Computational Mechanics (2023) 71:1179–1203
https://doi.org/10.1007/s00466-023-02306-x

ORIGINAL PAPER

Extended isogeometric analysis of multi-material and multi-physics problems using hierarchical B-splines

Mathias Schmidt[1] · Lise Noël[2] · Keenan Doble[3] · John A. Evans[3] · Kurt Maute[3]

Received: 29 November 2022 / Accepted: 17 February 2023 / Published online: 20 March 2023

Abstract
This paper presents an immersed, isogeometric finite element framework to predict the response of multi-material, multi-physics problems with complex geometries using locally refined discretizations. To circumvent the need to generate conformal meshes, this work uses an extended finite element method (XFEM) to discretize the governing equations on non-conforming, embedding meshes. A flexible approach to create truncated hierarchical B-splines discretizations is presented. This approach enables the refinement of each state variable field individually to meet field-specific accuracy requirements. To obtain an immersed geometry representation that is consistent across all hierarchically refined B-spline discretizations, the geometry is immersed into a single mesh, the XFEM background mesh, which is constructed from the union of all hierarchical B-spline meshes. An extraction operator is introduced to represent the truncated hierarchical B-spline bases in terms of Lagrange shape functions on the XFEM background mesh without loss of accuracy. The truncated hierarchical B-spline bases are enriched using a generalized Heaviside enrichment strategy to accommodate small geometric features and multi-material problems. The governing equations are augmented by a formulation of the face-oriented ghost stabilization enhanced for locally refined B-spline bases. We present examples for two- and three-dimensional linear elastic and thermo-elastic problems. The numerical results validate the accuracy of our framework. The results also demonstrate the applicability of the proposed framework to large, geometrically complex problems.

Keywords Immersed finite element method · Extended isogeometric analysis · Multi-material problems · Multi-physics problems · Truncated hierarchical B-splines · Lagrange extraction

1 Introduction

Finite element analysis is frequently used to predict the response of systems described by partial differential equations defined over a spatial domain. In classical finite element methods, the domain is discretized using a single mesh that conforms to the external boundaries and internal material interfaces. For problems with complex shapes and multiple material phases, the construction of this conformal mesh is often a major bottleneck in the analysis process, see Bazilevs et al. [3]. Conformal mesh generation may also hamper the automation of the finite element analysis for problems with changing geometry as encountered in, for example, phase-change problems with dynamically evolving interfaces or shape and topology optimization.

The state variable fields may exhibit large spatial gradients at boundaries and material interfaces, as well as in the vicinity of small geometric features. To resolve these spatial gradients, a sufficiently fine discretization is needed. While uniformly fine discretization may yield an accurate approximation, the associated computational cost may exceed practical limits. A locally refined mesh balances discretization needs and computational cost. However, generating locally refined meshes for standard finite element methods further increases the complexity of mesh generation.

✉ Mathias Schmidt
schmidt43@llnl.gov

[1] Computational Engineering Division, Lawrence Livermore National Laboratory, 7000 East Ave, Livermore, CA 94550, USA

[2] Department of Precision and Microsystems Engineering, Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

[3] Department of Aerospace Engineering Sciences, Aerospace Mechanics Research Center, University of Colorado Boulder, 3775 Discovery Dr, Boulder, CO 80309-0429, USA

Many engineering and science applications involve multiple, often coupled state variable fields. For such problems, each field may require differently refined discretizations. For example, consider the two-material, thermo-elastic problem of a circle embedded in a rectangular plate shown in Fig. 1a. We assume that the circle is made of a material with a finite coefficient of thermal expansion (CTE), while the plate's CTE is zero. The plate is subject to a spatially varying heat flux along its right edge and clamped at its left edge. The heat flux increases the temperature in the system, causing the circle to expand which generates stresses in both the plate and circle due to the CTE mismatch. To predict the stress field at the material interface with high accuracy, accurate approximations of both the temperature and displacement fields are required. The contour of the norm of the diffusive flux is shown in Fig. 1b and the contour of the Von Mises stress field in Fig. 1c. The regions with large spatial gradients differ between the displacement and the temperature field. Separately adapting the discretizations associated with each field enables a sufficient resulution of each field while minimizing the overall computational cost. An example of such discretizations for the temperature and displacement fields is depicted in Fig. 2a, b, respectively.

Lagrange polynomials are the standard choice for basis functions in finite elements. The element-local nature of this class of basis functions and their interpolatory property facilitate a convenient and efficient implementation and enforcement of Dirichlet boundary and interface conditions. However, Lagrange basis functions limit the inter-element continuity to C^0, irrespective of the polynomial order of the approximation within the element. The lack of higher-order inter-element continuity affects the efficiency of Lagrange bases, measured by the number of degrees of freedom needed to achieve a desired accuracy; see for example Evans et al. [20].

In this work, we propose a computational framework that addresses the issues encountered with standard finite element approaches as outlined above. This framework synthesizes the following concepts which will be further discussed in detail below. An immersed finite element approach is adopted to eliminate the need for conformal mesh generation. Higher order B-spline basis functions are used to increase the discretization efficiency over Lagrange bases. For each state variable, a truncated hierarchically refined B-spline basis is generated to meet field-specific accuracy requirements.

Immersed Boundary Methods (IBMs) have gained in popularity in recent years, see Babuška and Melenk [2], Peskin [38], Mittal and Iaccarino [32]. The general idea of these methods is to immerse the geometry of the physical domain into a computational domain with a much simpler geometry. Thus, the geometries of the physical and computational domains are decoupled. The geometric simplicity of the computational domain allows for the convenient generation of IBM background meshes, such as tensor grids. The geometry is immersed into the IBM background mesh, which simplifies the discretization of the state variable fields.

In this work, we focus on finite element formulations of IBMs. Immersed Finite Element Methods (IFEMs), also referred to as geometrically unfitted or embedded domain finite element methods, include the Finite Cell Method, see Parvizian et al. [37], Düster et al. [16], Schillinger and Ruess [43], the CutFEM [39], the Generalized Finite Element Method (GFEM) [15,47], and the eXtended Finite Element Method (XFEM), see Belytschko and Black [4], Belytschko et al. [5]. In this paper, we consider specifically the latter approach. The XFEM augments the standard finite element basis with additional basis functions to represent discontinuities of the state variable field within an XFEM background element that is intersected by a boundary or an interface. The augmented finite element basis satisfies the partition of unity (PU) property. In this work, we adopt a Heaviside enrichment strategy for its flexibility in modeling interface and boundary conditions of multi-material problems with complex geometries, see Noël et al. [36].

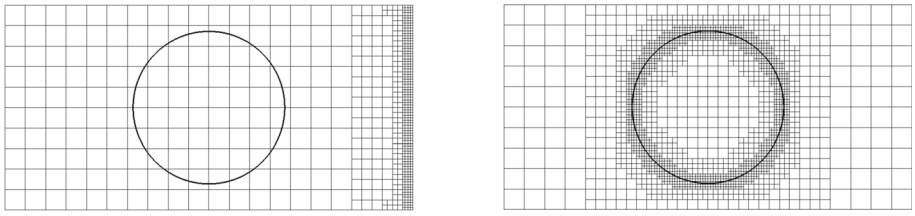
Traditionally, Heaviside enriched XFEM approaches discretize state variable fields by Lagrange basis functions which are defined on the XFEM background mesh. In this paper, we adopt higher order B-splines for discretizing state variable fields. The advantages of B-splines basis functions for finite element methods have been demonstrated in the context of Isogeometric Analysis (IGA), see Hughes et al. [26]. While IGA was originally developed to eliminate the discrepancy between CAD geometry representation and finite element analysis, Evans et al. [20] showed that B-spline

(a) Thermo-elastic multi-material problem subject to a spatially varying heat load

(b) Norm of the diffusive flux vector

(c) Von Mises stress field

Fig. 1 Illustration of an example thermo-elastic problem



(a) Locally refined mesh for the temperature field

(b) Locally refined mesh for the displacement field

Fig. 2 Example of locally refined discretizations for the thermo-elastic problem depicted in Fig. 1a

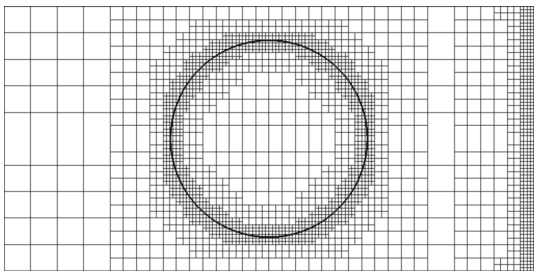


Fig. 3 Union of the locally refined meshes depicted in Fig. 2a, b

basis functions in general improve accuracy, robustness, and computational efficiency. In this work, we further utilize the refineability property of B-splines, see Garau and Vázquez [21].

B-spline basis functions were studied with IFEMs by Schillinger et al. [44], Schillinger and Rank [42], Schillinger and Ruess [43], Verhoosel et al. [50], Elfverson et al. [18], Divi et al. [14], Noël et al. [35]. Nguyen [33] introduced B-spline approximations to the XFEM and coined the term *X-IGA*. These works demonstrated that integrating higher order B-splines discretizations into IFEM approaches yields improved accuracy and computational efficiency compared to the use of Lagrange basis functions.

The ability to locally refine B-splines enables the convenient construction of locally refined approximation spaces, see Giannelli et al. [23], Buffa and Giannelli [7], Bracco et al. [6]. While standard hierarchical B-spline bases do not fulfill the PU property, truncating the bases restores the PU property [23]. Truncated Hierarchical B-spline (THB) basis functions form a sparse, strongly stable basis [23] and are employed in this work.

This paper contributes to the work on IFEMs as follows. We introduce a versatile discretization approach based on THBs defined on locally refined tensor meshes. Each state variable can be discretized individually by choosing the polynomial order and the local refinement independently from the discretization of other fields. This approach leads to an explicit control over the accuracy of the field approximations and the computational cost of the analysis. Computationally efficient algorithms and data structures are introduced to enable the application of this discretization approach to large-scale problems in two and three dimensions using parallel computing. To integrate the locally refined B-spline discretization approach into the XFEM, a single union mesh is generated from the individual B-spline meshes. Figure 3 shows the union mesh generated for the two locally refined meshes presented in Fig. 2. The union mesh serves as the XFEM background mesh in which the geometry is immersed. To increase the geometry resolution, the XFEM background mesh can be additionally refined. We introduce an exact extraction operator that expresses THBs in terms of Lagrange basis functions over each element of the XFEM background mesh.

The enrichment strategy of Makhija and Maute [30], Villanueva and Maute [51], Noël et al. [36] is generalized to enrich the locally refined THB basis functions considering their support for a given intersection geometry. The XFEM problem is augmented by an X-IGA formulation of the face-oriented ghost stabilization from Noël et al. [36]. The governing equations are integrated by standard quadrature rules on a geometry-conforming integration mesh that is constructed by cutting XFEM background elements that are intersected by the external boundaries or internal material interfaces. Boundary and interface conditions are enforced weakly by Nitsche's method. We illustrate the main characteristics and features of the proposed XFEM framework by numerical examples considering linear elastic and thermo-elastic problems. We study the convergence of geometric and state variable discretization errors with uniform and local mesh refinement for different B-spline orders. To demonstrate the applicability of our XFEM framework to complex multi-material, multi-physics problems, we apply our XFEM framework to the thermo-elastic analysis of a 3D polycrystalline micro-structure.

This paper is organized as follows: Sect. 2 outlines the Heaviside-enriched XFEM framework. Section 3 recalls the fundamentals of hierarchical B-splines as well as their truncation to restore the PU property. Section 4 details the meshing algorithms and data structures. Section 5 summarizes the governing equations, Nitsche's formulation, and face-oriented ghost stabilization for linear thermo-elasticity. Numerical two and three dimensional examples are presented in Sect. 6. The main findings are summarized in Sect. 7, together with recommendations for future work.

2 The extended finite element method

In this work, we adopt the XFEM to perform analysis on non-conforming background meshes. We follow the basic concepts of a generalized Heaviside enrichment strategy, introduced by Terada et al. [48]. In this section, the basic concepts of the XFEM as relevant for this work are briefly described.

We start from a non-conforming finite element approximation space with THB basis functions $\left\{B_j\right\}_{j=1}^{\tilde{n}}$ that are defined on a locally refined discretization. To approximate the state variable fields in different material phases, each basis function is enriched. The enrichment of a particular basis function depends on the number of topologically disconnected regions of all material phases in the support of this basis function. The approach is illustrated in Fig. 4 for a configuration with two material phases, Ω^1 and Ω^2. The support of the basis function B is depicted by the dashed red line. It spans three topologically disconnected regions, each occupied by one of the two material phases. Therefore, the basis function is enriched three times.

This approach is applicable to any number of material phases, intersection configurations, and basis function support. To define the enriched basis, we introduce the indicator function ψ_j^ℓ. This function is equal to 1 for points located within the portion of the support of the jth basis function corresponding to the enrichment level ℓ and equal to zero elsewhere. The enriched basis then is simply $\left\{B_j\psi_j^\ell : j \in \{1,\dots,\tilde{n}\} \text{ and } \ell \in \left\{1,\dots,L_j\right\}\right\}$ where L_j is the maximum number of enrichment levels for basis function B_j. Since the original THB basis satisfies the PU principle, so does the enriched THB basis.

The enriched finite element approximation of a vector-valued state variable $\mathbf{u}^h(\mathbf{x})$ can be written in terms of the enriched THB basis as follows:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{j=1}^{\tilde{n}} \left(\sum_{l=1}^{L_j} \psi_j^\ell(\mathbf{x})\, B_j(\mathbf{x})\, \mathbf{c}_j^\ell \right), \tag{1}$$

where $\mathbf{c}_j^\ell$ is the coefficient associated to the jth original THB basis function and the jth enrichment level. The indicator function ψ_j^ℓ enforces that only one set of enriched basis functions is used to describe the state variable at each point in the domain. A more detailed description of the enrichment strategy can be found in Noël et al. [36].

The Heaviside enriched XFEM formulation outlined above enables the modeling of C^{-1} intra-element discontinuities of state variables within a non-conforming background element. Essential boundary conditions can be enforced weakly by, for example, Nitsche's method [9,34] or the stabilized Lagrange multiplier method [22].

Immersing geometry into the XFEM background mesh can result in basis functions with small support within the geometric domain, leading to poorly conditioned systems of discretized governing equations. Various strategies to mitigate this issue have been studied in the literature, such as the face-oriented ghost stabilization [8,10,36], basis function removal [18,19], and pre-conditioning [13,29]. In this work, we extend the face-oriented ghost stabilization to hierarchically refined B-spline discretizations; as further described in Sect. 5.2.

3 Hierarchical B-splines

This section focuses on hierarchical B-splines for a locally refined discretization of state variable fields. First, the basic concepts of B-splines in one and multiple dimensions are recalled. Then the B-spline refinement and the construction of non-truncated (HB) and truncated hierarchical B-spline (THB) bases are described.

3.1 B-spline basis functions

In 1D, we define a knot vector $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$, for which $\xi \in \mathbb{R}$ and $\xi_1 \le \xi_2 \le \cdots \le \xi_{n+p+1}$. A univariate B-spline basis function $N_{i,p}(\xi)$ of degree p is constructed recursively starting from the piecewise constant basis function:

$$N_{i,0}(\xi) = \begin{cases} 1, & \text{if } \xi_i \le \xi \le \xi_{i+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

and using the Cox de Boor recursion formula [12] for higher degrees, $p > 0$:

$$\begin{aligned} N_{i,p}(\xi) = {} & \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) \\ & + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \end{aligned} \tag{3}$$

To guarantee a C^{p-1} continuity over the entire computational domain, none of the interior knots should be repeated. The corresponding B-spline basis exhibits a C^{p-1} continuity at every knot in the interior of the domain, while it is C^∞ continuous in between the knots. A knot span is defined as the half open interval $[\xi_i, \xi_{i+1})$ and a B-spline element is defined as a non-empty knot span.

In 2D and 3D, tensor-product B-spline basis functions $B_{i,p}(\xi)$ are constructed by applying the tensor-product operation to univariate B-spline basis functions in each parametric direction. Denoting the parametric space dimension as d_p, a tensor-product B-spline basis is constructed starting from d_p knot vectors $\Xi^m = \{\xi_1^m, \xi_2^m, \cdots, \xi_{n_m+p_m+1}^m\}$ with p_m

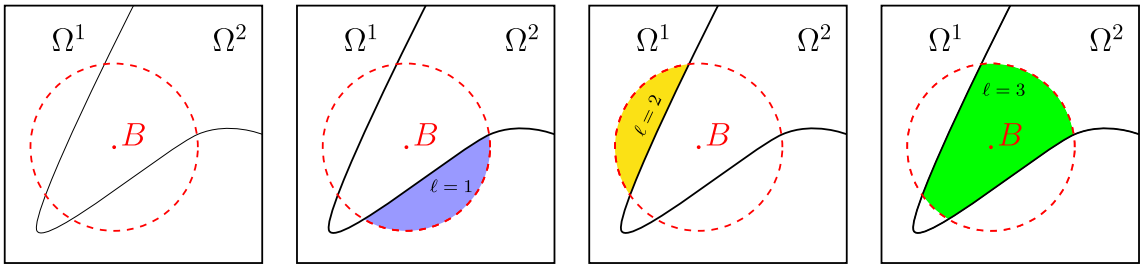


Fig. 4 Basis function enrichment for a two-material problem

being the polynomial degree and n_m the number of basis functions in the parametric direction $m = 1, \ldots, d_p$. A tensor-product B-spline basis function is generated from d_p univariate B-spline basis functions $N^m_{i_m,p_m}(\xi^m)$ in each parametric direction m as follows:

$$B_{\mathbf{i},\mathbf{p}}(\boldsymbol{\xi}) = \prod_{m=1}^{d_p} N^m_{i_m,p_m}(\xi^m), \tag{4}$$

where the position in the tensor-product structure is given by the index vector $\mathbf{i} = \{i_1, \ldots, i_{d_p}\}$, and the vector $\mathbf{p} = \{p_1, \ldots, p_{d_p}\}$ defines the polynomial degree in each direction. Similarly to the univariate case, an element is defined as the tensor-product of d_p non-empty knot spans. Additionally, a B-spline space $\mathcal{V}$ is defined as the span of B-spline basis functions.

3.2 B-spline refinement

Hierarchical refinement of uniform B-splines is achieved by subdivision. A univariate B-spline basis function is expressed as a linear combination of $p + 2$ contracted, translated, and scaled copies of itself:

$$N_p(\xi) = 2^{-p} \sum_{j=0}^{p+1} \binom{p+1}{j} N_p(2\xi - j), \tag{5}$$

where the binomial coefficient is defined as:

$$\binom{p+1}{j} = \frac{(p+1)!}{j!(p+1-j)!}. \tag{6}$$

Figure 5 shows the refinement of a quadratic univariate B-spline basis function obtained by subdivision.

The subdivision in Eq. (4) for a univariate B-spline basis can be extended to tensor-product B-spline basis functions B_p as follows, see Schillinger et al. [44]:

$$B_{\mathbf{p}}(\boldsymbol{\xi}) = \sum_{\mathbf{j}} \left(\prod_{m=1}^{d} 2^{-p_m} \binom{p_m+1}{j_m} N_{p_m}(2\xi^m - j_m) \right), \tag{7}$$

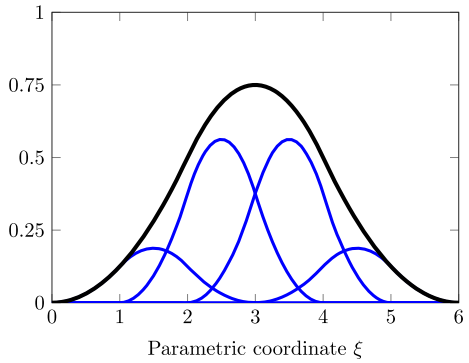


Fig. 5 Subdivision of a quadratic B-spline basis function (black) into $p + 2$ contracted B-spline basis functions (blue) of half the knot span width. (Color figure online)

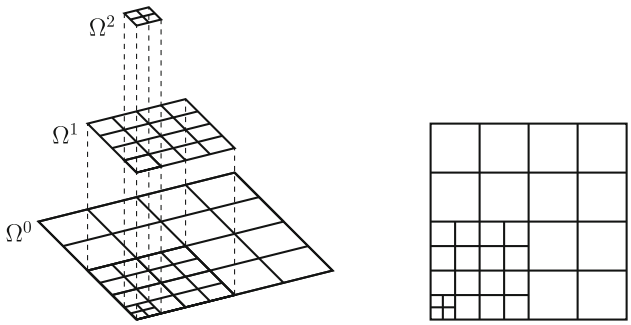


Fig. 6 Hierarchically refined mesh

where the index vector $\mathbf{j} = \{i_1, \ldots, i_{d_p}\}$ collects the positions in the tensor-product structure.

3.3 Hierarchical B-splines

To define a hierarchical mesh of depth n, a sequence of subdomains Ω^l is introduced:

$$\Omega^{n-1} \subseteq \Omega^{n-2} \subseteq \cdots \subseteq \Omega^0 = \Omega, \tag{8}$$

where each subdomain Ω^l is a refined sub-region of Ω^{l-1}. Consequently, Ω is equal to the union of all the subdomains Ω^l.

To build a hierarchical B-spline basis, a sequence of tensor-product B-spline spaces is introduced:

$$\mathcal{V}^0 \subset \mathcal{V}^1 \subset \mathcal{V}^2 \subset \mathcal{V}^3 \subset \cdots . \tag{9}$$

Each B-spline space $\mathcal{V}^l$ has a corresponding basis $\mathcal{B}^l$ and a corresponding mesh $\mathcal{K}^l$ of tensor-product elements.

A hierarchical B-spline basis $\mathcal{H}$ is constructed recursively based on the sequence of B-spline bases $\mathcal{B}^l$ that span the domains Ω^l. In an initial step, the basis functions defined on the coarsest level, $l = 0$, are collected and assigned to $\mathcal{H}^0$. The hierarchical B-spline basis $\mathcal{H}^{l+1}$ is constructed by taking the union of all basis functions B in $\mathcal{H}^l$ whose support is not fully enclosed in Ω^{l+1} and all basis functions B in $\mathcal{B}^{l+1}$ whose support lies in Ω^{l+1}. The recursive algorithm reads [21]:

$$\begin{cases} \mathcal{H}^0 =: \mathcal{B}^0 \\ \mathcal{H}^{l+1} =: \{B \in \mathcal{H}^l \mid \operatorname{supp}(B) \nsubseteq \Omega^{l+1}\} \cup \\ \qquad \{B \in \mathcal{B}^{l+1} \mid \operatorname{supp}(B) \subseteq \Omega^{l+1}\}, \\ \qquad \text{for } l = 0, \ldots, n-2, \end{cases} \tag{10}$$

where the index l denotes the level of refinement. Basis functions collected in $\mathcal{H}$, where $\mathcal{H} := \mathcal{H}^{n-1}$, are called active, while basis functions in $\mathcal{B}^l$ not present in $\mathcal{H}$ are said to be inactive.

Associated with a hierarchical B-spline basis is a hierarchically refined mesh

$$\mathcal{K} := \cup_{l=0}^{n-1} \left\{ K \in \mathcal{K}^l : K \in \Omega^l \text{ and } K \notin \Omega^{l+1} \right\} \tag{11}$$

wherein Ω^n is taken to be the empty set. An example of a hierarchically refined mesh associated with a two-dimensional hierarchical B-spline basis is displayed in Fig. 6. A hierarchical B-spline basis is smooth over each element of its associated hierarchically refined mesh.

A hierarchical B-spline basis $\mathcal{H}$ is illustrated for a one-dimensional example in Fig. 7. The top row shows a one-dimensional hierarchically refined mesh. Below the mesh, the basis functions for three refinement levels are shown. Following the recursion rule of Eq. (10), a B-spline basis $\mathcal{H}$ is created through an initialization step with all bases in the subdomain Ω^0 refined to a level $l = 0$. All bases in the subdomain Ω^{l+1} with higher refinement level $l + 1$ are added recursively, while existing basis functions of level l fully enclosed in Ω^{l+1} are discarded. The active B-spline basis functions $\mathcal{H}$ are shown in black, while the inactive B-spline basis functions are shown in gray.

Hanging nodes are a byproduct of h-refinement in hierarchical refined meshes and are naturally handled by the B-spline bases. In contrast, hanging nodes in classic finite elements require extra treatments, such as the introduction of multi-point constraints.

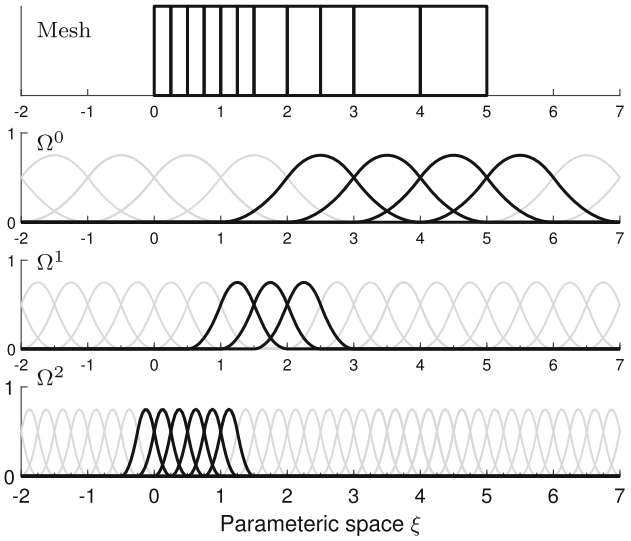


Fig. 7 One-dimensional hierarchical B-spline basis attained using three levels of refinement

3.4 Truncated B-splines

By construction, the hierarchical B-spline basis presented above does not fulfill the PU property. The truncated hierarchical B-spline basis constitutes an alternative to the hierarchical B-spline basis that does satisfy the PU property. Truncation also reduces the number of overlapping functions on adjacent hierarchical levels, see Giannelli et al. [23]. Considering a basis function B^l, part of $\mathcal{B}^l$ and defined on the domain Ω^l, its representation in terms of the finer basis of level $l + 1$ is given as:

$$B^l = \sum_{B^{l+1} \in \mathcal{B}^{l+1}} c^{l+1}_{B^{l+1}} \left(B^l \right) B^{l+1}, \tag{12}$$

where $c^{l+1}_{B^{l+1}}$ is the coefficient associated to a basis function B^{l+1}.

The truncation of this basis function B^l, whose support overlaps with the support of finer basis functions B^{l+1}, part of $\mathcal{B}^{l+1}$ and defined on Ω^{l+1}, is attained as follows [21,23]:

$$\begin{aligned} \operatorname{trunc}^{l+1}(B^l) &= \sum_{\substack{B^{l+1} \in \mathcal{B}^{l+1}, \\ \operatorname{supp}(B^{l+1}) \nsubseteq \Omega^{l+1}}} c^{l+1}_{B^{l+1}} \left(B^l \right) B^{l+1}, \\ &= B^l - \sum_{\operatorname{supp}(B^{l+1}) \subseteq \Omega^{l+1}} c^{l+1}_{B^{l+1}} \left(B^l \right) B^{l+1}. \end{aligned} \tag{13}$$

Following the creation of a hierarchical B-spline basis $\mathcal{H}$, a THB basis $\mathcal{T}$ is constructed recursively by considering the

truncation in Eq. (13), see Giannelli et al. [23], Garau and Vázquez [21]:

$$\begin{cases} \mathcal{T}^0 =: \mathcal{B}^0 \\ \mathcal{T}^{l+1} =: \{\text{trunc}^{l+1}(B) \mid B \text{ in } \mathcal{T}^l \wedge \text{supp}(B) \nsubseteq \Omega^{l+1}\} \\ \qquad \cup \{B \in \mathcal{B}^{l+1} \mid \text{supp}(B) \subseteq \Omega^{l+1}\}, \\ \qquad \text{for } l = 0, \ldots, n-2. \end{cases} \tag{14}$$

The truncated basis $\mathcal{T}$ spans the same space as the non-truncated basis $\mathcal{H}$ and it admits a strong stability property [24]. Moreover, the smaller support results in a reduction in the number of nonzero basis functions per element and consequently a sparser system of linear equations in a finite element analysis.

The effect of the truncation is illustrated in Fig. 8. A univariate truncated and non-truncated basis $\mathcal{T}$ and $\mathcal{H}$ are juxtaposed. The first and second levels correspond to Ω^0 and Ω^1 respectively, while the bottom level represents the combination of the functions on these two levels. The comparison shows the reduced support of the truncated B-spline basis functions.

4 Implementation details

This section details the implementation of the THB discretization described above. A computationally efficient approach is introduced to represent hierarchically refined meshes via a poly-tree data structure. Local mesh refinement strategies are outlined. A methodology for expressing THB basis functions associated with multiple, differently refined meshes in terms of Lagrange shape functions over each element in a union background mesh is presented.

4.1 Poly-tree data structure for hierarchically refined B-spline discretizations

The generation of a large hierarchical refined mesh can be expensive due to the high memory consumption, especially when creating all possible elements for each refinement level. The cost is compounded when creating multiple, differently refined hierarchical meshes. In this paper, we present a computationally and memory efficient approach to build differently refined meshes.

We first construct a Poly-Tree (PT) data structure, i.e., a quadtree in 2D and an octree in 3D. The PT data structure represents a set of hierarchically refined meshes. The tree depth corresponds to a refinement level l with the base level, 0, representing a coarse uniform tensor mesh. The nodes of this PT data structure are labeled PT cells and used to construct discretizations of state variable fields. Note that the PT cells are not associated with any specific set of basis functions.

PT cells with higher refinement levels $l > 0$ are created recursively. Starting from level 0, the PT cells are recursively subdivided into 4 and 8 PT cells in 2D and 3D, respectively. Considering a PT cell at a refinement level l, the PT cell at refinement level $l-1$ from which the PT cell is created is referred to as its parent. The PT cells at refinement level $l+1$ created by subdivision of a PT cell at level l are called its children. The PT cells are only created once and only as needed.

To efficiently represent multiple, differently refined meshes with the same PT data structure, we introduce the concept of PT cell activation states. Each PT cell has multiple activation states which are represented by Activation Indices (AI).

Possible activation states for a particular AI are either *active*, *refined*, or *inactive*. *Active* PT cells for a specific AI have *refined* parent PT cells and *inactive* children PT cells.

Although PT cells are only constructed once, the activation state concept allows for the creation of multiple, differently refined meshes. A single activation state can be stored as a binary number with only two bits. Thus, the activation states for all AIs of a PT cell are stored in $2 \cdot \#\text{AIs}$ bits. An illustration of a single quad-tree data structure with two AIs is presented in Fig. 9. The coloring of the PT cells indicates the activation state for a specific AI. For a particular AI, PT cells in blue, red, and white are *active*, *refined*, and *inactive*, respectively. Blue framed PT cells on the zero level illustrate the resulting mesh of active PT cells on a given AI. This enables the efficient construction of differently refined hierarchical B-spline discretizations based on a single PT data structure and a set of AIs. To see this, let Ω_I^l be the union of active and refined cells associated with level l and AI I. Then, we can create a hierarchical B-spline basis $\mathcal{H}_I$ and a corresponding hierarchically refined mesh $\mathcal{K}_I$ for AI I from the sequence

$$\Omega_I^{n-1} \subseteq \Omega_I^{n-2} \subseteq \cdots \subseteq \Omega_I^0 = \Omega. \tag{15}$$

The resulting hierarchically refined mesh $\mathcal{K}_I$ is precisely the set of active PT cells for AI I.

In our PT data structure, we only store PT cells that are either active or refined for at least one AI I. The element edge length $h^0_{\text{PTcell},m}$ for parametric direction m and on the coarsest level 0 is computed based on the given mesh size and the number of coarsest elements in each dimension, both of them are predefined by the user. The size of mesh elements corresponding to PT cells on levels $l > 0$ may be computed based on the coarsest element edge length and the refinement level via $h^l_{\text{PTcells},m} = 2^{-l} h^0_{\text{PTcell},m}$. Basis functions are assigned to mesh elements based on a pre-defined elemental basis function ordering. The elemental basis function ordering in this work follows the Exodus II standard [46].

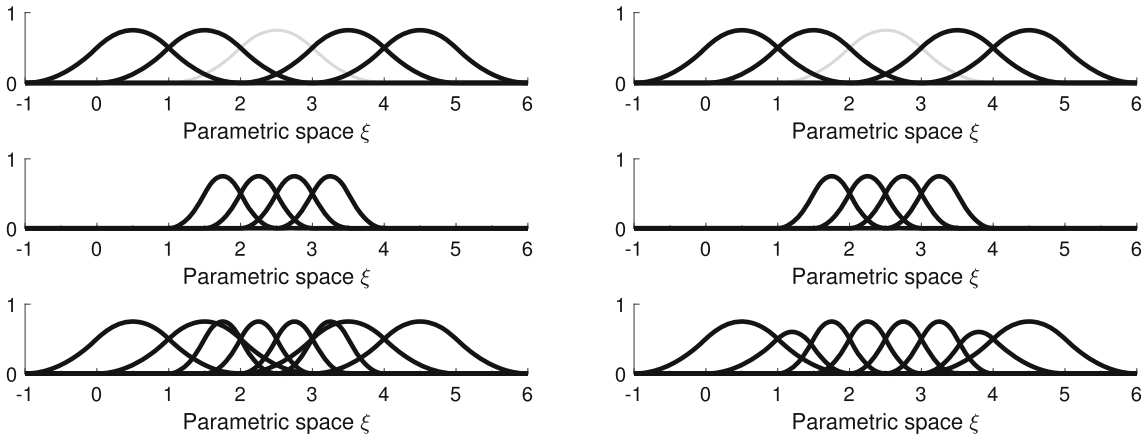


Fig. 8 Comparison of univariate HB-spline (left) and THB-spline basis functions (right)

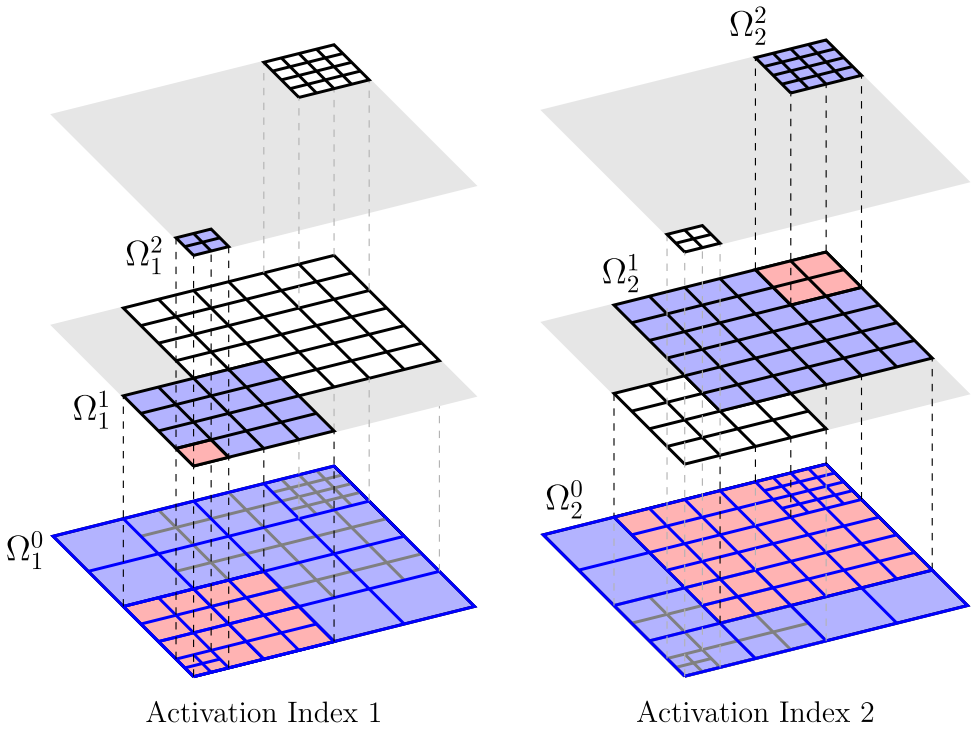


Fig. 9 One hierarchical refined quad-tree data structure with two sets of AIs

4.2 Local refinement strategies

To generate a PT data structure that supports different hierarchically refined meshes, the PT is recursively refined for each AI and corresponding refinement criteria. The refinement procedure is described in Algorithm 1. For a specific AI, *active* PT cells are flagged for refinement based on chosen refinement criteria. Adjacent PT cells can also be flagged for refinement to increase the size of the refined region. Moreover, additional active PT cells for the current AI may be flagged considering mesh regularity requirements as discussed in Sect. 4.2.1. In case the PT data structure is generated in parallel using a domain decomposition strategy, flagged PT cells are communicated across adjacent subdomains to guarantee consistent refinement. Refinement is then performed by creating new children PT cells through subdividing all flagged PT cells unless the children PT cells do already exist. New PT cells are initialized with an *inactive* activation state for all AIs. For the AI currently considered for refinement, the activation state of the children PT cells is set to *active* while the parent PT cell is set to *refined*.

Algorithm 1 Refinement algorithm

```
for Activation Index (AIs) do
    Flag PT cells based on refinement criteria for AI
    Collect flagged PT cells into queue for refinement
    while changes in refinement queue do
        for all PT cells in queue for refinement do
            Apply refinement buffer Algorithm 2
            Collect flagged PT cells in queue for refinement
        end for
    end while
    Communicate refinement queue
    if Refined children PT cell does not exist then
        Create refined children PT cells
        Initialize new PT cells as inactive for all AIs
    end if
    Flag children PT cells as active for AI
    Flag parent PT cell as refined for AI
end for
```

4.2.1 Mesh regularity requirements

For construction of THB bases as discussed in Sect. 3, mesh regularity requirements need to be considered when constructing the PT data structure. The difference in refinement level between adjacent PT cells in the refined PT is limited to one. Furthermore, all active neighbor PT cells inside a so-called buffer zone of an active PT cell on level l must be of level greater or equal $l-1$, see Eq. (14).

The buffer range $d^l_{\text{buffer},m}$ for a particular active PT cells is calculated by multiplying the PT cells size with a user-defined buffer parameter b_{buffer}, i.e., $d^l_{\text{buffer},m} = b_{\text{buffer}}\, h^l_{\text{PTcells},m}$. When creating a B-spline basis, the width of the buffer zone in parametric direction m must be larger than or equal to the width of the basis function supports in parametric dimension m. In this work each state variable field is interpolated with an individual interpolation order p. To satisfy the mesh regularity requirement for all interpolation functions, the buffer parameter must be chosen as $b_{\text{buffer}} \geq p_{\max}$, where $p_{\max}$ is the maximal polynomial degree of all used bases.

The refinement procedure for enforcing a buffer zone is summarized in Algorithm 2. The algorithm is applied to each PT cell initially flagged for refinement in Step 2 of Algorithm 1 and starts by determining its parent PT cell. The refinement status of the parent's neighbors, i.e., cells within the buffer range of the considered parent, are checked. If these neighbors are neither refined nor flagged for refinement, the distance d_m between the centers of the considered PT cell and its neighbor's children cells is calculated. This operation is trivial even if these neighboring children cells do not exist due to the PT data structure information. If any distance d_m is smaller than the buffer range $d^l_{\text{buffer},m}$, the particular neighbor cell is flagged for refinement. The algorithm is then applied recursively to all newly flagged neighbor cells, until no further cells are flagged for refinement. An efficient access to hierarchical mesh information, such as neighborhood relationships, is provided by the PT data structure discussed in Sect. 4.4. A visual representation of Algorithm 2 is provided in Fig. 10 for $b_{\text{buffer}} = 1$. PT cells with a green fill are cells flagged for refinement while PT cells with a green pattern fill are neighboring PT cells of the parent cell.

Algorithm 2 Refinement buffer algorithm

```
PT cell flagged for refinement in Algorithm 1
Get parent
Get parent's neighbors in buffer range d^l_buffer
if neighbor is active and not flagged for refinement then
    Calculate distance d_ma considered cell and neighbors non-existing children PT cells
    if d_m < d_buffer then
        Flag neighbor for refinement
        Apply refinement buffer Algorithm 2 for neighbor
    end if
end if
```

4.3 Union background mesh and extraction operators

The framework presented above allows for a different hierarchically refined B-spline discretization for each state variable field. These discretizations are defined on locally refined background meshes that are not aligned with the external boundaries and material interfaces. To simplify the finite element formation and assembly process, a union[1] background mesh $\mathcal{K}_{union}$ is constructed from the hierarchically refined meshes for each state variable field as follows:

$$\mathcal{K}_{union} := \{K : K \in \mathcal{K}_I \text{ for some } I \text{ and there is no } K' \in \mathcal{K}_J \text{ for } J \neq I \text{ such that } K' \subsetneq K\}. \quad (16)$$

That is, an element of a hierarchically refined mesh for one of the state variable fields is an element of the union background mesh if it contains no finer element belonging to a hierarchically refined mesh for one of the other state variable fields. An example of a union background mesh is displayed in Fig. 3. The union background mesh is specially defined so that the hierarchical B-spline basis functions associated with each state variable field are smooth over each element of the union background mesh. The union background mesh can be conveniently created via the PT data structure by combining all AIs used to generate THB background meshes. In particular, a PT cell corresponds to an element of the union background mesh if it is active for at least one activation

[1] Note that the union background mesh is not attained via a set union of the separate hierarchically refined meshes. Instead, elements in this set union that fully contain smaller elements in the set union do not belong to the union background mesh.

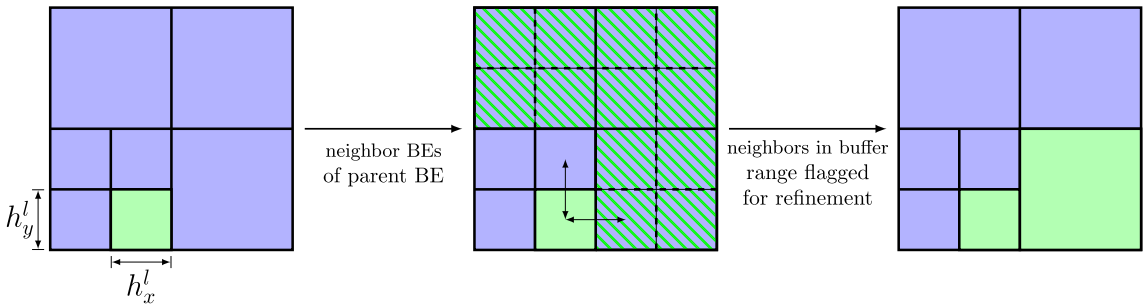


Fig. 10 Visualization of refinement buffer Algorithm 2

index and active or inactive for all other activation indices. The union background mesh serves as the XFEM background mesh in which the geometry is immersed. Intersected XFEM background elements of the union background mesh are cut to generate a body-fitted integration mesh. Each THB discretization is enriched separately considering the B-spline interpolation and the immersed geometry represented on the union background mesh. For details on the enrichment strategy, the reader is referred to Noël et al. [36]. To increase the geometry resolution, the union mesh can be further refined, either locally or globally.

To facilitate operations performed on the union background mesh and on the integration mesh in the XFEM analysis, the THB basis functions for each locally refined background mesh are represented in terms of Lagrange shape functions on each element of the union background mesh via Lagrange extraction, see Schillinger et al. [45], D'Angella et al. [11]. Namely, the THB basis functions for a locally refined background mesh can be represented element-wise in terms of Lagrange shape functions as

$$B_k(\boldsymbol{\xi}) = \sum_j B_k(\boldsymbol{\xi}_j) N_j^L(\boldsymbol{\xi}) = \sum_j T_{jk}^L N_j^L(\boldsymbol{\xi}) \tag{17}$$

where B_k is the kth THB basis function, $\left\{N_j^L\right\}_j$ are the Lagrange shape functions over the element, $\left\{\boldsymbol{\xi}_j\right\}_j$ are the locations at which the Lagrange shape functions $\left\{N_j^L\right\}_j$ are interpolatory, and $T_{jk}^L = B_k(\boldsymbol{\xi}_j)$. We refer to T^L as a Lagrange extraction operator. The Lagrange shape functions over each element can in turn be represented in terms of Lagrange shape functions over a child element as

$$N_j^L(\boldsymbol{\xi}) = \sum_i N_j^L(\hat{\boldsymbol{\xi}}_i) \hat{N}_i^L(\hat{\boldsymbol{\xi}}_i) = \sum_i T_{ij}^h \hat{N}_i^L(\hat{\boldsymbol{\xi}}_i) \tag{18}$$

where $\left\{\hat{N}_i^L\right\}_i$ are the Lagrange shape functions over the child element, $\hat{\boldsymbol{\xi}}_i$ are the locations at which the Lagrange shape functions $\left\{\hat{N}_i^L\right\}_i$ are interpolatory, and $T_{ij}^h = N_j^L(\hat{\boldsymbol{\xi}}_i)$. We refer to T^h as an h-refinement extraction operator. It follows that the THB basis functions can be expressed in terms of the Lagrange shape functions over the child element as

$$B_k(\boldsymbol{\xi}) = \sum_i T_{ik} \hat{N}_i^L(\boldsymbol{\xi}) \tag{19}$$

where

$$T_{ik} = \sum_j T_{jk}^L T_{ij}^h \tag{20}$$

is a Lagrange extraction operator that can be computed using the aforementioned Lagrange and h-refinement extraction operators. This process can be repeated to represent THB basis functions in terms of Lagrange shape functions and easily computable Lagrange extraction operators over descendent elements of the child element as well, thus enabling us to represent the THB basis functions for each state variable field in terms of Lagrange shape functions and Lagrange extraction operators over each element of the union background mesh. The extraction process from a quadratic B-spline basis to a quadratic, once refined Lagrange basis is illustrated in Fig. 11. Our framework limits the background element refinement to a factor of two, as presented in Sect. 4.2.1. This consequently limits the number of h-refinement extraction matrices to four in 2D and eight in 3D. These matrices can be precomputed and efficiently selected exploiting the PT data structure.

4.4 Background mesh data structure

In this subsection, we discuss computational aspects of the PT data structure for storing and managing hierarchically refined meshes. This data structure aims at improving the overall computational efficiency and minimizes inter-process communication for parallel computations. The presented implementation is limited to quadtrees in 2D and octrees in 3D, which is sufficient for IFEMs.

Given the dimensions of the computational domain and the number of elements in each spatial direction, the base level

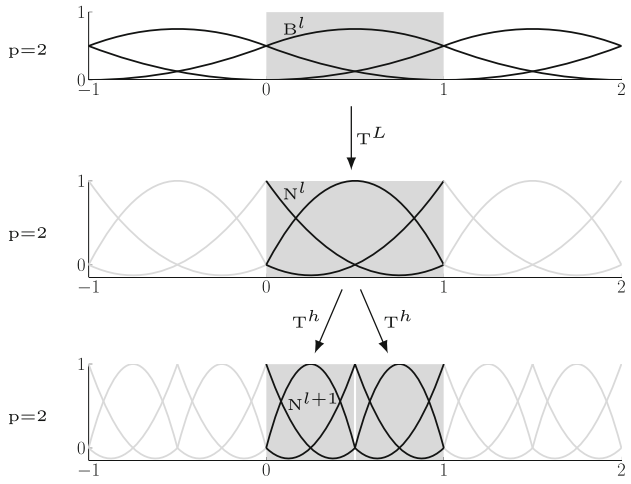


Fig. 11 Illustration of the extraction process

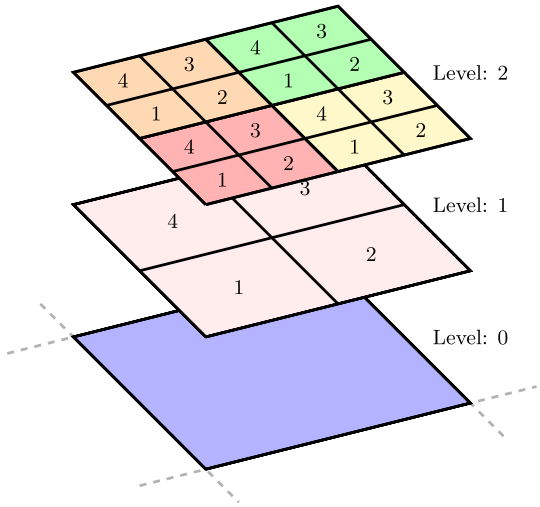


Fig. 12 A coarse two dimensional PT cell refined through recursive subdivision and resulting quadtree data structure

0 of the PT data structure is generated. Adopting a domain decomposition approach, the PT cells are grouped into subdomains. The mesh is hierarchically refined, and the union mesh is created in parallel on each processor. To facilitate the refinement process and the construction of the Lagrange extraction operators, we create overlapping subdomains to build an efficient PT data structure in parallel. In many cases a decomposition strategy assigning approximately the same number of PT cells to each subdomain is sufficient. However, as refinement can lead to a significant imbalance in number of PT cells across subdomains, it may be beneficial to choose a decomposition strategy that accounts for refinement.

At the coarsest level, 0, a unique PT cell ID can be calculated based on the global PT cell location as illustrated in the uppermost graphic of Fig. 13. With the ID of the parent PT cell at refinement level 0 and the position in the PT structure, unique IDs of refined PT cells can be determined directly, without inter-process communication and building local-to-global ID maps. The positions of the PT structure for a two dimensional twice refined PT cells are illustrated in Fig. 12.

Furthermore, the subdomain-local PT data structure allows for efficient access to cell neighborhood and hierarchy relationships. The PT data structure also speeds up the construction of extraction operators and may be used for the identification of basis functions. Basis functions can be uniquely identified by utilizing the PT structure in combination with the elemental Exodus II basis function index.

To reduce the communication across subdomains, an aura of PT cells is constructed around the set of PT cells owned by a specific processor, leading to overlapping subdomains. The width of the aura is chosen to be b_{buffer} PT cells on the coarsest refinement level with b_{buffer} defined as described in Sect. 4.2.1.

The PT cells in an aura are needed to define all THB basis functions that are nonzero over non-aura elements. There-

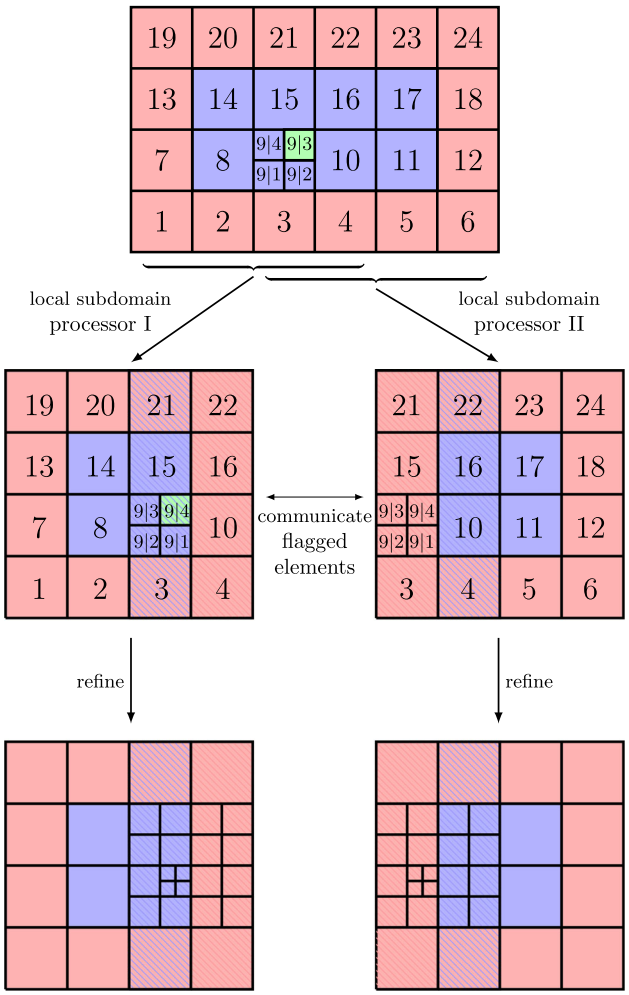


Fig. 13 Parallel local refinement of a global 2×4 domain

fore, most computations can be performed on a subdomain level, interlaced with only a few inter-subdomain communications. This includes mesh refinement along the processor boundaries, the construction of Lagrange extraction operators, and the evaluation of face-oriented ghost stabilization residuals and Jacobians, see Sect. 5. To illustrate the parallel concept, Fig. 13 presents a global two dimensional domain of size 2×4 elements in blue. PT cells in red indicate aura cells. The global domain is decomposed into two processor local subdomains. Unique global PT cell IDs are calculated based on IDs of the coarsest refinement level. Red striped PT cells are aura cells shared with the neighboring processor. Blue striped PT cells are cells of the inverse aura. The green cells are flagged for refinement. Parallel consistent refinement is guaranteed through communication of the flagged PT cell IDs on aura and inverse aura.

5 Thermo-elastic model

The XIGA framework presented above is applicable to a broad range of physical problems that are described by partial differential equations. In this paper, we demonstrate the functionality of the developed framework with static elastic and thermo-elastic problems. The elastic model assumes infinitesimal strains and a linear elastic, isotropic material response. The thermal model accounts for linear diffusive heat transfer. The structural response depends on the temperature field through inelastic, isotropic thermal expansion. In this section, we present the variational form of the stabilized governing equations of the thermo-elastic model.

5.1 Governing equations

The weak form of the residual is decomposed into the following four contributions:

$$\mathcal{R} = \mathcal{R}^{\mathrm{T}} + \mathcal{R}^{\mathrm{U}} + \mathcal{R}^{\mathrm{N}} + \mathcal{R}^{\mathrm{G}} = 0, \tag{21}$$

where $\mathcal{R}^{\mathrm{T}}$ and $\mathcal{R}^{\mathrm{U}}$ combine the volumetric contributions and contributions of Neumann boundary conditions to the thermal and structural residuals, respectively. The contributions of Nitsche's formulation for Dirichlet boundary conditions are collected in $\mathcal{R}^{\mathrm{N}}$ and the contributions of the face-oriented ghost stabilization in $\mathcal{R}^{\mathrm{G}}$.

In the absence of body loads, the thermal and structural residual, $\mathcal{R}^{\mathrm{T}}$ and $\mathcal{R}^{\mathrm{U}}$, are defined over all material domains Ω^m with $m = 1, \ldots, N_m$ as follows:

$$\mathcal{R}^{\mathrm{T}} = \sum_{m=1}^{N_m} \int_{\Omega^m} \nabla \delta\mathbf{T}\, \boldsymbol{\kappa}\, \nabla \mathbf{T}\, d\Omega - \int_{\Gamma^m_{N,T}} \delta\mathbf{T}\, q_N\, d\Gamma, \tag{22}$$

and

$$\mathcal{R}^{\mathrm{U}} = \sum_{m=1}^{N_m} \int_{\Omega^m} \delta\boldsymbol{\varepsilon}_u : \boldsymbol{\sigma}\, d\Omega - \int_{\Gamma^m_{N,u}} \delta\mathbf{u} \cdot \mathbf{f}_N\, d\Gamma, \tag{23}$$

where $\mathbf{T}$ and $\delta\mathbf{T}$ are the temperature trial and test function, respectively. The isotropic thermal conductivity tensor is denoted by $\boldsymbol{\kappa}$. A heat flux q_N is applied at the boundaries $\Gamma^m_{N,T}$. The displacement trial and test functions are denoted by $\mathbf{u}$ and $\delta\mathbf{u}$, respectively. Traction forces, $\mathbf{f}_N$, are applied on the boundaries $\Gamma^m_{N,u}$. The Cauchy stress tensor is denoted by $\boldsymbol{\sigma} = \mathbf{D}\,\boldsymbol{\varepsilon}_e$ where $\mathbf{D}$ is the isotropic constitutive tensor and $\boldsymbol{\varepsilon}_e$ is the total infinitesimal strain tensor with $\boldsymbol{\varepsilon}_e = \boldsymbol{\varepsilon}_u - \boldsymbol{\varepsilon}_T$. The mechanical strain $\boldsymbol{\varepsilon}_u$ is computed by $\boldsymbol{\varepsilon}_u = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)$. The thermal strain is denoted by $\boldsymbol{\varepsilon}_T = \alpha\,(\mathbf{T} - \mathbf{T}_0)\,\mathbf{I}$ where α is the CTE and T_0 is the reference temperature.

To weakly enforce Dirichlet boundary and interface conditions, we use an unsymmetric formulation of Nitsche's method. The associated residual, $\mathcal{R}^{\mathrm{N}}$, is decomposed into boundary and interface terms, separately for the thermal and elastic subproblems as follows:

$$\mathcal{R}^{\mathrm{N}} = \mathcal{R}^{\mathrm{N},D}_{\mathbf{T}} + \mathcal{R}^{\mathrm{N},D}_{\mathbf{u}} + \mathcal{R}^{\mathrm{N},I}_{\mathbf{T}} + \mathcal{R}^{\mathrm{N},I}_{\mathbf{u}}, \tag{24}$$

where the thermal and elastic contributions from Dirichlet boundary conditions are denoted by $\mathcal{R}^{\mathrm{N},D}_{\mathbf{T}}$ and $\mathcal{R}^{\mathrm{N},D}_{\mathbf{u}}$, respectively. The thermal and elastic contributions from interface conditions are $\mathcal{R}^{\mathrm{N},I}_{\mathbf{T}}$ and $\mathcal{R}^{\mathrm{N},I}_{\mathbf{u}}$, respectively. These Dirichlet boundary residual contributions are defined as follows:

$$\begin{aligned}\mathcal{R}^{\mathrm{N},D}_{T} = &\sum_{m=1}^{N_m} - \int_{\Gamma^{D,T}} \delta\mathbf{T}\,(\boldsymbol{\kappa}\,\nabla\mathbf{T}) \cdot \mathbf{n}_\Gamma\, d\Gamma \\ &+ \int_{\Gamma^{D,T}} (\boldsymbol{\kappa}\,\nabla\delta\mathbf{T}) \cdot \mathbf{n}_\Gamma\,(\mathbf{T} - \mathbf{T}_D)\, d\Gamma \\ &+ \sum_{e=1}^{N^B_e} \gamma^e_{D,T} \int_{\Gamma^{D,T}_e} \delta\mathbf{T}\,(\mathbf{T} - \mathbf{T}_D)\, d\Gamma,\end{aligned} \tag{25}$$

and

$$\begin{aligned}\mathcal{R}^{\mathrm{N},D}_{u} = &\sum_{m=1}^{N_m} - \int_{\Gamma^{D,u}} \delta\mathbf{u} \cdot \boldsymbol{\sigma}(\mathbf{u})\, \mathbf{n}_\Gamma\, d\Gamma \\ &+ \int_{\Gamma^{D,u}} \boldsymbol{\sigma}(\mathbf{u})\, \mathbf{n}_\Gamma \cdot (\mathbf{u} - \mathbf{u}_D)\, d\Gamma \\ &+ \sum_{e=1}^{N^B_e} \gamma^e_{D,u} \int_{\Gamma^{D,u}_e} \delta\mathbf{u} \cdot (\mathbf{u} - \mathbf{u}_D)\, d\Gamma.\end{aligned} \tag{26}$$

The temperature T_D is prescribed on the boundary $\Gamma^{D,T}$ and the displacement $\mathbf{u}_D$ is prescribed on $\Gamma^{D,u}$. The vector $\mathbf{n}_\Gamma$ denotes the outward pointing normal on the boundary. The

summations over all N_e^B elements in the union background mesh in Eqs. (25) and (26) penalize constraint violations along the boundaries $\Gamma_{D,T}^e$ and $\Gamma_{D,u}^e$ within the elements of the union background mesh. We henceforth refer to the elements of the union background mesh as background elements. The penalty parameters $\gamma_{D,T}^e$ and $\gamma_{D,u}^e$ depend on the size of the background element, h_e^B, and are defined as follows:

$$\gamma_{D,T} = c_{D,T}\frac{\kappa}{h_e^B} \quad \text{and} \quad \gamma_{D,u} = c_{D,u}\frac{E}{h_e^B}, \tag{27}$$

where κ is the isotropic material conductive and E is the Young's modulus of the linear elastic material. The parameters $c_{D,T} \geq 0$ and $c_{D,u} \geq 0$ control the accuracy of enforcing the Dirichlet boundary conditions.

Continuity of temperature and displacement fields and balance of heat flux and traction must be satisfied at all interfaces $\Gamma^{m,n} = \Omega^m \cap \Omega^n \neq \emptyset$. Nitsche's method for the thermal and structural interface conditions yields the following residual contributions:

$$\begin{aligned}\mathcal{R}_T^{\mathrm{N},I} = &\sum_{\Gamma^{m,n}}\sum_{e=1}^{N_e^B} -\int_{\Gamma_e^{m,n}} [\![\delta\mathbf{T}]\!]\,\{\boldsymbol{\kappa}\,\nabla\mathbf{T}\}\cdot\mathbf{n}_{m,n}\,d\Gamma \\ &+\int_{\Gamma_e^{m,n}} \{\boldsymbol{\kappa}\,\nabla\delta\mathbf{T}\}\cdot\mathbf{n}_{m,n}\,[\![\mathbf{T}]\!]\,d\Gamma \\ &+\gamma_{I,T}^e\int_{\Gamma_e^{m,n}} [\![\delta\mathbf{T}]\!]\,[\![\mathbf{T}]\!]\,d\Gamma,\end{aligned} \tag{28}$$

$$\begin{aligned}\mathcal{R}_u^{\mathrm{N},I} = &\sum_{\Gamma^{m,n}}\sum_{e=1}^{N_e^B} -\int_{\Gamma_e^{m,n}} [\![\delta\mathbf{u}]\!]\cdot\{\boldsymbol{\sigma}(\mathbf{u})\}\,\mathbf{n}_{m,n}\,d\Gamma \\ &+\int_{\Gamma_e^{m,n}} \{\boldsymbol{\sigma}(\delta\mathbf{u})\}\,\mathbf{n}_{m,n}\cdot[\![\mathbf{u}]\!]\,d\Gamma \\ &+\gamma_{I,u}^e\int_{\Gamma_e^{m,n}} [\![\delta\mathbf{u}]\!]\cdot[\![\mathbf{u}]\!]d\Gamma,\end{aligned} \tag{29}$$

where the jump operator is defined as $[\![\cdot]\!] = (\cdot)^m - (\cdot)^n$. The numerical interface flux and traction are defined by the averaging operator as $\{\cdot\} = w^m(\cdot)^m + w^n(\cdot)^n$, where w^m and w^n are weights. The vector $\mathbf{n}_{m,n}$ denotes the normal vector on the interface pointing from phase m to phase n. The accuracy of enforcing the interface condition is controlled by the penalty terms with $\gamma_{I,T}^e$ and $\gamma_{I,u}^e$ being the elemental penalty factors.

We follow the work of Annavarapu et al. [1] and define weights for the numerical heat flux as follows:

$$\begin{aligned} w_T^m &= \frac{\mathrm{meas}(\Omega^m)/\kappa^m}{\mathrm{meas}(\Omega^m)/\kappa^m + \mathrm{meas}(\Omega^n)/\kappa^n}, \\ w_T^n &= \frac{\mathrm{meas}(\Omega^n)/\kappa^n}{\mathrm{meas}(\Omega^m)/\kappa^m + \mathrm{meas}(\Omega^n)/\kappa^n}, \end{aligned} \tag{30}$$

and for the numerical traction as follows:

$$\begin{aligned} w_u^m &= \frac{\mathrm{meas}(\Omega^m)/E^m}{\mathrm{meas}(\Omega^m)/E^m + \mathrm{meas}(\Omega^n)/E^n}, \\ w_u^n &= \frac{\mathrm{meas}(\Omega^n)/E^n}{\mathrm{meas}(\Omega^m)/E^m + \mathrm{meas}(\Omega^n)/E^n}, \end{aligned} \tag{31}$$

where $\mathrm{meas}(\Omega^k)$ is the surface in 2D or the volume in 3D of the domain occupied by the phase k within the background element. The elemental penalty factors in Eqs. (28) and (29) are defined by:

$$\begin{aligned} \gamma_{I,T}^e &= 2\,c_{I,T}\,\frac{\mathrm{meas}(\Gamma^{m,n})}{\mathrm{meas}(\Omega^m)/\kappa^m + \mathrm{meas}(\Omega^n)/\kappa^n} \\ \gamma_{I,u}^e &= 2\,c_{I,u}\,\frac{\mathrm{meas}(\Gamma^{m,n})}{\mathrm{meas}(\Omega^m)/E^m + \mathrm{meas}(\Omega^n)/E^n}, \end{aligned} \tag{32}$$

where the operator $\mathrm{meas}(\Gamma^{m,n})$ measures the length in 2D or the area in 3D of the interface within the background element. The parameters $c_{I,T} \geq 0$ and $c_{I,u} \geq 0$ control the accuracy of enforcing the interface conditions.

5.2 Face-oriented ghost stabilization

The utilized immersed method may suffer from numerical instabilities caused by basis functions with small support within the geometric domain. This may occur when an interface moves close to the boundary of the support of a basis function. Such configurations may result in ill-conditioning of the system of linear equations as well as imprecise spatial gradients of the state variable field, see de Prenter et al. [13]. Recent works have proposed a variety of strategies to mitigate the issues arising from basis function with a small support. Most notable strategies are the basis function removal as presented by Elfverson et al. [18], the concept of web-splines as presented by Höllig et al. [27] and Marussig and Hughes [31], or face-oriented ghost stabilization introduced by Burman and Hansbo [10].

This work adopts the face-oriented ghost stabilization approach presented by Burman and Hansbo [10] and adapted by Noël et al. [36] to fit the basis function enrichment strategy described in Sect. 2.

Let Ω^m be a material subdomain phase m and K_{Ω^m} the set of background elements that have a non-empty intersection with Ω^m:

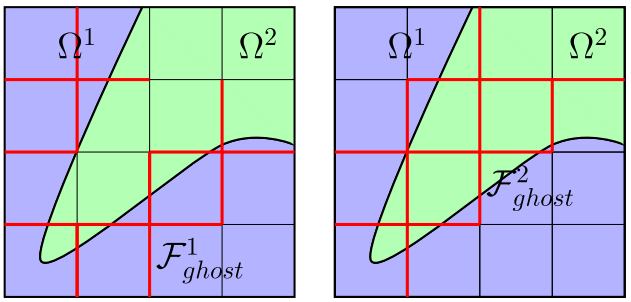


Fig. 14 Set of ghost facets $\mathcal{F}^1_{\text{ghost}}$ and $\mathcal{F}^2_{ghost}$ used for the face-oriented ghost stabilization for a two-material problem

$$K_{\Omega^m} := \left\{ K \in \mathcal{K}_{union} : K \cap \Omega^m \neq \emptyset \right\}. \tag{33}$$

We define $\mathcal{F}^m_{int}$ as the set of interior facets of K_{Ω^m}, i.e., the facets F shared between two background elements $\Omega^{m,+}_F$ and $\Omega^{m,-}_F$ of K_{Ω^m}. Let $\tilde{\Gamma}$ be the union of all material interfaces and geometric boundaries, here defined as material interfaces between void and non-void regions. The set of ghost facets for phase m is:

$$\mathcal{F}^m_{ghost} := \left\{ F \in \mathcal{F}^m_{int} : \Omega^{m,+}_F \cap \tilde{\Gamma} \neq \emptyset \text{ or } \Omega^{m,-}_F \cap \tilde{\Gamma} \neq \emptyset \right\}. \tag{34}$$

Consider a ghost facet F^m shared between two adjacent background elements $\Omega^{m,+}_F$ and $\Omega^{m,-}_F$. The normal to the facet is $\mathbf{n}_F$ and is chosen as $\mathbf{n}_F = \mathbf{n}^{m,+}_F = -\mathbf{n}^{m,-}_F$. The material layout subdivides the element $\Omega^{m,+}_F$ into $N^{m,+}_F$ connected subdomains $\Omega^{m,+}_{F,i}$ with $i = 1, \ldots, N^{m,+}_F$ and the element $\Omega^{m,-}_F$ into $N^{m,-}_F$ connected subdomains $\Omega^{m,-}_{F,j}$ with $j = 1, \ldots, N^{m,-}_F$ (Fig. 14).

We define $\mathbf{u}^{m,+}_{F,i}$ as the polynomial extension of the field $\mathbf{u}|_{\Omega^{m,+}_{F,i}}$ to all of $\mathbb{R}^d$ and let $\mathbf{u}^{m,-}_{F,j}$ be the polynomial extension of the field $\mathbf{u}|_{\Omega^{m,-}_{F,j}}$ to all of $\mathbb{R}^d$.

Additionally, our formulation requires that: $|\partial\Omega^{m,+}_{F,i} \cap \partial\Omega^{m,-}_{F,j}| \neq 0$, and the ghost stabilization is only applied between $\mathbf{u}^{m,+}_{F,i}$ and $\mathbf{u}^{m,-}_{F,j}$ when the boundaries of $\Omega^{m,+}_{F,i}$ and $\Omega^{m,-}_{F,j}$, $\partial\Omega^{m,+}_{F,i}$ and $\partial\Omega^{m,-}_{F,j}$, respectively, meet along a portion of the facet F with a non-zero measure, e.g., the boundaries meet along more than a point in two dimensions and along more than a line in three dimensions.

With the above terminology defined, the contribution of the ghost stabilization for the displacement field to the residual equations is:

$$\mathcal{R}^{\mathrm{G}}_{\mathbf{u}} = \sum_{m=1}^{N_m} \sum_{F \in \mathcal{F}^m_{ghost}} \sum_{i=1}^{N^{m,+}_F} \sum_{\mathcal{J}^m_{\mathcal{F},i}} \left[\int_F \gamma^{\mathbf{u}}_G \, h^{\tilde{k}} [\![\partial^p_n \mathbf{u}]\!] [\![\partial^p_n \mathbf{u}]\!] d\Gamma \right], \tag{35}$$

where the set $\mathcal{I}^m_{\mathcal{F},i}$ is defined as:

$$\mathcal{J}^m_{\mathcal{F},i} := \{ k \in \{1 \ldots N^{m,-}_F\} : |\partial\Omega^{m,+}_{F,i} \cap \partial\Omega^{m,-}_{F,j}| \neq 0 \}, \tag{36}$$

the jump operator $[\![\cdot]\!]$ is defined as:

$$[\![\partial^p_n \mathbf{u}]\!] = \left(\partial^p_n \mathbf{u}^{m,+}_{F,i} - \partial^p_n \mathbf{u}^{m,-}_{F,j} \right), \tag{37}$$

and

$$[\![\partial^p_n \delta\mathbf{u}]\!] = \left(\partial^p_n \delta\mathbf{u}^{m,+}_{F,i} - \partial^p_n \delta\mathbf{u}^{m,-}_{F,j} \right), \tag{38}$$

where $\partial^k_n(\cdot)$ is the kth order normal derivative operator $\partial^k_n(\cdot) = \nabla^k(\cdot) \cdot \mathbf{n}_F$ where $\nabla^k(\cdot)$ is the kth order spatial gradient. The parameter p is the polynomial order of the approximation, i.e., the THB discretization. It should be noted that only the pth contribution is nontrivial for the THB discretization used in this work. The ghost penalty parameter is denoted $\gamma^{\mathbf{u}}_G$ and is defined as a multiple of the Young's modulus E of the considered material.

The ghost penalization for the temperature field is defined similarly as:

$$\mathcal{R}^{\mathrm{G}}_{\mathbf{T}} = \sum_{m=1}^{N_m} \sum_{F \in \mathcal{F}^m_{ghost}} \sum_{i=1}^{N^{m,+}_F} \sum_{\mathcal{J}^m_{\mathcal{F},i}} \left[\int_F \gamma^{\mathbf{T}}_G \, h^{\tilde{k}} [\![\partial^p_n \mathbf{T}]\!] [\![\partial^p_n \mathbf{T}]\!] d\Gamma \right], \tag{39}$$

where the jump operator is defined as:

$$[\![\partial^p_n \delta\mathbf{T}]\!] = \left(\partial^p_n \delta\mathbf{T}^{m,+}_{F,i} - \partial^p_n \delta\mathbf{T}^{m,-}_{F,j} \right), \tag{40}$$

and

$$[\![\partial^p_n \mathbf{T}]\!] = \left(\partial^p_n \mathbf{T}^{m,+}_{F,i} - \partial^p_n \mathbf{T}^{m,-}_{F,j} \right). \tag{41}$$

The ghost penalty parameters, $\gamma^{\mathbf{u}}_G$ and $\gamma^{\mathbf{T}}_G$, are defined as multiples of the Young's modulus E and the conductivity κ of the considered material, respectively. The parameter $\tilde{k}$ is defined as $\tilde{k} = 2(p-1)+1$ and enables control over the influence of the ghost penalty term. Due to the application of C^{p-1} continuous THB bases, only jumps in gradients of order p must be penalized. Further details on how to choose the ghost penalty term can be found in Burman and Hansbo [10]. In this work we commonly choose a penalty parameter $\gamma_G = 0.001$.

6 Numerical examples

In this section, we present 2D and 3D examples which illustrate the basic concepts and computational performance

of the proposed discretization framework. Canonical examples show that our approach yields optimal convergence rates for linear, quadratic, and cubic B-spline approximations. An example with a stress singularity illustrates the computational advantages of locally refined discretizations. Finally, the analysis of a polycrystalline micro-structure demonstrates the applicability of our framework to complex multi-material problems.

To quantify the accuracy of the XFEM analyses, we define the error of a generic vector state variable field $\mathbf{v}$ in the L^2 norm and H^1 semi-norm as:

$$||\mathbf{v} - \mathbf{v}^h||_{L^2} = \sqrt{\sum_{m=1}^{N_m} \int_{\Omega^m} \left|\mathbf{v} - \mathbf{v}^h\right|^2 d\Omega}, \tag{42}$$

and

$$|\mathbf{v} - \mathbf{v}^h|_{H^1} = \sqrt{\sum_{m=1}^{N_m} \int_{\Omega^m} \left|\nabla\mathbf{v} - \nabla\mathbf{v}^h\right|^2 d\Omega}, \tag{43}$$

where $\mathbf{v}^h$ is the approximate field and $\mathbf{v}$ is the reference solution which is either the analytical solution if available or a solution computed on a sufficiently refined discretization.

The systems of discretized governing equations are solved by the direct solver PARDISO for 2D problems, see Kourounis et al. [28]. A Generalized Minimal Residual (GMRES) method in combination with a dual threshold incomplete LU factorization with a degree of fill of 5.0 is used for 3D problems, see Saad [41]. The GMRES iterations are terminated if a relative drop of 10^{-10} of the normalized linear residual is achieved. All geometric and material parameters are given with each example in self-consistent units unless stated otherwise.

In the following subsections, we first present examples of single- and multi-material problems considering either a thermal or mechanical response. These examples characterize the fundamental features of the proposed analysis framework within single-physics settings. The last two examples consider coupled thermo-elastic single- and multi-material problems.

6.1 Two-material elastic bar

This example studies the convergence rates of the proposed immersed B-spline discretization framework. We consider the 3D bar shown in Fig. 15. The bar has the dimension $1.0 \times 0.5 \times 0.5$. The left face of the bar is clamped, and the bar experiences a body load in axial direction $b_x = 2x^2$. The bar is composed of two linear elastic, isotropic materials, separated by an interface which is inclined against the x-axis. To facilitate comparison against an analytical solution, both materials are assigned the same properties: a Young's modulus $E = 1.0$ and a Poisson's ratio $\nu = 0.0$. Note that this setup allows for a 1D analytical model. The analytical displacement in x-direction is:

$$u(x) = \frac{L}{6EA}\left(4L^3x - x^4\right), \tag{44}$$

where $L = 1.0$ is the bar length and $A = 0.25$ the cross-sectional area.

To study the influence of the intersection configuration on the finite element solution with B-spline background mesh refinement, we rotate the interface in ten steps between $\frac{\pi}{4}$ and $\frac{\pi}{2}$ degrees. For each orientation, we increase the number of background mesh elements from $8 \times 4 \times 4$ to $128 \times 64 \times 64$ B-spline elements through uniform mesh refinement. We repeat this study for linear, quadratic, and cubic B-spline discretizations.

In Fig. 16, we plot the mean error for all intersection configurations and for each approximation order over the B-spline element edge length as described in Eqs. (42) and (43). The error bars represent the standard deviation for all intersection configurations per order and mesh refinement. For each B-spline order, we visualize the convergence rate by the triangles inserted in Fig. 16. The numerical results show that the convergence rates of the L^2 error norm agree with the theoretical, optimal convergence rates of $p+1$, where p is the polynomial B-spline order. Similarly, the theoretical, optimal convergence rate of p, as determined by Evans et al. [20], Remacle et al. [39], is achieved in the H^1 semi-norm for linear, quadratic, and cubic discretizations. Our results suggest that the proposed immersed finite element approach recovers optimal convergence rates for sufficiently smooth state variable fields with uniform mesh refinement.

6.2 Single-material thermal diffusion problem

Analysis problems often include regions where the state variable fields exhibit large spatial gradients. To reduce the local and global approximation errors, a fine discretization is needed in these regions. The following thermal diffusion example demonstrates the ability of the proposed immersed finite element framework to construct and locally refine discretizations.

We solve a thermal diffusion problem as described in Eq. (22) in an L-shaped domain, assuming a single-material with an isotropic conductivity $\kappa = 1$, see Fig. 17. For this problem, the exact solution in polar coordinates is given as [17]:

$$T(r,\theta) = r^{\frac{2}{3}} \sin\left(\frac{2}{3}\theta\right) \tag{45}$$

The exact solution is enforced weakly to the entire boundary of the physical domain. The origin of the coordinate system is

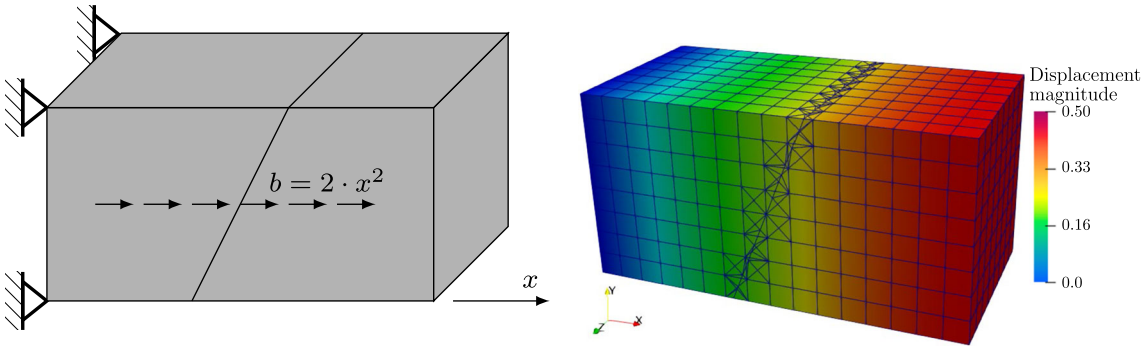


Fig. 15 Three dimensional bar problem: problem setup (left); contours of displacement magnitude evaluated on coarsest discretization (right)

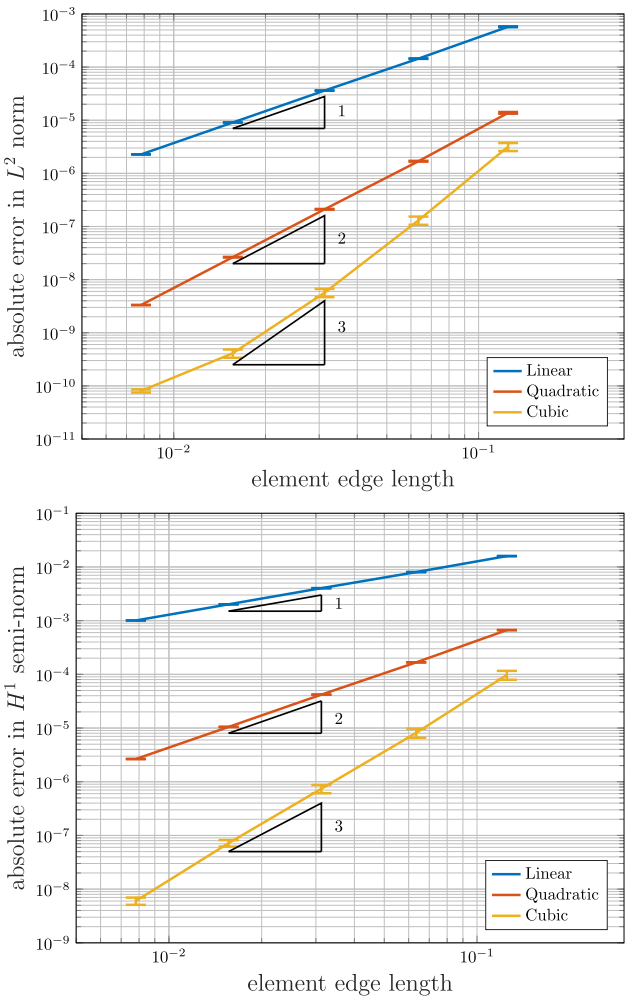


Fig. 16 Convergence rates in the L^2 error norm and H^1 semi-norm for the presented two-material bar problem

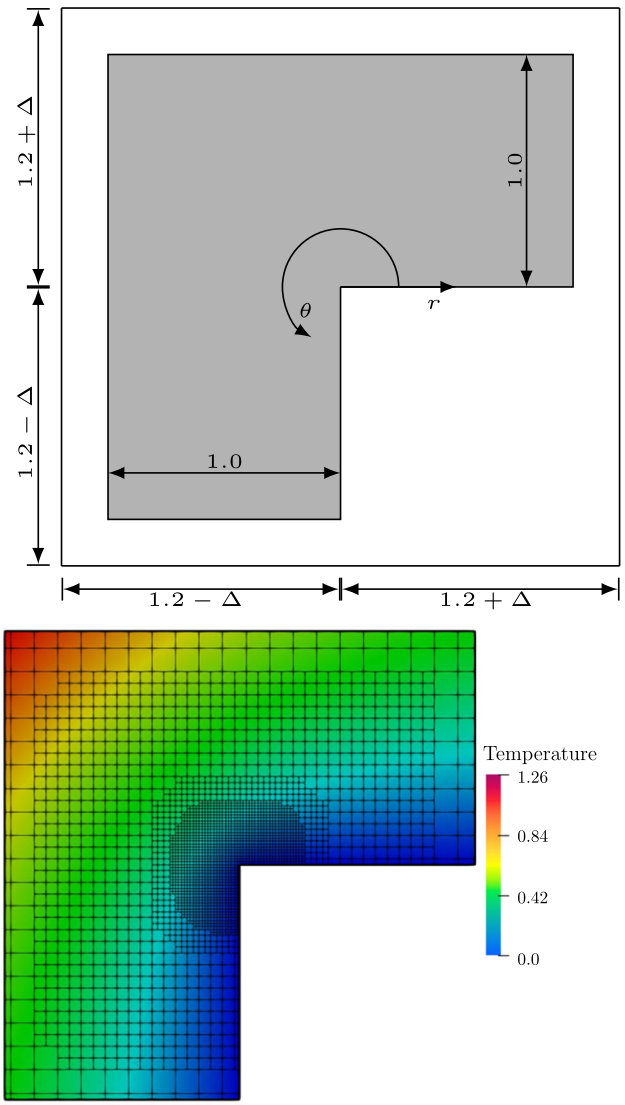


Fig. 17 Single-material thermal diffusion problem: L-shaped domain immersed into a rectangular computational domain (top); locally refined B-Spline discretization around the reentrant corner and temperature field (bottom)

located at the reentrant corner. Note that the spatial gradients increase for $r \to 0$ and are infinite at $r = 0$.

Using the proposed analysis framework, the physical domain is immersed into a rectangular computational domain as shown in Fig. 17. We compare the convergence rates for

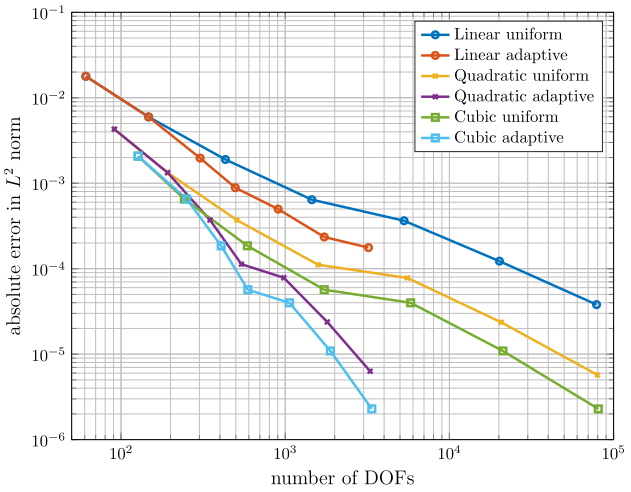


Fig. 18 Error convergence rate of temperature field with mesh refinement in the L^2 norm

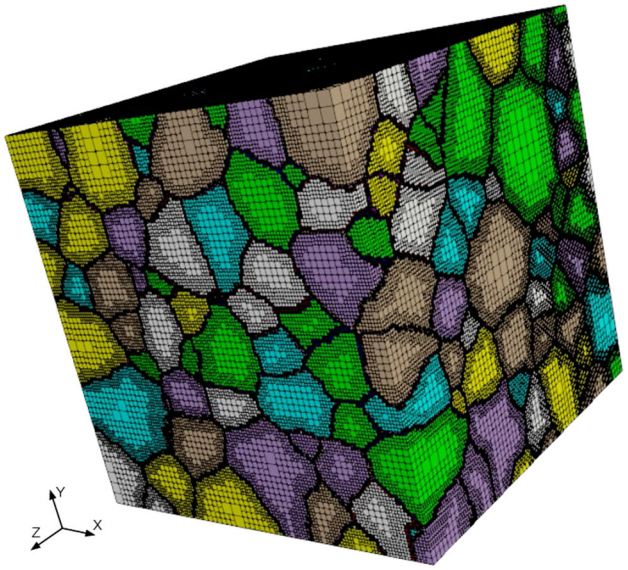


Fig. 19 Locally refined XFEM analysis model of a polycrystalline micro-structure

uniform h-refinement and for local h-refinement around the reentrant corner.

The coarsest THB background mesh has 6×6 elements. Sequences of THB discretizations are constructed by uniform or local refinement with up to 6 refinement levels, see Sect. 4. The offset Δ between the computational and physical domains is chosen as $\Delta = -0.127$. This offset guarantees that the L-shaped domain boundary does not align with the THB background mesh for any mesh refinement configuration.

For different B-spline order and mesh refinements, we plot the error in the L^2 norm versus the number of Degrees Of Freedom (DOFs) in Fig. 18. Less DOFs are required to meet a specific error requirement for local refinement when compared to uniform refinement. This suggests that local refinement may lead to a significant reduction in computational cost. Due to the singularity at the reentrant corner, optimal convergence rates with mesh refinement cannot be recovered. Moreover, this example demonstrates that the proposed framework allows for combining local h-refinement with higher order B-spline discretizations. For example, using cubic B-spline and local refinement leads to the lowest error for any number of DOFs. We do not present H^1 semi-norm error plots as they are not meaningful due to the singularity at the reentrant corner.

6.3 Polycrystalline micro-structure

The proposed analysis framework is suited to model complex multi-material problems. While the previous examples established accuracy for single-material problems, this example demonstrates the multi-material capabilities of our approach and studies the computational cost associated with the generation of THB and union background meshes.

We consider a representative volume element of a polycrystalline micro-structure and analyze its structural response. The edge length of the volume element is 150 μm. The grain geometries as well as the grain material are defined through a 3D image file with $149 \times 149 \times 149$ voxels; see, for example, Rodgers [40]. The voxels define 471 individual grains. To demonstrate the multi-material capabilities, unique material properties are assigned to each grain. The Young's modulus E varies in the range of $[50 \cdot 10^3 \ \cdots \ 500 \cdot 10^3]$ MPa and the Poisson ratio ν in a range of $[0.25 \ldots 0.35]$.

From the 3D image, we generate a level-set field for each phase such that the grain geometries are represented by the zero isocontours of the level-set fields. The grain geometries are immersed into a cubic domain, as illustrated in Fig. 19. Note that the voxel-based grain shapes could be preprocessed to obtain smoother grain interfaces. However, this option is not utilized here to demonstrate the ability of the proposed analysis framework to operate directly on complex non-smooth voxel-based geometries.

The mechanical behavior of the polycrystalline micro-structure is modeled by linear elasticity with an isotropic material behavior. Perfect bonding is assumed at the grain interfaces. To demonstrate the ability of performing an XFEM analysis for this micro-structure, we apply a pressure load of $F = $ -4.0 MPa on the $z = 0$ face and a zero-displacement Dirichlet boundary condition on the opposite face. The initial uniform B-spline background mesh has $32 \times 32 \times 32$ elements. In addition, two local refinement steps are performed at all grain boundaries. The displacement magnitude and Von Mises stress contours are shown in Fig. 20.

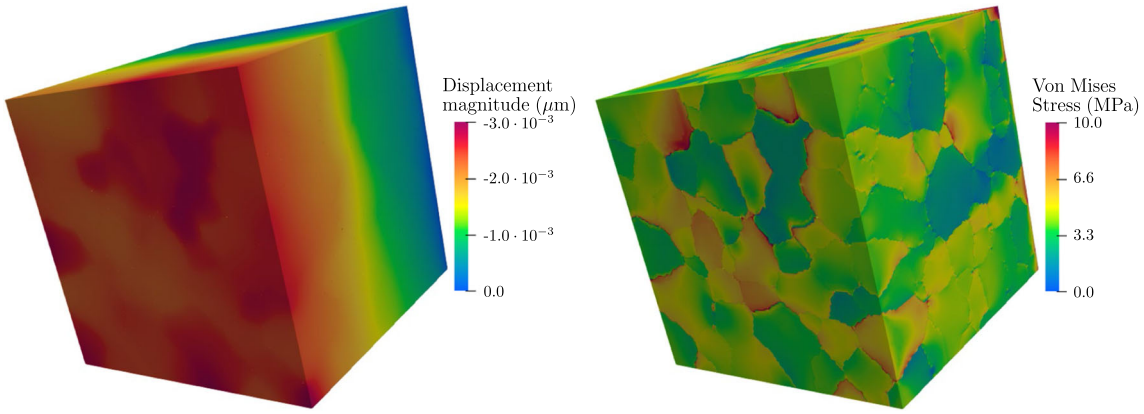


Fig. 20 XFEM analysis of polycrystalline micro-structure: displacement magnitude contours (left); Von Mises stress contours (right)

6.3.1 Scalability study

The PT data structure and domain decomposition strategy presented in Sect. 4.4 enable an efficient, parallel implementation of the proposed discretization framework. With the polycrystalline example, we study the overall performance and parallel scalability of this implementation.

For the scalability study, we start with $200 \times 200 \times 200$ PT cells at refinement level 0 and refine locally twice around the grain boundaries. For the demonstration of the scalability, we purposefully exploit a finer mesh than as in the previous analysis to guarantee a large ratio of interior to aura PT cells on each processor local subdomain for a large processor count. A small ratio of interior to aura elements negatively impacts the scalability due to increased inter-processor communication.

The resulting THB and union background meshes have each a total of 105, 558, 692 elements. The creation of the PT data structure and the derived discretizations can be subdivided into five distinct steps, see also Sect. 4. These steps include the refinement of the PT cells, the construction of the THB and union background meshes, the construction of the extraction operators, and the construction of facets needed for ghost stabilization.

To characterize the scalability of our implementation, we generate the THB and union background meshes in parallel, varying the number of subdomains from 4 to 160. Figure 21 shows the execution time for the mesh generation only, i.e., the time needed for XFEM analyses is omitted. The computations are preformed on four Intel Xeon Platinum 8160 "Skylake" nodes with 24 cores each, distributing the subdomains equally across all cores.

Mostly linear scaling with the processor count is observed for the refinement operation and the construction of the projection operators. Such behavior is expected as these steps do not need any inter-process communication. The construction of THB and union background meshes, as well as the construction of facets, show a linear scaling for a low processor count. They start to plateau with a high processor count, as these steps require communication of IDs in the aura. The aura size is based on refinement level l^0 and the buffer size as outlined in Sect. 4.4. Increasing the processor count while keeping the total domain size constant increases the ratio of aura PT cells to interior cells. This consequently increases communication and affects scalability.

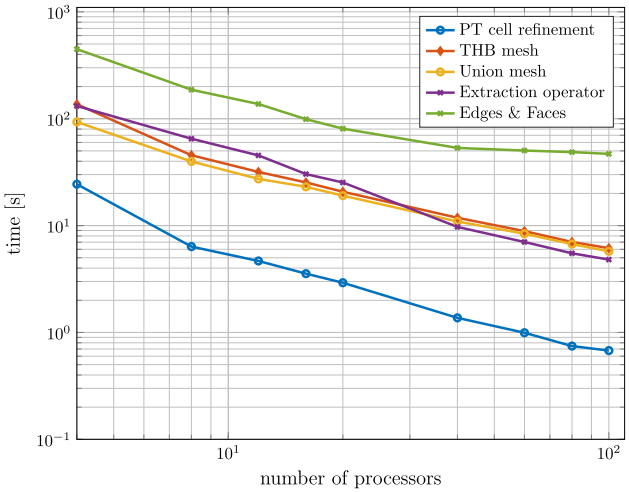


Fig. 21 Scalability study with processor count of the hierarchical meshing library, using $200 \times 200 \times 200$ PT cells at refinement level 0 and two refinement steps at grain boundaries

The scalability study demonstrates that a mesh with over 10^8 elements can be generated on just four nodes with a total of 96 processors in less than 50 s. Furthermore, when creating a higher order discretizations, the computational time for the refinement of the background meshes and the construction of faces stays the same as neither are affected by the interpolation order. The construction of the discretizations, as well as the calculation of the extraction operators, needs slightly more memory and computational time as the support of higher order basis functions is increased.

6.4 Thermo-elastic plate with elliptic hole

This example considers a single-material but multi-field configuration. We study the problem of a thermo-elastic plate with an elliptical inclusion under in-plane tension due to a thermal load as illustrated in Fig. 22. By exploiting symmetry, we only model a quarter of the domain. The setup consists of a solid, two-dimensional, square domain with a length of $L = 2.0$ and an elliptical inclusion at the origin with a semi-major axis of $A = 0.8136$ and a semi-minor axis of $B = 0.5753$. These values for the semi-axes are chosen such that the immersed interface does not align with the THB background mesh for all refinement levels. The temperature field is due to a uniform heat flux $q = 10.0$ applied along the elliptical hole and a prescribed temperature $T = 1.0$ at the right domain boundary. The matrial conductivity is $\kappa = 1.0$. The structural response is described by a linear thermo-elastic model and an isotropic constitutive behavior, with a Young's modulus $E = 1.0$, a Poisson's ratio $\nu = 0.3$, and a coefficient of thermal expansion $\alpha = 1.0$. The reference temperature is set to $T_0 = 0.0$.

To study the convergence rate of the XFEM solution with mesh refinement, we evaluate the thermal and structural response on a series of B-spline discretizations. Starting from the coarsest THB background mesh with 10×10 elements, uniformly refined B-spline discretizations with 20×20, 40×40, 80×80, 160×160 and 320×320 elements are considered. The geometry of the elliptical inclusion is defined by an analytical level set function. A reference solution is computed numerically by calculating both fields on a uniformly refined B-spline mesh that is 7 times refined, 1280×1280 compared to the coarsest background mesh. Numerical studies lead by the authors have shown that the computation of the reference solution on a finer mesh does not noticeably improve the quality of the error computation.

For all B-spline discretizations, the union background mesh is 7 times uniformly refined relative to the coarsest background mesh. This eliminates the geometrical error between the coarse and the reference solution. Moreover, it simplifies the integration of the L^2 and H^1 errors as both the coarse and the reference solutions are represented on equally refined XFEM background meshes.

To illustrate the interdependence of discretization errors in multi-physics problems, we vary the refinement for the temperature discretization while the displacement field is always evaluated on the finest B-spline discretization, i.e., 320×320 elements. The contours of the absolute errors of the temperature and displacement fields are visualized in Fig. 23 for the case where the temperature field is discretized by linear B-splines on a 20×20 element mesh. This coarse temperature field discretization leads to errors in the interior of the elements. Therefore, the error visualization presents the characteristic tiled error patterns.

The L^2 and H^1 semi-norm errors of the temperature and the displacement fields are presented in Fig. 24. First, we consider the error of the temperature field. The convergence of the L^2 error norm for linear, quadratic, and cubic B-spline discretizations shows optimal convergence rate of $p+1$. For the H^1 semi-norm, we observe optimal convergence rates of p. As expected, using a higher-order basis functions results in smaller errors for the temperature field.

Next, we consider the error of the displacement field, which is evaluated with linear, quadratic, and cubic basis functions and depends on the temperature field that is discretized by either linear, quadratic, and cubic B-splines. The errors of the displacement field in the L^2 norm and H^1 semi-norm are presented in Fig. 24. The convergence rate of the error with mesh refinement in the L^2 norm is independent of the order of the displacement field basis functions and instead only depends on the order of the temperature field. Since the displacement field is discretized on a much finer mesh than the thermal field, the error of the displacement field is dominated by the error of the thermal field. The same observation can be made for the convergence rate in the H^1 semi-norm.

To gain further insight, we refine simultaneously the thermal and displacement fields with a specific difference in refinement level between both fields. The THB background element edge lengths of the thermal and displacement fields are denoted by H and h, respectively. The study is performed for a maximal difference in element size of $H = 4h$. This study is performed for linear, quadratic, and cubic interpolation orders for the thermal and the displacement fields. The setup of the thermal problem is identical the one presented above, see Fig. 24. The convergence rate of the displacement field with mesh refinement in the L^2 and H^1 semi-norm is presented in Fig. 25. We observe that the absolute error and convergence rate of the displacement problem only shows minimal differences when choosing an up to two times coarser thermal field than the displacement field.

Lastly, we present a convergence study where we allow for a difference in polynomial order. The thermal and displacement field are refined simultaneously such that both fields are on the same refinement level, i.e., $H = h$. Results are presented for a polynomial order of the displacement field p_U for $p_U = p_T$ and $p_U = p_T + 1$, where p_T is the order of the thermal field. The convergence rates with mesh refinement of the displacement field in the L^2 and H^1 semi-norm are presented in Fig. 26. We observe that choosing the thermal field one polynomial order lower than the displacement field results in the same convergence rate of the displacement field in the H^1 semi-norm. For this particular problem, theoretical results are not available for the rate of the convergence for the displacement field in the L^2 norm, as the well-known Aubin-Nitsche technique cannot be applied in the thermo-elastic setting.

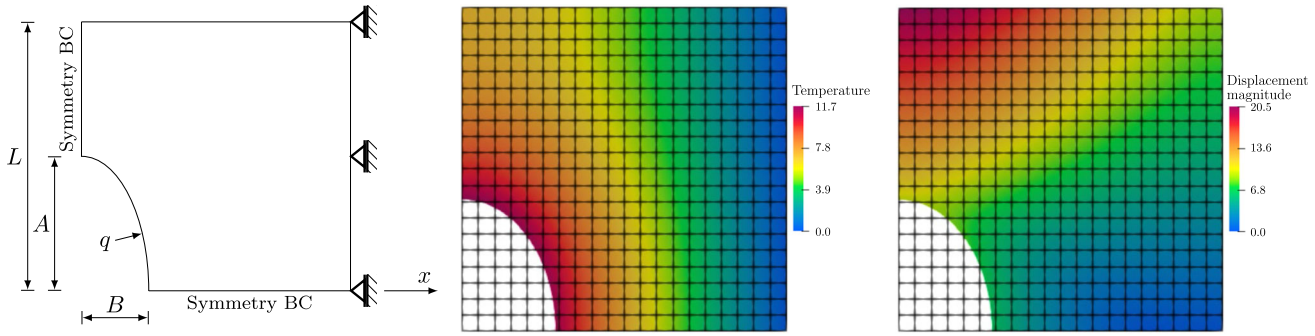


Fig. 22 Thermo-elastic plate with elliptic hole under thermal load: problem setup (left); temperature contours (center); displacement magnitude contours (right)

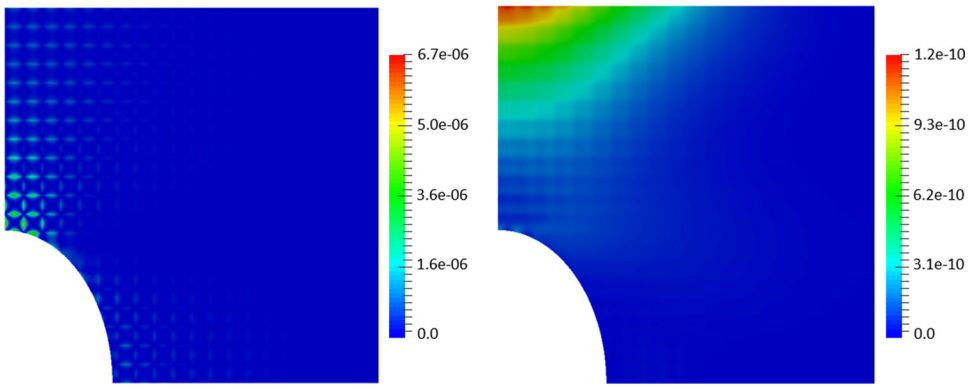


Fig. 23 Visualization of absolute error of temperature (left) and displacement field (right)

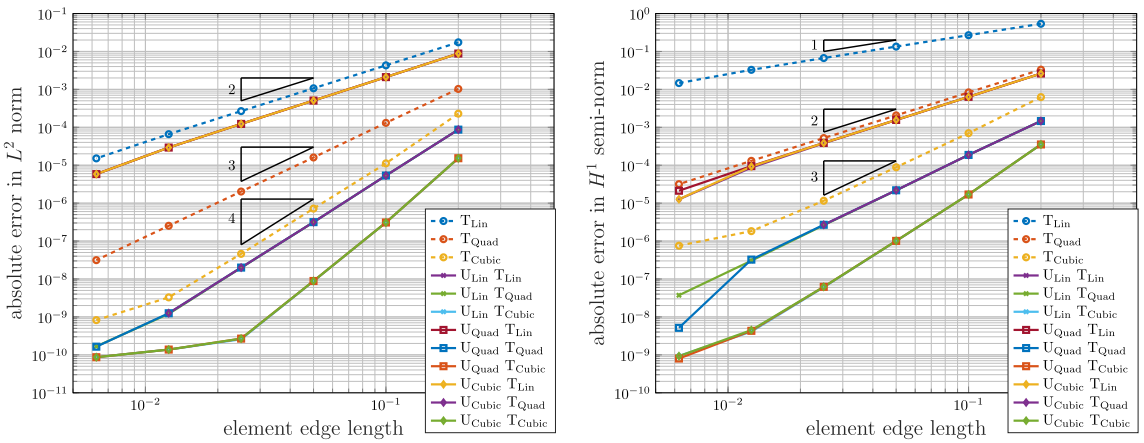


Fig. 24 Error convergence rate of temperature and displacement fields with mesh refinement in the L^2 norm (left) and H^1 semi-norm (right)

This example illustrates that for multi-physics problems, the discretization of individual fields needs to be performed in concert. The proposed discretization framework provides an efficient and flexible tool to select the interpolation order and mesh refinement level for individual fields to obtain a numerical solution that meets accuracy requirements with minimal computational costs, i.e., with minimal number of DOFs.

6.5 Thermo-elastic multi-material problem

This final example studies a multi-material, multi-physics configuration. For such problems, areas with large spatial gradients can vary based on the type of physics and material. Differently locally refined discretizations for each field reduce the computational cost while simultaneously enabling for an accurate evaluation of the physical responses. To demonstrate this aspect, we consider the two-material, thermo-elastic problem of an expanding circle embedded in a non-expanding plate, as presented in the introduction in

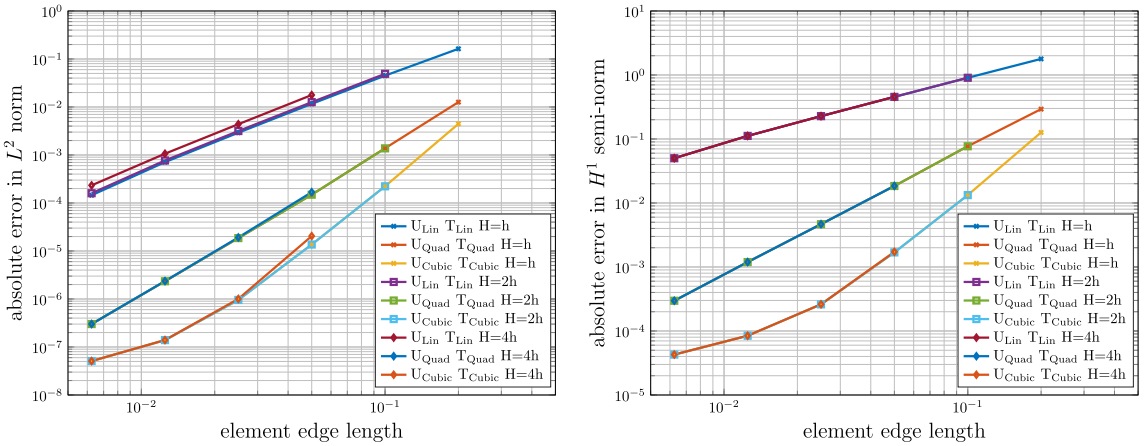


Fig. 25 Error convergence rate the displacement field with mesh refinement. L^2 norm (left) and H^1 semi-norm (right). Temperature and displacement field are refined simultaneously

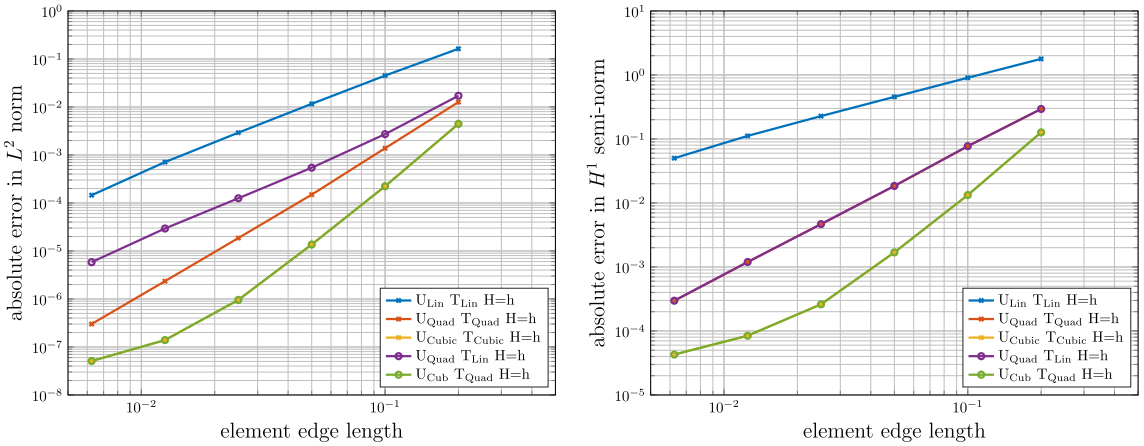


Fig. 26 Error convergence rate the displacement field with mesh refinement in the L^2 norm (left) and H^1 semi-norm (right) using identical refinements for the temperature and displacement fields

Fig. 1a. The circular inclusion is occupied by a material A and the plate by a material B. The temperature field is due to a spatially varying heat load of $q = 100.0 \cdot \sin(10.0 \cdot y) + 110.0$ applied along the left domain boundary and a prescribed temperature $T = 0.0$ at the right domain boundary. The conductivities of materials A and B are identical and are $\kappa_A = \kappa_B = 1.0$. The structural response of both materials is described by a linear thermo-elastic model and an isotropic, constitutive behavior, with Young's moduli $E_A = E_B = 1.0$, Poisson's ratios $\nu_A = \nu_B = 0.3$ and coefficients of thermal expansion $\alpha_A = 1.0 \cdot 10^{-5}$ and $\alpha_B = 10.0 \cdot 10^{-5}$. The reference temperature is set to $T_0 = 0.0$. Perfect bonding is assumed at the interface between the two materials Γ^{AB}.

An XFEM analysis is performed to approximate the stress field at the material interface with high accuracy. We compare the temperature, displacement, Von Mises stress, and heat flux using uniformly and locally refined B-spline discretizations. A quadratic polynomial order is used for the temperature and displacement fields. The coarsest THB background mesh has 20×10 elements. For the uniform refinement case, the discretizations of the temperature and displacement fields are three times uniformly refined. In the local refinement case, the discretization for the temperature field is three times locally refined at the left domain boundary to accurately represent the spatially varying load. In contrast, the discretization of the displacement field is three times locally refined around the circular material interface. The obtained heat flux and stress distributions are presented in in Fig. 27 and show that the temperature and displacement fields exhibit large spatial gradients in these regions.

The number of DOFs resulting from the uniformly and locally refined THB background meshes are presented in Table 1. The locally refined discretization of the thermal field reduces the number of DOFs by a factor of ~ 17 when compared to the uniform discretization. When locally refining the displacement field, the number of DOFs is ~ 4.6 times smaller compared to the linear system associated with a uniform discretization. To accurately represent the geometry,

Table 1 Comparison of the number of DOFs for temperature and displacement field for and locally uniformly refined discretizations

	Local refinement	Uniform refinement
$\#\text{DOFs}_{\text{Temp}}$	818	13,992
$\#\text{DOFs}_{\text{Disp}}$	6020	27,981

Table 2 Comparison of error of the stress field at the interface Γ^{AB} in the L^2 error norm for different refinement configurations

Local Refinement	L^2 error norm	# DOFs
(a) Both fields at Γ^{AB} and Γ^{N}	1.2023e−6	10, 626
(b) Disp at Γ^{AB} and Temp at Γ^{N}	2.8881e−6	6838
(c) Disp at Γ^{AB}	2.8913e−6	5764
(d) Temp at Γ^{N}	3.8764e−4	2126

the union background mesh is four times uniformly refined, irrespective of the B-spline discretizations of the temperature and displacement fields.

The displacement and temperature fields, as well as the Von Mises stress and the heat flux magnitude contours, are shown in Fig. 27. Qualitatively, the resulting fields for uniform and local refinement are equivalent.

The following four locally refined configurations are examined: (a) refinement for both the structural and thermal field at Γ^{AB} and at the Neumann boundary Γ^{N}; (b) refinement of the structural field at Γ^{AB} and refinement of the thermal field at Γ^{N}; (c) refinement of the structural field at Γ^{AB}; and (d) refinement of the thermal field at Γ^{N}. In all cases the union background mesh is chosen to be uniformly refined.

Table 2 presents the L^2 error norm of the Von-Mises stress at the material interface Γ^{AB} for different local refinement configurations together with the size of the linear system. Local refinement for the first two Configurations, (a) and (b), yields the smallest error in the L^2 error norm. However, Configuration (b) results in a significant reduction of the computational cost as measured by the number of DOFs.

In addition, Configuration (c) yields a similar small error as Configuration (a) and (b). This behavior might be unexpected but can be explained with the fine uniform refinement of the union background mesh which allows for an accurate integration of the spatial varying load even with a coarse thermal discretization. Configuration (d) results in the highest error as only the thermal field is locally refined at the Neumann boundary.

7 Conclusion

This paper presents an immersed isogeometric finite element analysis framework with local mesh refinement based on a Heaviside enriched XFEM. Hierarchical, locally refined discretizations allow for refinement of the finite element approximations in regions of interest, balancing accuracy and computational cost. THB functions are utilized as they provide an elegant way to construct suitable, locally refined discretization spaces. Moreover, B-spline basis functions are an appealing choice over Lagrange basis functions because of their higher inter-element continuity and their increased computational efficiency. In multi-material, multi-physics problems, the resolution requirements may be different for individual state variable fields. The proposed framework allows for separate discretizations with different polynomial orders for each physical field. Furthermore, each discretization can be refined individually, both globally and locally, to meet field-specific accuracy requirements. In contrast to using the same polynomial order and refinement for all state variable fields, the proposed framework may lower significantly the computational cost. In this paper, THB background meshes are refined based on geometric refinement indicators. However, the framework permits any refinement indicators and can be used for adaptive mesh refinement strategies.

A PT data structure and mesh generation algorithms are presented for the efficient construction of differently refined meshes, both in terms of run time performance and memory needs. The concept of PT cell activation states enables using the same data structure to construct a set of different hierarchically refined THB background meshes. Using the PT data structure and the activation state concept, a union background mesh is constructed such that elements of the union background mesh are at the highest (or higher) refinement level of all corresponding elements of the THB background meshes. The union mesh supports the discretizations of all THB background meshes.

The union background mesh serves as the XFEM background mesh in which the geometry is immersed. In this paper, the geometry is represented by level set functions, and intersected elements are cut recursively by the zero isocontours of the level set fields. This process yields a single integration mesh which is aligned with the boundaries and interfaces defined by the level set functions. In this paper, the weak form of the governing equations is integrated by standard quadrature rules on the integration mesh. However, other quadrature schemes can be applied, such as the ones proposed by Thiagarajan and Shapiro [49] and Gunderman et al. [25]. The B-spline basis functions are enriched using the generalized Heaviside enrichment strategy of Noël et al. [36]. To facilitate operations performed on the union background mesh and on the integration mesh in the XFEM analysis, the THB basis functions are represented by Lagrange basis func-

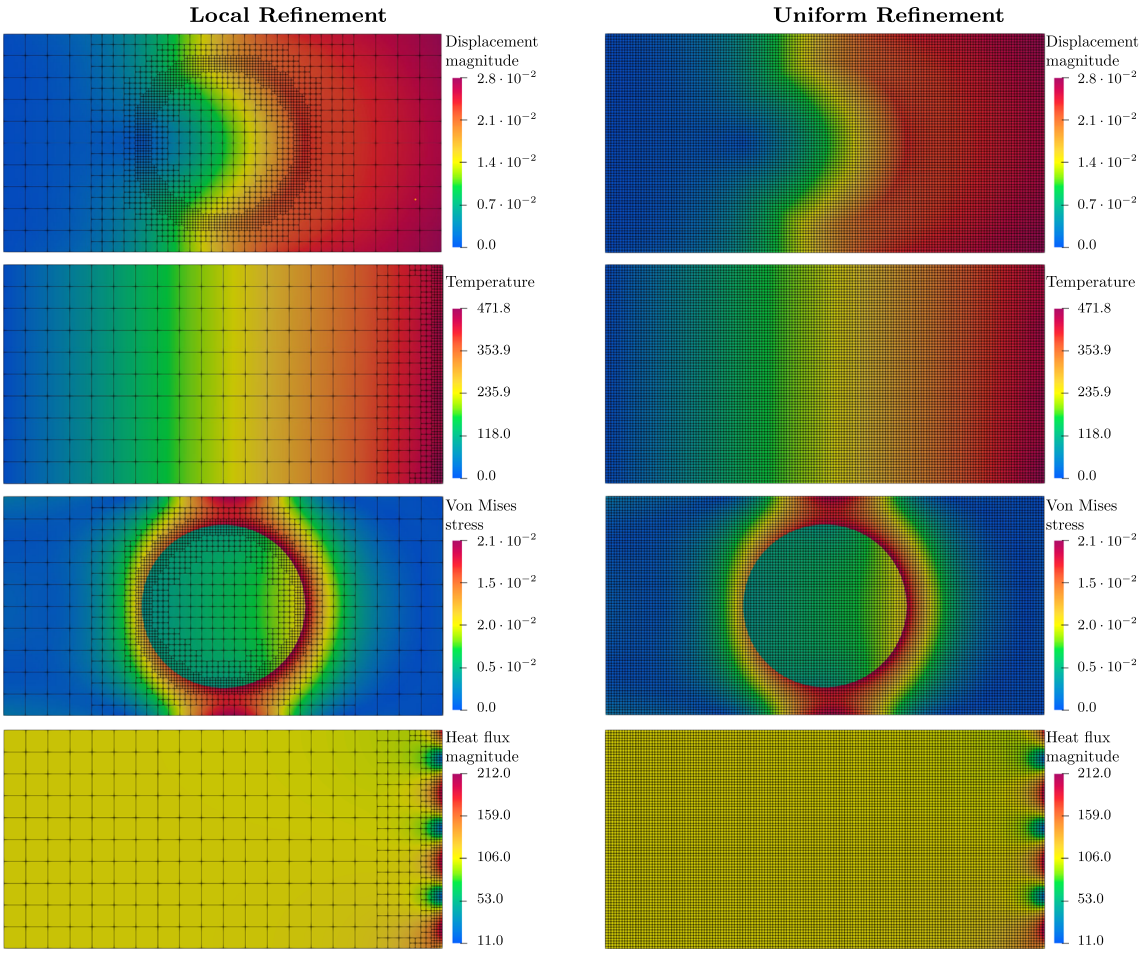


Fig. 27 Comparison of XFEM analysis results for locally and uniformly refined discretizations

tions defined on the union background mesh via extraction operators. The union background mesh supports extraction operators for each THB discretization.

Numerical examples suggest that the proposed immersed isogeometric finite element framework generates discretizations that converge at theoretical, optimal convergence rates with mesh refinement. The application of our framework to a multi-material polycrystalline micro-structure shows that it is well suited to discretize complex multi-material problems in 3D. A scalability study demonstrates that the proposed implementation scales with an increasing number of processors. The coupled thermo-elastic examples highlight the benefits of tailoring the discretization of individual state variable fields to their field-specific accuracy requirements.

Future work will focus on extending the proposed framework to utilize additional refinement criteria such as finite element error estimators. Moreover, more complex physics for which local mesh refinement is crucial, such as fluid flow at high Reynolds numbers, will be addressed.

Acknowledgements The first, second, fourth, and fifth authors acknowledge the support for this work from the Defense Advanced Research Projects Agency (DARPA) under the TRADES program (Agreement HR0011-17-2-0022). The first author acknowledges partial auspice of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-JRNL-842737). The third and fifth authors acknowledge the support of Sandia National Laboratories under PO 2120843. The fourth and fifth authors acknowledge the support of the National Science foundation under Grant 2104106. The opinions and conclusions presented in this paper are those of the authors and do not necessarily reflect the views of the sponsoring organization.

References

1. 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
2. Babuška I, Melenk JM (1997) The partition of unity method. Int J Numer Methods Eng 40:727–758. https://doi.org/10.1002/(SICI)1097-0207(19970228)40:4<727::AID-NME86>3.0.CO;2-N
3. Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJ, Lipton S, Scott MA, Sederberg TW (2010) Isogeometric analysis using

t-splines. Comput Methods Appl Mech Eng 199:229–263. https://doi.org/10.1016/j.cma.2009.02.036
4. Belytschko T, Black T (1999) Elastic crack growth in finite elements with minimal remashing. Int J Numer Methods Eng 45:601–620
5. Belytschko T, Gracie R, Ventura G (2009) A review of extended/generalized finite element methods for material modeling. Model Simul Materials Sci Eng 17:66. https://doi.org/10.1088/0965-0393/17/4/043001
6. Bracco C, Buffa A, Giannelli C, Vázquez R (2019) Adaptive isogeometric methods with hierarchical splines: an overview. Discrete Continu Dyn Syst Ser A 39:241–262. https://doi.org/10.3934/dcds.2019010
7. Buffa A, Giannelli C (2017) Adaptive isogeometric methods with hierarchical splines: optimality and convergence rates. Math Models Methods Appl Sci 27:2781–2802. https://doi.org/10.1142/S0218202517500580
8. Burman E (2010) La pénalisation fantôme. Comptes Rendus Mathematique 348:1217–1220. https://doi.org/10.1016/j.crma.2010.10.006
9. Burman E (2012) A penalty-free nonsymmetric Nitsche-type method for the weak imposition of boundary conditions. SIAM J Numer Anal 50:1959–1981. https://doi.org/10.1137/10081784X
10. Burman E, Hansbo P (2014) Fictitious domain methods using cut elements: III. A stabilized Nitsche method for stokes' problem. ESAIM Math Model Numer Anal 48:859–874. https://doi.org/10.1051/m2an/2013123
11. D'Angella D, Kollmannsberger S, Rank E, Reali A (2018) Multi-level bézier extraction for hierarchical local refinement of isogeometric analysis. Comput Methods Appl Mech Eng 328:147–174. https://doi.org/10.1016/j.cma.2017.08.017
12. de Boor C (1972) On calculating with B-splines. J Approx Theory 6:50–62
13. de Prenter F, Verhoosel CV, van Zwieten GJ, van Brummelen EH (2017) Condition number analysis and preconditioning of the finite cell method. Comput Methods Appl Mecha Eng 316:297–327. https://doi.org/10.1016/j.cma.2016.07.006
14. Divi SC, Verhoosel CV, Auricchio F, Reali A, van Brummelen EH (2020) Error-estimate-based adaptive integration for immersed isogeometric analysis. Comput Math Appl 80:2481–2516. https://doi.org/10.1016/j.camwa.2020.03.026
15. Duarte CA, Babusï I, Oden JT (2000) Generalized finite element methods for three-dimensional structural mechanics problems. Comput Struct 77:215–232
16. Düster A, Parvizian J, Yang Z, Rank E (2008) The finite cell method for three-dimensional problems of solid mechanics. Comput Methods Appl Mech Eng 197:3768–3782. https://doi.org/10.1016/j.cma.2008.02.036
17. D'Angella D, Zander N, Kollmannsberger S, Frischmann F, Rank E, Schröder A, Reali A (2016) Multi-level hp-adaptivity and explicit error estimation. Adv Model Simul Eng Sci. https://doi.org/10.1186/s40323-016-0085-5
18. Elfverson D, Larson MG, Larsson K (2018) Cutiga with basis function removal. Adv Model Simul Eng Sci 5:1–19. https://doi.org/10.1186/s40323-018-0099-2
19. Embar A, Dolbow J, Harari I (2010) Imposing Dirichlet boundary conditions with Nitsche's method and spline-based finite elements. Int J Numer Methods Eng 83:877–898. https://doi.org/10.1002/nme.2863
20. Evans JA, Bazilevs Y, Babuška I, Hughes TJ (2009) n-widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. Comput Methods Appl Mech Eng 198:1726–1741. https://doi.org/10.1016/j.cma.2009.01.021
21. 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
22. Gerstenberger A, Wall WA (2008) An extended finite element method/Lagrange multiplier based approach for fluid–structure interaction. Comput Methods Appl Mech Eng 197:1699–1714. https://doi.org/10.1016/j.cma.2007.07.002
23. Giannelli C, Jüttler B, Speleers H (2012) THB-splines: the truncated basis for hierarchical splines. Comput Aided Geom Des 29:485–498. https://doi.org/10.1016/j.cagd.2012.03.025
24. Giannelli C, Bert J, Speleers H (2014) Strongly stable bases for adaptively refined multilevel spline spaces. Adv Comput Math 40:459–490. https://doi.org/10.1007/s10444-013-9315-2
25. 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/10.1016/j.cad.2020.102944
26. Hughes TJ, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput Methods Appl Mech Eng 194:4135–4195. https://doi.org/10.1016/j.cma.2004.10.008
27. Höllig K, Reif U, Wipper J (2001) Weighted extended b-spline approximation of Dirichlet problems. SIAM J Numer Anal 39:442–462. https://doi.org/10.1137/s0036142900373208
28. Kourounis D, Fuchs A, Schenk O (2018) Toward the next generation of multiperiod optimal power flow solvers. IEEE Trans Power Syst 33:4005–4014. https://doi.org/10.1109/TPWRS.2017.2789187
29. Lang C, Makhija D, Doostan A, Maute K (2014) A simple and efficient preconditioning scheme for heaviside enriched XFEM. Comput Mech 54(5):1357–1374. https://doi.org/10.1007/s00466-014-1063-8
30. Makhija D, Maute K (2014) Numerical instabilities in level set topology optimization with the extended finite element method. Struct Multidisc Optim 49:185–197. https://doi.org/10.1007/s00158-013-0982-x
31. Marussig B, Hughes TJ (2018) A review of trimming in isogeometric analysis: challenges, data exchange and simulation aspects, vol 25. Springer, Netherlands, p 1183101792209. https://doi.org/10.1007/s11831-017-9220-9
32. Mittal R, Iaccarino G (2005) Immersed boundary methods. Annu Rev Fluid Mech 37:239–261. https://doi.org/10.1146/annurev.fluid.37.061903.175743
33. Nguyen VP (2012) An introduction to standard and enriched isogeometric analysis through a simple Matlab code isogeometric analysis: a review and computer implementation aspects. Math Comput Simul 117:89–116. https://doi.org/10.1016/j.matcom.2015.05.008
34. Nitsche J (1971) Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg 36:9–15. https://doi.org/10.1007/BF02995904
35. Noël L, Schmidt M, Messe C, Evans JA, Maute K (2020) Adaptive level set topology optimization using hierarchical B-splines. Struct Multidiscip Optim 62:1669–1699. https://doi.org/10.1007/s00158-020-02584-6
36. Noël L, Schmidt M, Doble K, Evans JA, Maute K (2022) XIGA: an extended isogeometric analysis approach for multi-material problems. Comput Mech. https://doi.org/10.1007/s00466-022-02200-y
37. Parvizian J, Düster A, Rank E (2007) Finite cell method?: h- and p-extension for embedded domain problems in solid mechanics. Comput Mech 41:121–133. https://doi.org/10.1007/s00466-007-0173-y
38. Peskin CS (2002) The immersed boundary method. Acta Numer 11:479–517. https://doi.org/10.1017/S0962492902000077

39. Remacle J, Lambrechts J, Seny B (2012) Blossom-quad: a non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm. International 104:1102–1119. https://doi.org/10.1002/nme
40. Rodgers T (2015) Exploration of process-structure linkages in simulated additive manufacturing microstructures. Harv Dataverse. https://doi.org/10.7910/DVN/KJMK9Z
41. Saad Y (2003) Iterative methods for sparse linear systems. Soc Ind Appl Math. https://doi.org/10.1137/1.9780898718003
42. Schillinger D, Rank E (2011) An unfitted hp-adaptive finite element method based on hierarchical b-splines for interface problems of complex geometry. Comput Methods Appl Mech Eng 200:3358–3380. https://doi.org/10.1016/j.cma.2011.08.002
43. 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 Comput Methods Eng 22:391–455. https://doi.org/10.1007/s11831-014-9115-y
44. Schillinger D, Dedè L, Scott MA, Evans JA, Borden MJ, Rank E, Hughes TJR (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
45. Schillinger D, Ruthala PK, Nguyen A (2016) Lagrange extraction and projection for nurbs basis functions: a direct link between isogeometric and standard nodal finite element formulations. Int J Numer Methods Eng 66:1102–1119. https://doi.org/10.1002/nme.3279/full
46. Shemon ER, Attaway CW (2014) User manual for exodus ii mesh converter. www.anl.gov
47. Strouboulis T, Copps K, Babuska I (2000) The generalized finite element method: an example of its implementation and illustration of its performance. Int J Numer Methods Eng 47:1401–1417. https://doi.org/10.1002/(SICI)1097-0207
48. Terada K, Asai M, Yamagishi M (2003) Finite cover method for linear and non-linear analyses of heterogeneous solids. Int J Numer Methods Eng 58:1321–1346. https://doi.org/10.1002/nme.820
49. Thiagarajan V, Shapiro V (2016) Adaptively weighted numerical integration in the finite cell method. Comput Methods Appl Mech Eng 311:250–279
50. Verhoosel CV, van Zwieten GJ, van Rietbergen B, de Borst R (2015) Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. Comput Methods Appl Mech Eng 284:138–164. https://doi.org/10.1016/j.cma.2014.07.009
51. Villanueva CH, Maute K (2014) Density and level set-XFEM schemes for topology optimization of 3-D structures. Comput Mech 54:133–150. https://doi.org/10.1007/s00466-014-1027-z