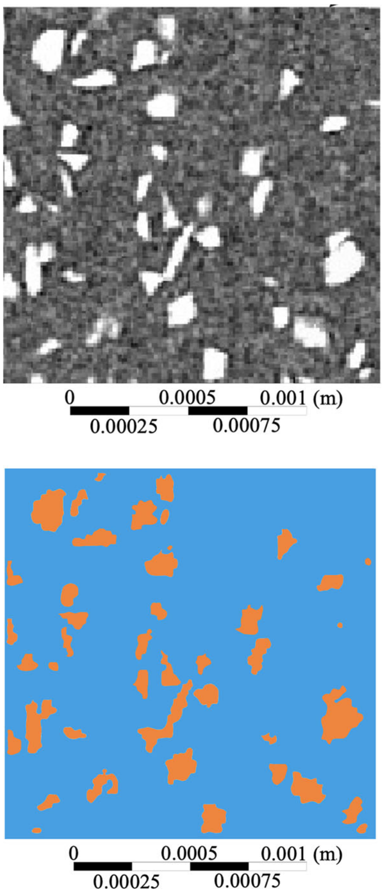
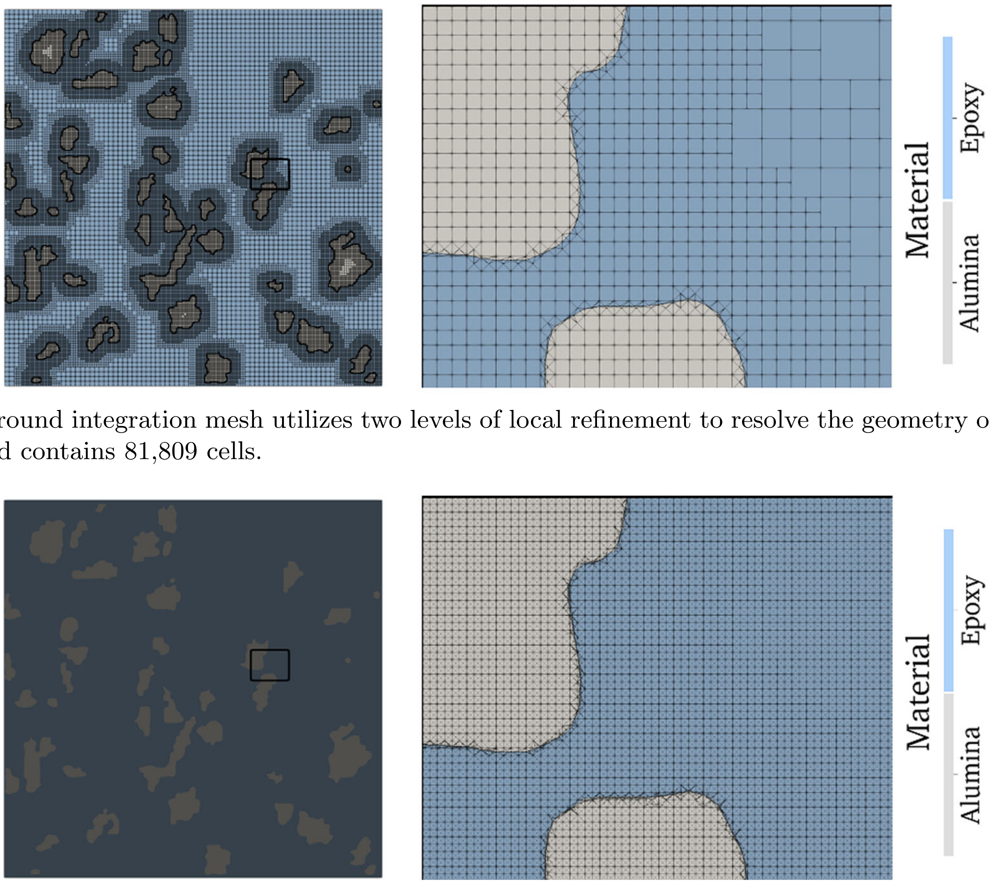


Fig. 11 Top:Micro-CT image of alumina-epoxy composite, where the white sections signify alumina particles and the grey is the surrounding epoxy. Bottom: Smoothed image used to generate the LSF geometric description

With this sample, a heated compression-shear test was simulated. The top and bottom displacements were imposed as $\boldsymbol{u}_{top} = [-0.01, -0.01]$ mm and $\boldsymbol{u}_{bottom} = [0, 0]$ mm. The temperature at the top and bottom edges were specified as $T_{top} = 0$°C and $T_{bottom} = 100$°C. The environmental



(a) The foreground integration mesh utilizes two levels of local refinement to resolve the geometry of the material interfaces, and contains 81,809 cells.

(b) The mesh used for the classical boundary-fitted FEM comparison was generated with 3 levels of uniform refinement. The mesh elements were uniformly triangulated and modified to improve mesh quality. The mesh contains 1,675,860 cells.

Fig. 12 The whole domain is shown in the images on the left with the box indicating the region shown in the zoomed in view on the left

temperature was $T_0 = 0^o C$. The sides were left both traction and heat flux free.

The governing equations for heat conduction and linear elasticity were coupled via the constitutive law adding a thermal expansion component to the total strain as in Eq. (26). Dirichlet boundary conditions for each field were imposed using Nitsche's method. Nitsche's terms enforcing continuity in T and $\boldsymbol{u}$ were also imposed upon the allumina-epoxy interfaces.

Bi-linear B-splines are used for the temperature field and bi-quadratic B-splines are used for each component of the displacement field, and both fields are interpolated using bi-quadratic Lagrange polynomials. For the boundary-fitted FEM comparison, Dirichlet boundary conditions were likewise enforced using Nitsches method, a bi-linear Lagrange basis was used for the temperature field, and a bi-quadratic Lagrange basis was used for the displacement. The results for the temperature field, the temperature gradient magnitude, the displacement magnitude, and the strain magnitude are shown in Figs. 13, 14, 15, and 16, respectively. Within the figures, the foreground meshes are drawn in black while the background meshes are overlaid in white. The same foreground mesh, with two levels of local refinement to resolve the composite geometries, is used for each discretization.

Progressive levels of local refinement were applied to the background B-spline function spaces, as seen in the rows of Figs. 13, 14, 15, and 16. This local refinement was implemented with THB-splines, as described in Sect. 2.2. The regions of refinement with high-order splines are larger than those for the low-order to adequately support the truncated basis. This can be seen by comparing the white background meshes used for the temperature field depicted in Figs. 13 and 14 with those used for the displacement field in Figs. 15 and 16. The number of degrees of freedom associated with the various background discretizations are shown in Table 1.

With two levels of local refinement, the results were in qualitative agreement with the ones of the uniformly refined classical FEM example with far fewer degrees of freedom. As previously noted, the uniformly refined meshes were initially generated with the same workflow used to create the bound-

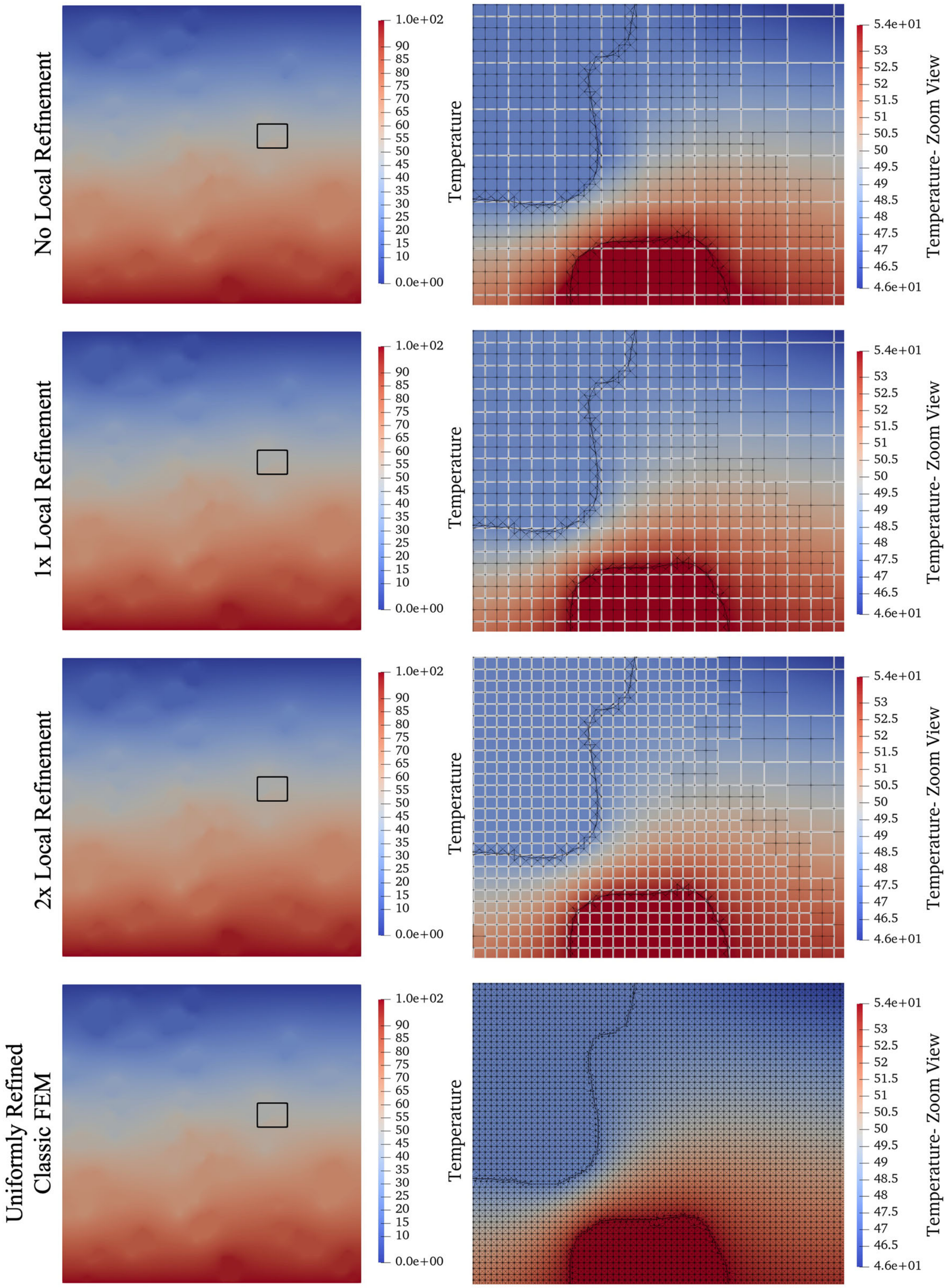


Fig. 13 The temperature field results are compared for four different mesh configurations. The left shows the entire domain with the box indicating the region shown in the zoomed in view on the right

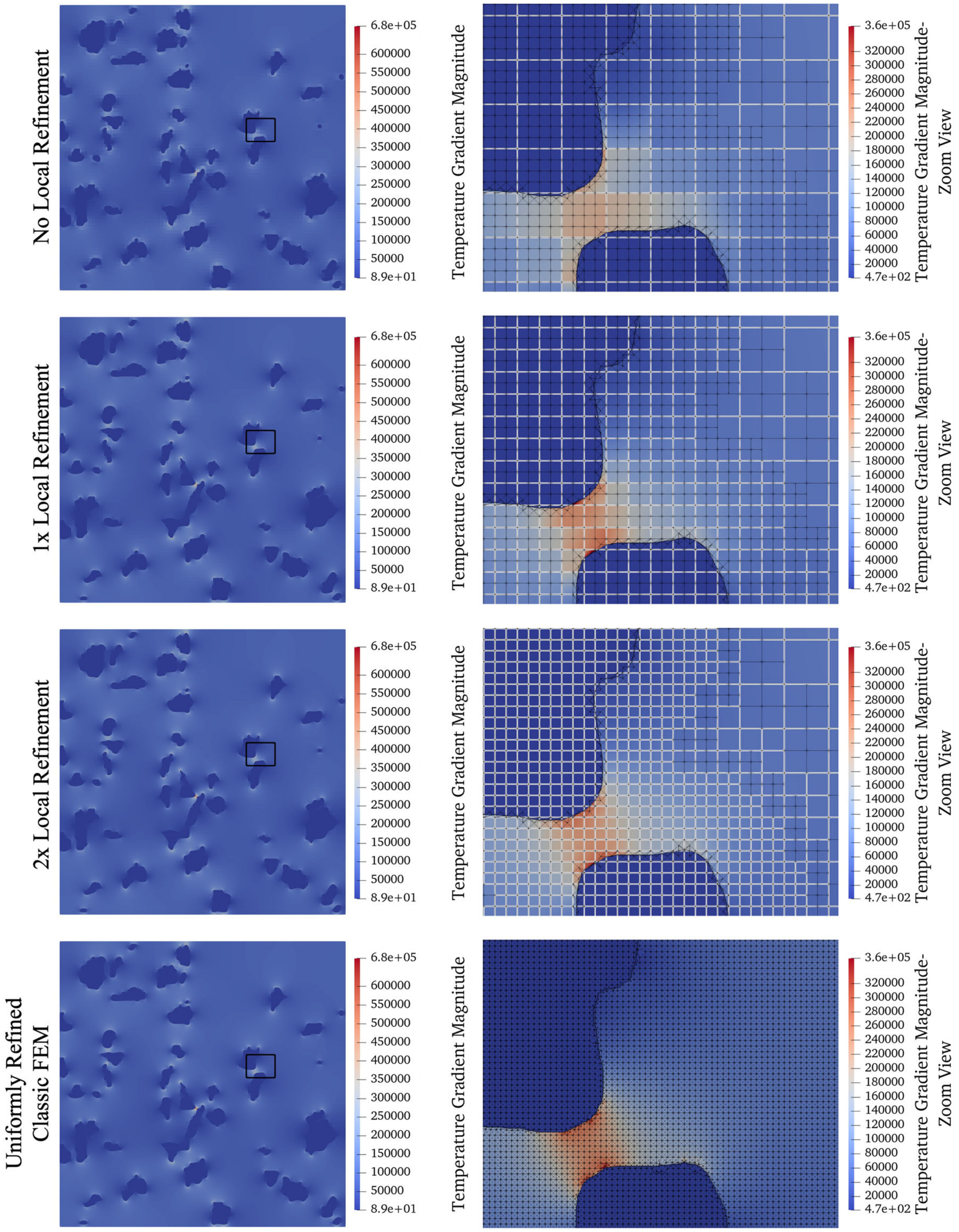


Fig. 14 The the temperature gradient magnitude field is compared for four different mesh configurations. The left shows the entire domain with the box indicating the region shown in the zoomed in view on the right

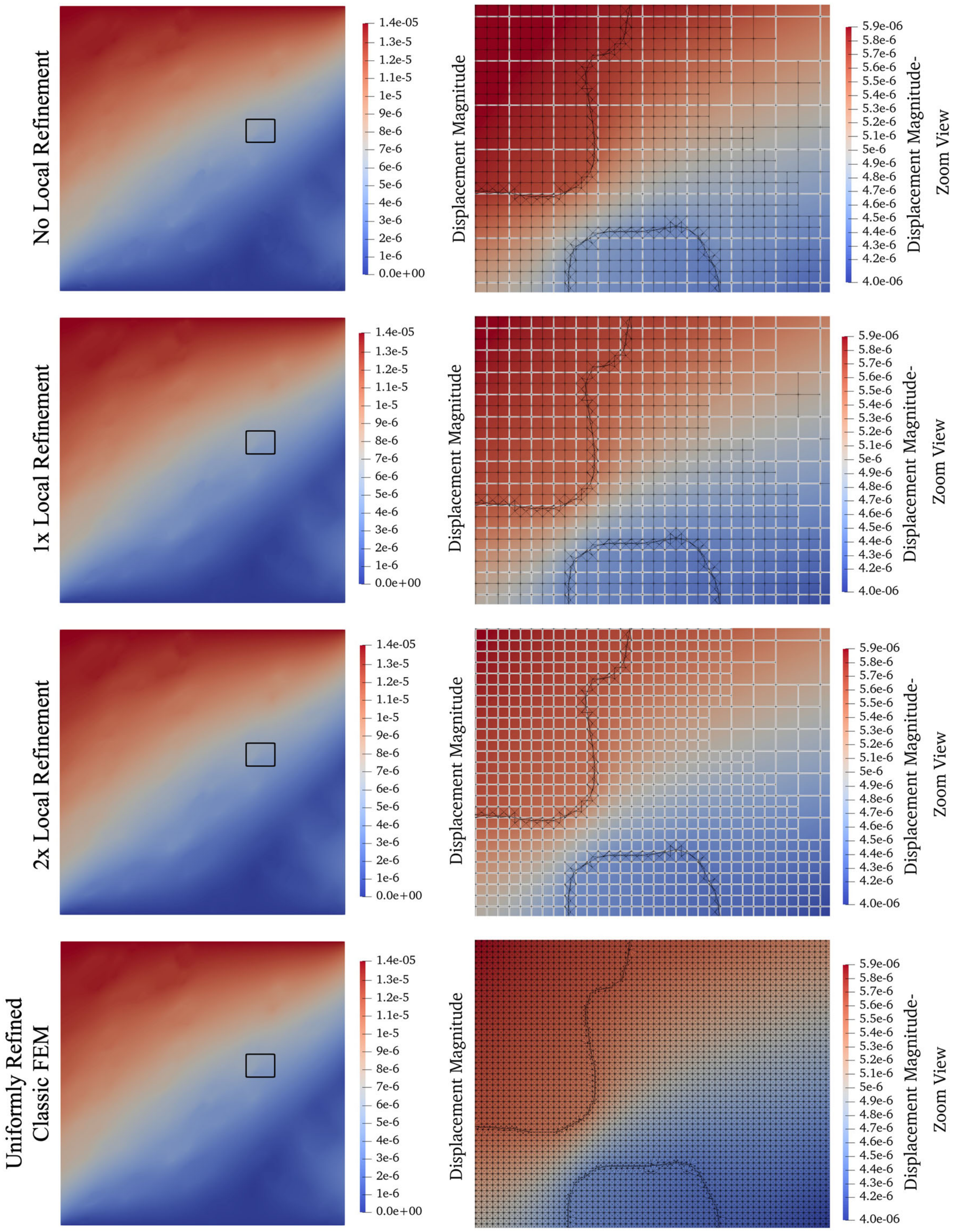


Fig. 15 The displacement magnitude field results are compared for four different mesh configurations. The left shows the entire domain with the box indicating the region shown in the zoomed in view on the right

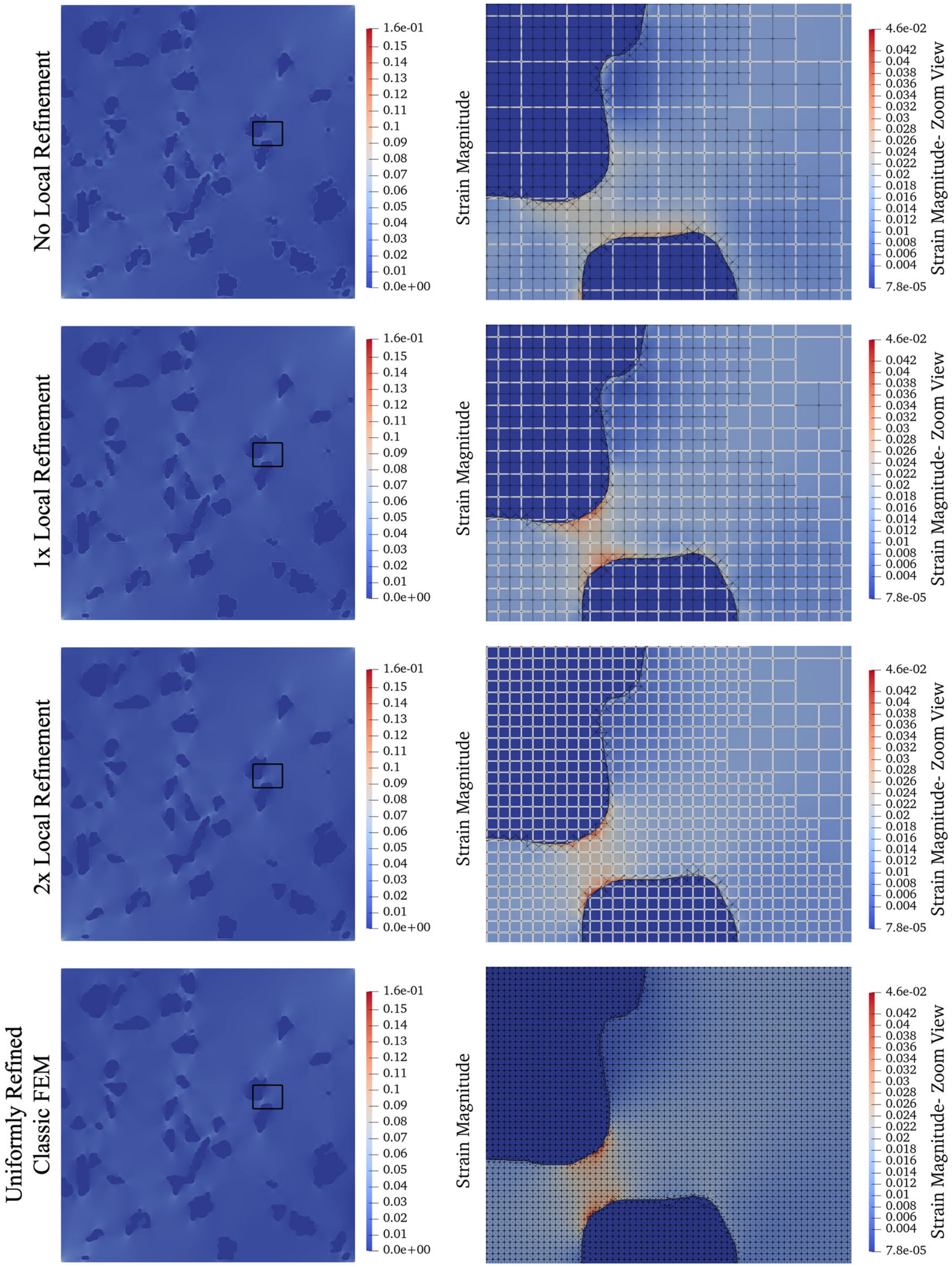


Fig. 16 The mechanical strain magnitude field results are compared for four different mesh configurations. The left shows the entire domain with the box indicating the region shown in the zoomed in view on the right

Table 1 Degrees of freedom (DOFs) associated with the background function spaces at various levels of local refinement and with the uniformly refined FEM function spaces

	No Local Refinement	1x Local Refinement	2x Local Refinement	Uniformly Refined Boundary-Fitted Mesh
Bi-linear basis DOFs	8321	17,247	48,191	5,027,580
Bi-quadratic basis DOFs	9381	22,166	60,729	10,055,160

ary conforming foreground meshes, a ‘hands-off’ method requiring only a bitmap image file. A more efficient mesh with fewer elements could be created for use in classical FEM but only with either significant user intervention or with sophisticated meshing software. Additionally, the workflow used here can easily be extended to 3D image stacks, which are not easily processed for classical FEM.

6 Conclusions

This work presents a new interpolation-based immersed boundary method for multi-material and multi-physics problems. This method employs enriched truncated hierarchically refined B-spline background spaces and discontinuous hierarchically refined Lagrange integration spaces.

Domain geometry and material interfaces are represented by level set functions, which can be generated by, for example, geometric primitives or from 2D or 3D images. The domain is embedded in a grid with an associated B-spline basis and the level set function is discretized and used to compute Heaviside characteristic functions. Basis functions are inspected and individually enriched with respect to the disconnected material subregions within their supports.

Subdivision is used to generate a sequence of refined B-spline function spaces and associated tensor-product meshes from the original function space and mesh. In this work, the meshes are refined along domain interfaces. A hierarchically refined B-spline function space is defined recursively, along with its associated hierarchically refined mesh. This hierarchical function space is then truncated to form a partition of unity. The refined basis functions can then be enriched using the Heaviside enrichment functions when intersecting material interfaces.

The enriched and refined background bases are interpolated by a discontinuous foreground basis. This foreground basis requires a boundary-fitted mesh, but is not subject to usual mesh conditioning constraints. The foreground meshes are thus constructed by the discretized level set function boundary descriptions and hierarchically refined quadrilateral meshes. Cells in the refined quadrilateral mesh intersected by the domain and material interface geometry are triangulated to form a mixed-element type foreground mesh, which can be used by classical finite element codes to define a foreground basis. The level of refinement used to create the foreground mesh may exceed the level of refinement used for the background basis, allowing for greater geometric resolution without increasing the system’s degrees of freedom associated with the background basis.

The background basis is interpolated using extraction operators. Extraction operators are constructed by evaluating the background basis at the locations of the foreground basis nodes. Existing classical finite element codes assemble the linear systems using the Lagrange foreground basis. The linear system is then projected onto the interpolated basis with the extraction operators and the system is solved with the interpolated basis. Interpolation allows the enriched and refined B-splines basis to be utilized in traditional finite element codes without the addition of complex integration subroutines, broadening the applicability of this method.

Several benchmark problems validated this method. Interpolated enriched bases were used in both 2D and 3D for a multi-material thermal diffusion problem with exact geometric representation. The numerical approximations were compared with analytic solutions and errors were computed. The L_2 and H_1 errors converged at ideal rates with mesh refinement. Foreground only refinement was implemented for a multi-material linear elasticity PDE involving a circular domain. When compared to the analytic solution, errors from a non-refined foreground mesh converged ideally with a bi-linear basis, but the L_2 convergence rate of the bi-quadratic basis was limited to 2^{nd} order due to geometric error in the discretization of the domain boundary. With sufficient foreground refinement, the ideal 3^{rd} order convergence for the bi-quadratic basis was observed. Lastly, micro-CT images were used to generate a geometric discretization of an alumina-epoxy composite sample. Thermo-elastity was simulated, coupling separately discretized temperature and displacement fields through a thermal expansion component. Unique THB-spline bases were used for each field, bi-linear for temperature, and bi-quadratic for displacement. The results were in qualitative agreement with the ones of a uniformly refined traditional boundary conforming finite element simulation.

In this work, the open-source code FEniCSx is used to demonstrate the efficacy of interpolation-based immersed boundary methods, and future work will expand implementation to other existing finite element codes. Within FEniCSx, additions to the interpolation-based immersed boundary workflow will be the implementation of stabilization techniques to address issues of linear conditioning. The method will also be further developed for additional applications, including transient and optimization problems.

Appendix A

A review of multivariate B-spline function spaces

Recall first the construction of a 1D univariate B-spline basis $\left\{B_{i,p}(\xi)\right\}_{i=1}^{n}$, where p is the polynomial order and n is the number of basis functions. The domain is discretized with a knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$ such that $\{\xi_i\}_{i=1}^{n+p+1} \subset \mathbb{R}$ and $\xi_1 \leq \xi_2 \leq \cdots \leq \xi_{n+p+1}$. The functions are then constructed recursively from the piecewise constant basis function

$$B_{i,0}(\xi) = \begin{cases} 1, & \text{if } \xi_i \leq \xi \leq \xi_{i+1} \\ 0, & \text{else} \end{cases}, \tag{A1}$$

using the Cox-de Boor recursion formula [17]

$$B_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1}(\xi). \tag{A2}$$

If no interior knots are repeated, the basis is C^{p-1} continuous at each knot in the interior domain, and C^{∞} continuous between the knots. The basis will form a partition of unity if the first and last knots are repeated $p+1$ times. Higher dimension basis functions can be constructed by applying tensor-product operations to the univariate functions, such that

$$B_{\boldsymbol{i},\boldsymbol{p}}(\boldsymbol{\xi}) = \prod_{m=1}^{d_p} B^{m}_{i_m,p_m}(\xi^m), \tag{A3}$$

where d_p is the parametric space dimension, and there are d_p knot vectors $\Xi^m = \{\xi^m_1, \xi^m_2, \ldots, \xi^m_{n_m+p_m+1}\}$, where n_m is the number of basis functions and p_m is the polynomial order of the mth parametric direction. Here $\boldsymbol{i} = \{i_1, \ldots, i_{d_p}\}$ is a multi-index and $\boldsymbol{p} = \{p_1, \ldots, p_{d_p}\}$ is the vector of polynomial degrees. The B-spline basis of the set of these functions is denoted by $\mathcal{B}_{\boldsymbol{p}} := \{B_{\boldsymbol{i},\boldsymbol{p}}\}$.

Appendix B

Discretization of multi-material equations for linear elasticity

Let a domain of n material subdomains $\overline{\Omega} = \overline{\Omega}^1 \cup \overline{\Omega}^2 \cup \cdots \cup \overline{\Omega}^n \subset \mathbb{R}^{d_p}$ be the domain of interest, with varied material properties μ and λ:

$$\mu(\boldsymbol{x}) = \mu^m, \boldsymbol{x} \in \Omega^m \text{ and} \tag{B4}$$

$$\lambda(\boldsymbol{x}) = \lambda^m, \boldsymbol{x} \in \Omega^m. \tag{B5}$$

Following the problem set up in Sect. 5.2, a source term $\boldsymbol{b} : \Omega \rightarrow \mathbb{R}^{d_p}$, a traction term $\overline{\boldsymbol{h}} : \partial\overline{\Omega} \rightarrow \mathbb{R}^{d_p}$ on $\Gamma_{\overline{\boldsymbol{h}}} \subset \partial\overline{\Omega}$ and Dirichlet boundary data $\overline{\boldsymbol{u}} : \partial\overline{\Omega} \rightarrow \mathbb{R}^{d_p}$ on $\Gamma_{\overline{\boldsymbol{u}}} \subset \partial\overline{\Omega}$ are ascribed.

The strong form of this problem is then: Find $\boldsymbol{u} : \Omega \rightarrow \mathbb{R}^{d_p}$ such that $\forall\, m \in \mathcal{M}$

$$\begin{aligned} -\nabla \cdot \boldsymbol{\sigma}^m &= \boldsymbol{b} \quad \text{in } \Omega^m, \\ [\![\boldsymbol{u}]\!] &= 0 \quad \text{on all } \Gamma_{km} \\ [\![\boldsymbol{\sigma}]\!] \cdot \boldsymbol{n} &= 0 \quad \text{on all } \Gamma_{km} \\ \boldsymbol{\sigma}^m \cdot \boldsymbol{n} &= \overline{\boldsymbol{h}} \quad \text{on } \Gamma^m_{\overline{\boldsymbol{h}}}, \\ \boldsymbol{u} &= \overline{\boldsymbol{u}} \quad \text{on } \Gamma_{\overline{\boldsymbol{u}}} \end{aligned} \tag{B6}$$

where $\Gamma_{km} = \overline{\Omega}^k \cap \overline{\Omega}^m \neq \emptyset$, $k \in \mathcal{M}$ and $k \neq m$ are the material interfaces. $\Gamma^m_{\overline{\boldsymbol{h}}} = \Gamma_{\overline{\boldsymbol{h}}} \cup \partial\overline{\Omega}^m$, and $\Gamma^m_{\overline{\boldsymbol{u}}} = \Gamma_{\overline{\boldsymbol{u}}} \cup \partial\overline{\Omega}^m$ are intersections of the domain boundaries with the material subdomain boundaries, and $\boldsymbol{n}$ denotes the surface normal. Here $[\![\cdot]\!] = (\cdot)^k - (\cdot)^k$ is again the jump of a given quantity over the Γ_{km} interface, and $\boldsymbol{n}$ is the surface normal. For each material subregion the displacement is $\boldsymbol{u}^m = \boldsymbol{u}(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega^m$. The Cauchy stress tensor in each material subdomain is defined

$$\boldsymbol{\sigma}^m = \boldsymbol{C}^m : \boldsymbol{\varepsilon}^m = 2\mu^m \boldsymbol{\varepsilon}^m + \lambda^m \mathrm{tr}(\boldsymbol{\varepsilon}^m)\boldsymbol{I} \tag{B7}$$

in terms of the strain $\boldsymbol{\varepsilon}^m$, whose definition will vary depending on application, and $\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{\sigma}^m$, $\boldsymbol{x} \in \Omega^m$.

The computational domain Ω is embedded into a hierarchically refined background mesh $\mathcal{K}_u$, generated using a sequence of refined meshes $\mathcal{K}^l$ and subdomains Ω^l_u, and associated with the THB basis $\mathcal{T}_u = \{B^u_i\}$. Each component of the displacement is then discretized with the function space

$$\mathcal{V}^h_u = \mathrm{span}\{B^u_i \,|\, \mathrm{supp}(B^u_i) \cap \Omega \neq \emptyset\}, \tag{B8}$$

where $B^u_i \in \mathcal{T}_u$, the basis of enriched THB-splines.

The discrete form is then: find $\boldsymbol{u}^h \in \boldsymbol{\mathcal{V}}_u^h = [\mathcal{V}_u^h, \mathcal{V}_u^h]$ such that all $\boldsymbol{v}^h \in \boldsymbol{\mathcal{V}}_u^h$

$$\sum_{m=1}^{n}\left[\int_{\Omega}^{m} \boldsymbol{\sigma}(\boldsymbol{u}^h) : \boldsymbol{\varepsilon}^u(\boldsymbol{v}^h)\, d\Omega\right] - \int_{\Omega} \boldsymbol{b}\cdot\boldsymbol{v}^h\, d\Omega - \int_{\Gamma_{\overline{h}}} \overline{\boldsymbol{h}}\cdot\boldsymbol{v}^h\, d\Gamma \; = \mathcal{R}_u^D + \mathcal{R}_u^I\,, \tag{B9}$$

where $\mathcal{R}_u^D$ and $\mathcal{R}_u^I$ are the Dirichlet and interface residuals,

$$\mathcal{R}_u^D = \sum_{m=1}^{n}\left[\mp\int_{\Gamma_{\overline{u}}^m} (\boldsymbol{u}^h - \overline{\boldsymbol{u}})\cdot\boldsymbol{\sigma}(\boldsymbol{v}^h)\cdot\boldsymbol{n}\, d\Gamma - \int_{\Gamma_{\overline{u}}^m} \boldsymbol{v}^h\cdot\boldsymbol{\sigma}(\boldsymbol{u}^h)\cdot\boldsymbol{n}\, d\Gamma + \int_{\Gamma_{\overline{u}}^m} \frac{\beta_u^D E}{h}(\boldsymbol{u}^h - \overline{\boldsymbol{u}})\cdot\boldsymbol{v}^h\, d\Gamma\right] \tag{B10}$$

and

$$\mathcal{R}_u^I = \sum_{i=i}^{n}\sum_{j=1+i}^{n}\left[-\int_{\Gamma_{ij}} [\![\boldsymbol{u}^h]\!]\cdot(\{\boldsymbol{\sigma}(\boldsymbol{v}^h)\}\cdot\boldsymbol{n})\, d\Gamma - \int_{\Gamma_{ij}} [\![\boldsymbol{v}^h]\!]\cdot(\{\boldsymbol{\sigma}(\boldsymbol{u}^h)\}\cdot\boldsymbol{n})\, d\Gamma + \int_{\Gamma_{ij}} \gamma_u^{ij} [\![\boldsymbol{u}^h]\!]\cdot[\![\boldsymbol{v}^h]\!]\, d\Gamma\right]. \tag{B11}$$

As with the temperature Dirichlet residual, the first line of Eq. (B10) is either negative for symmetric Nitsche's method, which is used in this work, or positive for non-symmmetric Nitsche's method. Once again, $\{\cdot\} = w_i(\cdot)_i - w_j(\cdot)_j$ is the weighted average of a given quantity, with weights as defined in Eq. (21) using the elastic modulus as the material parameter ω. The penalty parameter γ_u^{ij} is defined as

$$\gamma_u^{ij} = 2\beta_u^I \frac{(h^i)^{d_p-1} + (h^j)^{d_p-1}}{(h^i)^{d_p}/E^i + (h^j)^{d_p}/E^j}, \tag{B12}$$

where $\beta_u^I \geq 0$ is a user specified constant which controls the accuracy of the the interface condition, and $E(\boldsymbol{x}) = E^m$, $\boldsymbol{x} \in \Omega^m$.

The linear elastic subproblem can be compactly written as the variational problem: Find $\boldsymbol{u}^h \in \boldsymbol{\mathcal{V}}_u^h = [\mathcal{V}_u^h, \mathcal{V}_u^h,]$ such that, $\forall \boldsymbol{v}^h \in \boldsymbol{\mathcal{V}}_u^h$

$$a_u(\boldsymbol{u}^h, \boldsymbol{v}^h) = L_u(\boldsymbol{v}^h), \tag{B13}$$

where $a_u(\boldsymbol{u}, \boldsymbol{u})$ and $L_u(\boldsymbol{v})$ can be derived from Eq. (B9).

To approximate the solutions to multi-material linear elasticity problems, this workflow introduces an interpolated background function space

$$\widehat{\mathcal{V}}_u^h = \operatorname{span}\{\widehat{B}_i^u \mid \operatorname{supp}(\widehat{B}_i^u) \cap \Omega \neq \emptyset\}, \tag{B14}$$

where the interpolated background basis functions are defined

$$\widehat{B}_i^u := \sum_{j=1}^{\nu} M_{ij}^u N_j, \tag{B15}$$

where

$$M_{ij}^u := B_i^u(\boldsymbol{x}_j) \tag{B16}$$

is the displacement component Lagrange extraction operator. $\{N_j\}_{j=1}^{\nu}$ is the basis of a Lagrange FE space with nodal points $\boldsymbol{x}_j$ such that $N_i(\boldsymbol{x}_j) = \delta_{ij}$. Here ν is the number of foreground basis functions. Note that for multi-physics problems, the same foreground space is used to interpolate the background bases for both the temperature and the displacements.

The approximation of each displacement component is then

$$u_k^h = \sum_{i=1}^{n_u} \widehat{B}_i^u d_i^{u_k} = \sum_{i=1}^{n_u}\sum_{j=1}^{\nu} M_{ij}^u N_j d_i^{u_k}, \tag{B17}$$

where $\{d_i^{u_k}\}_{i=1}^{n_u}$ are the unknown coefficients associated with each state variable field and n_u is the number of basis functions in the interpolated background B-spline basis. $k \in \{1, \ldots, d_p\}$ are the indices associated with each of the displacement components, with d_p denoting the physical dimension. The vector-valued approximation of displacement is defined as

$$\boldsymbol{u}^h = \sum_{k=1}^{d_p}\sum_{i=1}^{n_u} \widehat{B}_i^u d_i^{u_k} \boldsymbol{e}^k \tag{B18}$$

where $\boldsymbol{e}^k$ are the directional unit vectors. For brevity in notation, new capital letter indices $I = d_p(i-1) + k$ and $J = d_p(j-1) + k$ are defined such that the vector value basis functions are

$$\widehat{\boldsymbol{B}}_I^u = \widehat{B}_i^u \boldsymbol{e}^k \text{ and } \boldsymbol{N}_J = N_j^u \boldsymbol{e}^k, \tag{B19}$$

and the approximation of displacement can be written

$$\boldsymbol{u}^h = \sum_{I=1}^{(d_p\cdot n_u)} \widehat{\boldsymbol{B}}_I^u d_I^u = \sum_{I=1}^{(d_p\cdot n_u)}\sum_{J=1}^{(d_p\cdot\nu)} M_{IJ}^{\boldsymbol{u}} \boldsymbol{N}_J d_I^u, \tag{B20}$$

where $d_I^u = d_i^{u_k}$, and $M^{\boldsymbol{u}}_{IJ}$ are the components of the vector valued displacement field extraction operator

$$M^{\boldsymbol{u}}_{IJ} = \widehat{\boldsymbol{B}}^u_I(\boldsymbol{x}_J)\,. \tag{B21}$$

The variational form in Eq. (B13) assembled using the interpolated basis forms the linear system

$$\boldsymbol{K}^{\boldsymbol{vv}}\boldsymbol{d}^{\boldsymbol{u}} = \boldsymbol{f}^{\boldsymbol{v}}, \tag{B22}$$

where

$$K^{\boldsymbol{vv}}_{IJ} = a_{\boldsymbol{u}}(\widehat{\boldsymbol{B}}^u_I, \widehat{\boldsymbol{B}}^u_J), \text{ and} \tag{B23}$$

$$f^{\boldsymbol{v}}_I = L_{\boldsymbol{u}}(\widehat{\boldsymbol{B}}^u_I). \tag{B24}$$

Applying extraction, the linear system in Eq. (B22) is rewritten as

$$(\boldsymbol{M}^{\boldsymbol{v}})^{\mathrm{T}}\boldsymbol{A}^{\boldsymbol{vv}}\boldsymbol{d}^{\boldsymbol{u}} = (\boldsymbol{M}^{\boldsymbol{u}})^{\mathrm{T}}\boldsymbol{b}^{u}, \tag{B25}$$

where $\boldsymbol{A}^{\boldsymbol{vv}}$ and $\boldsymbol{b}^{\boldsymbol{v}}$ are computed with the foreground basis

$$A^{\boldsymbol{vv}}_{IJ} = a_{\boldsymbol{u}}(\boldsymbol{N}_I, \boldsymbol{N}_J) \text{ and} \tag{B26}$$

$$b^{\boldsymbol{v}}_I = L_{\boldsymbol{u}}(\boldsymbol{N}_I). \tag{B27}$$

The quantities $\boldsymbol{A}^{\boldsymbol{vv}}$ and $\boldsymbol{b}^{\boldsymbol{v}}$ are evaluated and assembled with the boundary-fitted foreground mesh, following the workflow outlined in Sect. 4.2.

Acknowledgements The authors acknowledge the support for this work from the National Science Foundation under Grants 2103939 and 2104106. The authors also acknowledge and thank Dr. David Kamensky for his assistance in developing the interpolation workflow.

References

1. Alnæs M, Blechta J, Hake J et al (2015) The FEniCS project version 1.5. Arch Numer Softw 3(100):9-23. https://doi.org/10.11588/ans.2015.100.20553
2. Annavarapu C, Hautefeuille M, Dolbow JE (2012) A robust Nitsche's formulation for interface problems. Comput Methods Appl Mech Eng 225–228:44–54. https://doi.org/10.1016/j.cma.2012.03.008
3. Atallah NM, Canuto C, Scovazzi G (2022) The high-order Shifted Boundary Method and its analysis. Comput Methods Appl Mech Eng 394:114885. https://doi.org/10.1016/j.cma.2022.114885
4. Babuška I, Melenk JM (1997) The Partition of Unity Method. Int J Numer Methods Eng 40(4):727–758. https://doi.org/10.1002/(SICI)1097-0207(19970228)40:4$<$727:AID-NME86$>$3.0.CO;2-N
5. Badia S, Verdugo F, Martín AF (2018) The aggregated unfitted finite element method for elliptic problems. Comput Methods Appl Mech Eng 336:533–553. https://doi.org/10.1016/j.cma.2018.03.022
6. Baratta IA, Dean JP, Dokken JS et al (2023) DOLFINx: The next generation FEniCS problem solving environment. https://doi.org/10.5281/zenodo.10447666
7. Bauccio M (1994) ASM engineered materials reference book. ASM International, New York
8. Belytschko T, Black T (1999) Elastic crack growth in finite elements with minimal remeshing. Int J Numer Methods Eng 45(5):601–620. https://doi.org/10.1002/(SICI)1097-0207(19990620)45:5<601:AID-NME598>3.0.CO;2-S
9. Belytschko T, Moës N, Usui S et al (2001) Arbitrary discontinuities in finite elements. Int J Numer Methods Eng 50(4):993–1013. https://doi.org/10.1002/1097-0207(20010210)50:4<993:AID-NME164>3.0.CO;2-M
10. Boggs PT, Althsuler A, Larzelere AR et al (2005) DART system analysis. Technical Report, SAND2005-4647, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA (United States). https://doi.org/10.2172/876325
11. Borden MJ, Scott MA, Evans JA et al (2011) Isogeometric finite element data structures based on Bézier extraction of NURBS. Int J Numer Methods Eng 87(1–5):15–47. https://doi.org/10.1002/nme.2968
12. Burkhart TA, Andrews DM, Dunning CE (2013) Finite element modeling mesh quality, energy balance and validation methods: a review with recommendations associated with the modeling of bone tissue. J Biomech 46(9):1477–1488. https://doi.org/10.1016/j.jbiomech.2013.03.022
13. Burman E (2010) Ghost penalty. CR Math 348(21):1217–1220. https://doi.org/10.1016/j.crma.2010.10.006
14. Burman E, Hansbo P, Larson MG et al (2023) Extension operators for trimmed spline spaces. Comput Methods Appl Mech Eng 403:115707. https://doi.org/10.1016/j.cma.2022.115707
15. Cheng KW, Fries TP (2010) Higher-order XFEM for curved strong and weak discontinuities. Int J Numer Methods Eng 82(5):564–590. https://doi.org/10.1002/nme.2768
16. Cockburn B (2003) Discontinuous Galerkin methods. ZAMM J Appl Math Mech/Zeitschrift für Angewandte Mathematik und Mechanik 83(11):731–754. https://doi.org/10.1002/zamm.200310088
17. de Boor C (1972) On calculating with B-splines. J Approx Theory 6(1):50–62. https://doi.org/10.1016/0021-9045(72)90080-9
18. de Prenter F, Verhoosel CV, van Brummelen EH et al (2023) Stability and conditioning of immersed finite element methods: analysis and remedies. Arch Computat Methods Eng 30(6):3617–3656. https://doi.org/10.1007/s11831-023-09913-0
19. Divi SC, Verhoosel CV, Auricchio F et al (2020) Error-estimate-based adaptive integration for immersed isogeometric analysis. Comput Math Appl 80(11):2481–2516. https://doi.org/10.1016/j.camwa.2020.03.026
20. Düster A, Parvizian J, Yang Z et al (2008) The finite cell method for three-dimensional problems of solid mechanics. Comput Methods Appl Mech Eng 197(45):3768–3782. https://doi.org/10.1016/j.cma.2008.02.036
21. Elfverson D, Larson MG, Larsson K (2018) CutIGA with basis function removal. Adv Model Simul Eng Sci 5(1):6. https://doi.org/10.1186/s40323-018-0099-2

22. Engvall L, Evans JA (2020) Mesh quality metrics for isogeometric Bernstein–Bézier discretizations. Comput Methods Appl Mech Eng 371:113305. https://doi.org/10.1016/j.cma.2020.113305
23. Evans JA, Bazilevs Y, Babuška I et al (2009) n-widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. Comput Methods Appl Mech Eng 198(21–29):1726–1741. https://doi.org/10.1016/j.cma.2009.01.021
24. Fries TP, Omerović S (2016) Higher-order accurate integration of implicit geometries. Int J Numer Methods Eng 106(5):323–371. https://doi.org/10.1016/j.cma.2016.10.019
25. Fromm JE (2024) jefromm/EXHUME_dolfinx. https://github.com/jefromm/EXHUME_dolfinX
26. Fromm JE, Wunsch N, Xiang R et al (2023) Interpolation-based immersed finite element and isogeometric analysis. Comput Methods Appl Mech Eng 405:115890. https://doi.org/10.1016/j.cma.2023.115890
27. Garau EM, Vázquez R (2018) Algorithms for the implementation of adaptive isogeometric methods using hierarchical B-splines. Appl Numer Math 123:58–87. https://doi.org/10.1016/j.apnum.2017.08.006
28. Giannelli C, Jüttler B, Speleers H (2012) THB-splines: the truncated basis for hierarchical splines. Comput Aided Geometric Des 29(7):485–498. https://doi.org/10.1016/j.cagd.2012.03.025
29. Gunderman D, Weiss K, Evans JA (2021) High-accuracy mesh-free quadrature for trimmed parametric surfaces and volumes. Comput Aided Des 141:103093. https://doi.org/10.1016/j.cad.2021.103093
30. Gunderman D, Weiss K, Evans JA (2021) Spectral mesh-free quadrature for planar regions bounded by rational parametric curves. Comput Aided Des 130:102944. https://doi.org/10.48550/arXiv.2005.07780
31. Hansbo A, Hansbo P (2004) A finite element method for the simulation of strong and weak discontinuities in solid mechanics. Comput Methods Appl Mech Eng 193(33):3523–3540. https://doi.org/10.1016/j.cma.2003.12.041
32. Huang TH, Chen JS, Tupek MR et al (2022) A variational multiscale immersed meshfree method for fluid structure interactive systems involving shock waves. Comput Methods Appl Mech Eng 389:114396. https://doi.org/10.1016/j.cma.2021.114396
33. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 194(39):4135–4195. https://doi.org/10.1016/j.cma.2004.10.008
34. Hughes TJR, Reali A, Sangalli G (2008) Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method NURBS. Comput Methods Appl Mech Eng 197(49):4104–4124. https://doi.org/10.1016/j.cma.2008.04.006
35. Hughes TJR, Evans JA, Reali A (2014) Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems. Comput Methods Appl Mech Eng 272:290–320. https://doi.org/10.1016/j.cma.2013.11.012
36. Johansson A, Larson MG, Logg A (2020) Multimesh finite elements with flexible mesh sizes. Comput Methods Appl Mech Eng 372:113420. https://doi.org/10.1016/j.cma.2020.113420
37. Kamensky D (2021) Open-source immersogeometric analysis of fluid-structure interaction using FEniCS and tIGAr. Comput Math Appl 81:634–648. https://doi.org/10.1016/j.camwa.2020.01.023
38. Kamensky D, Bazilevs Y (2019) tIGAr: Automating isogeometric analysis with FEniCS. Comput Methods Appl Mech Eng 344:477–498. https://doi.org/10.1016/j.cma.2018.10.002
39. Kamensky D, Hsu MC, Schillinger D et al (2015) An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves. Comput Methods Appl Mech Eng 284:1005–1053. https://doi.org/10.1016/j.cma.2014.10.040
40. Knupp P (2007) Remarks on mesh quality. In: 46th AIAA aerospace sciences meeting and exhibit. https://www.semanticscholar.org/paper/Remarks-on-Mesh-Quality.-Knupp/4a229d7e0a7a2cdd5df2d2c0e57dee83ba397bbe
41. Knupp PM (2001) Algebraic mesh quality metrics. SIAM J Sci Comput 23(1):193–218. https://doi.org/10.1137/S1064827500371499
42. Main A, Scovazzi G (2018) The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems. J Comput Phys 372:972–995. https://doi.org/10.1016/j.jcp.2017.10.026
43. Main A, Scovazzi G (2018) The shifted boundary method for embedded domain computations. Part II: linear advection-diffusion and incompressible Navier-Stokes equations. J Comput Phys 372:996–1026. https://doi.org/10.1016/j.jcp.2018.01.023
44. MatWeb (2024) Overview of materials for Epoxy Cure Resin. https://www.matweb.com/search/datasheet_print.aspx?matguid=956da5edc80f4c62a72c15ca2b923494
45. Maute K (2023) MORIS. https://github.com/kkmaute/moris
46. Min C, Gibou F (2007) Geometric integration over irregular domains with application to level-set methods. J Comput Phys 226(2):1432–1443. https://doi.org/10.1016/j.jcp.2007.05.032 (https://www.sciencedirect.com/science/article/pii/S0021999107002410)
47. Moutsanidis G, Li W, Bazilevs Y (2021) Reduced quadrature for FEM, IGA and meshfree methods. Comput Methods Appl Mech Eng 373:113521. https://doi.org/10.1016/j.cma.2020.113521
48. Moës N, Dolbow J, Belytschko T (1999) A finite element method for crack growth without remeshing. Int J Numer Methods Eng 46(1):131–150. https://doi.org/10.1002/(SICI)1097-0207(19990910)46:1<131:AID-NME726>3.0.CO;2-J
49. Müller B, Kummer F, Oberlack M (2013) Highly accurate surface and volume integration on implicit domains by means of moment-fitting. Int J Numer Methods Eng 96(8):512–528. https://doi.org/10.1002/nme.4569
50. Nazir A, Gokcekaya O, Md Masum Billah K et al (2023) Multi-material additive manufacturing: a systematic review of design, properties, applications, challenges, and 3D printing of materials and cellular metamaterials. Mater Des 226:111661. https://doi.org/10.1016/j.matdes.2023.111661
51. Nitsche J (1971) Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. AbhMathSeminUnivHambg 36(1):9–15. https://doi.org/10.1007/BF02995904
52. Noël L, Schmidt M, Doble K et al (2022) XIGA: an eXtended IsoGeometric analysis approach for multi-material problems. Comput Mech 70(6):1281–1308. https://doi.org/10.1007/s00466-022-02200-y
53. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. J Comput Phys 79(1):12–49. https://doi.org/10.1016/0021-9991(88)90002-2
54. Parvizian J, Düster A, Rank E (2007) Finite cell method. Comput Mech 41(1):121–133. https://doi.org/10.1007/s00466-007-0173-y
55. Peskin CS (1972) Flow patterns around heart valves: a numerical method. J Comput Phys 10(2):252–271. https://doi.org/10.1016/0021-9991(72)90065-4
56. Rajak DK, Pagar DD, Menezes PL et al (2019) Fiber-reinforced polymer composites: manufacturing, properties, and applications. Polymers 11(10):1667. https://doi.org/10.3390/polym11101667
57. Schillinger D, Ruess M (2015) The finite cell method: a review in the context of higher-order structural analysis of CAD and image-based geometric models. Arch Computat Methods Eng 22(3):391–455. https://doi.org/10.1007/s11831-014-9115-y

58. Schillinger D, Dedè L, Scott MA et al (2012) An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. Comput Methods Appl Mech Eng 249–252:116–150. https://doi.org/10.1016/j.cma.2012.03.017
59. Schillinger D, Ruthala PK, Nguyen LH (2016) Lagrange extraction and projection for NURBS basis functions: A direct link between isogeometric and standard nodal finite element formulations. Int J Numer Meth Eng 108(6):515–534. https://doi.org/10.1002/nme.5216
60. Schlinkman RT, Baek J, Beckwith FN et al (2023) A quasi-conforming embedded reproducing kernel particle method for heterogeneous materials. https://doi.org/10.48550/arXiv.2304.06150
61. Schmidt M, Noël L, Doble K et al (2023) Extended isogeometric analysis of multi-material and multi-physics problems using hierarchical B-splines. Comput Mech 71(6):1179–1203. https://doi.org/10.1007/s00466-023-02306-x
62. Soghrati S (2014) Hierarchical interface-enriched finite element method: an automated technique for mesh-independent simulations. J Comput Phys 275:41–52. https://doi.org/10.1016/j.jcp.2014.06.016
63. Sommariva A, Vianello M (2009) Gauss-green cubature and moment computation over arbitrary geometries. J Comput Appl Math 231(2):886–896. https://doi.org/10.1016/j.cam.2009.05.014
64. Strouboulis T, Babuška I, Copps K (2000) The design and analysis of the generalized finite element method. Comput Methods Appl Mech Eng 181(1):43–69. https://doi.org/10.1016/S0045-7825(99)00072-9
65. Strouboulis T, Copps K, Babuška I (2000) The generalized finite element method: an example of its implementation and illustration of its performance. Int J Numer Methods Eng 47(8):1401–1417. https://doi.org/10.1002/(SICI)1097-0207(20000320)47:8<1401:AID-NME835>3.0.CO;2-8
66. Sudhakar Y, Wall WA (2013) Quadrature schemes for arbitrary convex/concave volumes and integration of weak form in enriched partition of unity methods. Comput Methods Appl Mech Eng 258:39–54. https://doi.org/10.1016/j.cma.2013.01.007
67. Terada K, Asai M, Yamagishi M (2003) Finite cover method for linear and non-linear analyses of heterogeneous solids. Int J Numer Methods Eng 58(9):1321–1346. https://doi.org/10.1002/nme.820
68. Tirvaudey M, Bouclier R, Passieux JC et al (2019) Non-invasive implementation of nonlinear isogeometric analysis in an industrial FE software. Eng Comput 37(1):237–261. https://doi.org/10.1108/EC-03-2019-0108
69. Vese LA, Chan TF (2002) A multiphase level set framework for image segmentation using the Mumford and Shah model. Int J Comput Vis 50(3):271–293. https://doi.org/10.1023/A:1020874308076
70. Wang D, Chen JS, Sun L (2003) Homogenization of magnetostrictive particle-filled elastomers using an interface-enriched reproducing kernel particle method. Finite Elem Anal Des 39(8):765–782. https://doi.org/10.1016/S0168-874X(03)00058-1
71. Wang Y, Baek J, Tang Y et al (2023) Support vector machine guided reproducing kernel particle method for image-based modeling of microstructures. Comput Mech. https://doi.org/10.1007/s00466-023-02394-9
72. Zhu Q, Yan J (2021) A mixed interface-capturing/interface-tracking formulation for thermal multi-phase flows with emphasis on metal additive manufacturing processes. Comput Methods Appl Mech Eng 383:113910. https://doi.org/10.1016/j.cma.2021.113910