



# Adaptive level set topology optimization using hierarchical B-splines

L. Noël<sup>1</sup> · M. Schmidt<sup>1</sup> · C. Messe<sup>1</sup> · J.A. Evans<sup>1</sup> · K. Maute<sup>1</sup>

Received: 25 September 2019 / Revised: 24 March 2020 / Accepted: 26 March 2020 / Published online: 29 July 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

This paper presents an adaptive discretization strategy for level set topology optimization of structures based on hierarchical B-splines. This work focuses on the influence of the discretization approach and the adaptation strategy on the optimization results and computational cost. The geometry of the design is represented implicitly by the iso-contour of a level set function. The extended finite element method is used to predict the structural response. The level set function and the state variable fields are discretized by hierarchical B-splines. While first-order B-splines are used for the state variable fields, up to third-order B-splines are considered for discretizing the level set function. The discretizations of the design and the state variable fields are locally refined along the material interfaces and selectively coarsened within the bulk phases. For locally refined meshes, truncated B-splines are considered. The properties of the proposed mesh adaptation strategy are studied for level set topology optimization where either the initial design is comprised of a uniform array of inclusions or inclusions are generated during the optimization process. Numerical studies employing static linear elastic material/void problems in 2D and 3D demonstrate the ability of the proposed method to start from a coarse mesh and converge to designs with complex geometries, reducing the overall computational cost. Comparing optimization results for different B-spline orders suggests that higher interpolation order promote the development of smooth designs and suppress the emergence of small features, without providing an explicit feature size control. A distinct advantage of cubic over quadratic B-splines is not observed.

**Keywords** Topology optimization · Level set · XFEM · Adaptive mesh refinement · Truncated hierarchical B-splines

## 1 Introduction

Following the seminal work of Bendsoe and Kikuchi (1988), topology optimization has become, over the last decades, a reliable and efficient design tool in various application fields (see Sigmund and Maute (2013) and Deaton and Grandhi (2014)). In general, an optimization problem is formulated to find the optimal material distribution, within a design domain, that maximizes a given objective while satisfying some given constraints on the geometry and/or the physical response. In specific regions of the design domains, the state variable fields may

exhibit large gradients or discontinuities. Additionally, the design variable fields may have to resolve complex shapes, intricate material arrangements, or extremely thin features. In these cases, an increased mesh resolution is necessary to achieve a sufficient accuracy of both the physical responses and the geometry description. Most topology optimization approaches operate on uniformly refined meshes, for which extreme resolution requirements can lead to a drastic increase in computational cost since the number of design and state degrees of freedom (DOFs) are increased. Hence, there is a need for efficient adaptive strategies in topology optimization.

In the 1980s, Bennet and Botkin (1985) and Kikuchi et al. (1986) explored mesh adaptation in combination with shape optimization. Their objective was to avoid the distortions of the finite element mesh arising from shape modifications and thus to control the quality of the finite element solution. Later, Maute and Ramm (1995) and Maute et al. (1998) and Schleupen et al. (2000) performed topology optimization and used separate models for the design and the analysis to allow for more flexibility in the optimization process. In these works, the effective design space is adapted at each optimization step with respect to the current material

---

Responsible Editor: Christian Gogu

✉ L. Noël  
lise.noel@colorado.edu

✉ K. Maute  
kurt.maute@colorado.edu

<sup>1</sup> Ann and H. J. Smead, Department of Aerospace Engineering Sciences, University of Colorado at Boulder, 3775 Discovery Dr, Boulder, CO 80309-0429, USA

distribution, to decrease both the number of finite elements in the analysis model and the number of design variables used for the geometry description. Ever since, adaptive topology optimization has been an active research topic, as it can provide both precise geometry representations and accurate mechanical responses at a reduced computational cost, substantially speeding up the design process.

Working with density-based approaches and solving compliance minimization problems, several adaptive strategies for topology optimization have been proposed. Costa and Alves (2003) carried out a refinement-only approach based on a density criterion. After a given number of optimization steps, the material and boundary elements are refined using  $h$ -adaptivity, i.e., the elements are split into smaller ones (see Yserentant (1986) and Krysl et al. (2003)). As no coarsening is introduced in the void regions and as no regularization is used, the optimization leads to mesh dependent designs that differ from the ones obtained with uniformly refined meshes. Stainko (2006) developed a refinement criterion based on a regularization filter indicator that locates the material interface. In this case,  $h$ -refinement is only applied around the interface and a reduced number of elements is generated during the optimization process. Extending the previous approaches, Wang et al. (2010) introduced mesh coarsening in the void regions, allowing for further cost reduction and achieving designs that only slightly differ from uniform ones. A similar approach is adopted in Nana et al. (2016) using unstructured meshes and in Nguyen-Xuan (2017) using polygonal elements. Also exploiting  $h$ -refinement, Bruggi and Verani (2011) trigger mesh adaptivity based on error estimators; one related to the geometric error and one to the analysis error.

In the previous approaches, the geometry description and the analysis are strongly coupled, as both the state and the design variables are defined on the same mesh. Following the initial idea by Maute and Ramm (1995), Guest and Smith Genut (2010) exploited the separation of the geometry and the analysis fields using Heaviside projection methods. After projection, multiple elements are influenced by a single design variable, introducing a redundancy in the design. The latter can be exploited to reduce the number of design variables, and therefore enhance computational performance. Wang et al. (2013, 2014) also refined the state and design fields separately by using two distinct meshes adapted by  $h$ -refinement and resorting to independent geometric and analysis error estimator criteria.

Later on, adaptive mesh refinement strategies were extended to other optimization techniques by, for example Wallin et al. (2012) in the context of phase fields and Panesar et al. (2017) for Bi-directional Evolutionary Structural Optimization (BESO). Adaptive mesh refinement has also been used to address stress-based optimization

problems. Salazar de Troya and Tortorelli (2018) used a density-based approach and adapted the mesh following a stress error estimator, while Zhang et al. (2018) used moving morphable voids described explicitly by B-splines and refined regions located around the design boundaries.

So far, most topology optimization approaches have been implemented within the framework of classical finite element, i.e., relying on low-order Lagrange interpolation functions. Nonetheless, using B-splines or NURBS as basis functions has become an increasingly popular approach, along with the development of isogeometric analysis (IGA) (see Hughes et al. (2005) and Cottrell et al. (2009)). B-Splines are used to both describe the geometry of a structure and solve for its mechanical responses in IGA. From a geometry point of view, using B-splines and NURBS facilitates compatibility with computer-aided design (CAD) software that are based on this type of basis functions. From an analysis point of view, using smooth and higher order bases, such as quadratic and cubic B-splines, leads to more accurate structural responses per DOF than classical  $C^0$  finite element approaches (see Hughes et al. (2008), Evans et al. (2009), and Hughes et al. (2014)). For topology optimization, B-splines can offer additional advantages. They tend to promote smoother designs, prevent the development of spurious features, and limit the need for filtering techniques. Several works successfully apply topology optimization in combination with IGA; see for example Dedè et al. (2012) for a phase-field approach, Qian (2013) for a density approach, Wang and Benson (2016) or Jahangiry and Tavakkoli (2017) for a level set approach, Wang et al. (2017) for lattice structure designs, or Lieu and Lee (2017) for multi-material designs. A comprehensive review is presented in Wang et al. (2018).

Classical B-splines approaches do not allow for local refinement, as their tensor-product structure inherently enforces global refinement. However, in solving topology optimization problems, local mesh adaptivity is key to achieve both a precise description of the design boundaries and accurate mechanical response computations while maintaining a reasonable computational cost. Hierarchical B-splines (HB-splines), as introduced by Forsey and Bartels (1988), naturally support local refinement, but do not form a partition of unity (PU), a beneficial property for numerical purposes. Therefore, the concept was extended to so-called truncated hierarchical B-splines (THB-splines) to recover the PU and other advantageous properties (see Giannelli et al. (2012)). Local adaptive mesh refinement relying on HB-splines or THB-splines was implemented by Schillinger et al. (2012) and Garau and Vázquez (2018).

Recently, research efforts have been dedicated to the exploitation of B-spline refinement in conjunction with level set topology optimization. Solving compliance minimization problems, Bandara et al. (2016) used B-spline

shape functions for both the geometry representation and the analysis with immersed boundary techniques. They proposed a global mesh refinement and coarsening strategy based on Catmull-Clark subdivision surfaces. Wang et al. (2019) focused on cellular structures and used B-spline shape functions to represent the geometry of each representative cell. In their approach, local refinement, i.e., cell subdivision, is achieved by knot insertion. Recently, Xie et al. (2020) used THB-splines as interpolation functions and the Solid Isotropic Material with Penalization (SIMP) method to perform density-based topology optimization with local refinement.

This paper proposes an adaptive mesh refinement strategy using HB-splines to perform level set topology optimization. We use a level set function (LSF) to describe the design geometry implicitly. The structural analysis relies on an immersed boundary technique to predict the system responses, here the eXtended Finite Element Method (XFEM) with a generalized Heaviside enrichment strategy. We discretize both the design and the state variable fields, i.e., the level set and the displacement fields, with HB-splines. Contrary to most approaches relying on lower order basis functions, we interpolate the design variables with higher order B-splines. For simplicity, the interpolation of the displacement field is restricted to first-order functions. For local refinement, we consider truncated B-splines. Although not widely used in the literature so far, THB-splines satisfy the PU and form a convex hull, allowing us to conveniently impose bounds on the design variables. Refinement is triggered according to a user-defined criterion, here the location of the design boundaries, and the mesh is refined along the interfaces and coarsened in the solid and void phases. As the design and state variable fields are defined on the same mesh, they are refined or coarsened simultaneously, which allows for a sufficiently accurate resolution of both the geometry and the physical response.

Two approaches to handle the design space in level set topology optimization are considered: (i) seeding an initial hole pattern and (ii) nucleating holes during the optimization process using a combined level set/density approach. We solve the optimization problems with mathematical programming techniques and in particular the Globally Convergent Method of Moving Asymptotes (GCMMA) (see Svanberg (2002)). The required sensitivity analysis is carried out with an adjoint formulation.

The ability of the proposed approach to generate complex geometries at a reduced computational cost starting from initial coarse meshes is assessed with two- and three-dimensional topology optimization problems. By varying the adaptive strategy and the underlying B-spline

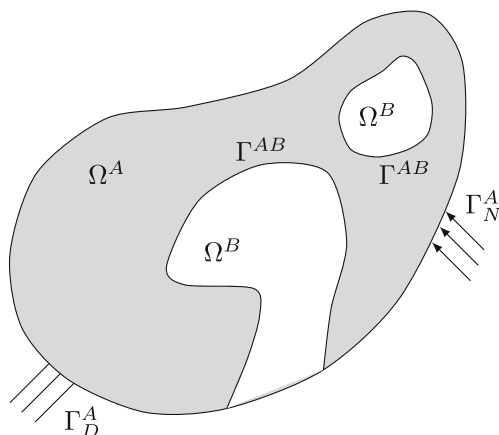
discretization, we can characterize the influence of the mesh adaptivity on the optimization results and the computational cost. Our numerical study results in interesting findings. First, the mesh adaptivity strategy influences the optimization process. Finer initial meshes and more frequent refinement operations lead to the development of more complex geometries exhibiting thin structural members. The properties of B-splines have an impact on the generated designs and an educated choice of the B-splines features can lead to advantageous behaviors for topology optimization. Higher order B-splines promote smoother geometries and tend to eliminate small spurious features from the design. Truncation allows us to conveniently handle enforcement of bounds on the design variables. The numerical examples also reveal that hole nucleation is crucial for the computational efficiency of the adaptive strategy, as it allows starting the optimization process on coarser meshes.

The remainder of the paper is organized as follows. Section 2 presents the level set description of the geometry using B-splines. Section 3 discusses the strategies adopted to achieve a sufficiently rich design space working with level set topology optimization. Section 4 details the mesh refinement and coarsening strategies, while Section 5 focuses on the construction of hierarchical B-spline bases. The structural analysis is described in Section 6; governing equations are detailed and the basic principles of XFEM are recalled. Section 7 is devoted to the optimization problem formulation, the used regularization schemes, and the sensitivity analysis implementation. Section 8 studies the proposed approach with 2D and 3D solid/void design examples. Finally, Section 9 summarizes the work and draws some conclusions.

## 2 Geometry description

Since the development of the level set method (LSM) by Osher and Sethian (1988), it has been extensively used in combination with shape and topology optimization to implicitly describe the geometry of the design (see van Dijk et al. (2013) and Sigmund and Maute (2013)).

A general description of the two-phase problems addressed in this paper is given in Fig. 1. Two material phases,  $A$  and  $B$ , are distributed over a  $d$ -dimensional design domain  $\Omega \subset \mathbb{R}^d$ . Material subdomains are non-overlapping, i.e.,  $\Omega = \Omega^A \cup \Omega^B$ . The boundaries of material subdomains  $A$  and  $B$  are denoted  $\partial\Omega^A$  and  $\partial\Omega^B$ . Dirichlet and Neumann boundary conditions are applied to  $\Gamma_D^I = \partial\Omega^I \cap \partial\Omega_D$  and  $\Gamma_N^I = \partial\Omega^I \cap \partial\Omega_N$  respectively, with  $I = A, B$ . The material phases are separated by an interface  $\Gamma^{AB} = \partial\Omega^A \cap \partial\Omega^B$ .



**Fig. 1** Design domain description

A LSF is used to describe the material distribution within the design domain  $\Omega$ . At a given point  $\mathbf{x}$  in space, the design geometry is defined as:

$$\begin{aligned} \phi(\mathbf{x}) &< 0, \forall \mathbf{x} \in \Omega^A, \\ \phi(\mathbf{x}) &> 0, \forall \mathbf{x} \in \Omega^B, \\ \phi(\mathbf{x}) &= 0, \forall \mathbf{x} \in \Gamma^{AB}. \end{aligned} \quad (1)$$

The LSF is interpolated using B-spline shape functions  $B_i(\mathbf{x})$  as:

$$\phi^h(\mathbf{x}) = \sum_i B_i(\mathbf{x}) \phi_i, \quad (2)$$

where  $\phi_i$  are the B-spline coefficients. In this work, the B-spline coefficients and corresponding shape functions are used to evaluate nodal level set values on the integration mesh, that is the mesh over which the weak form of the governing equations is integrated. These nodal values are used to interpolate the level set field linearly within an integration element. The linear approximation within an element simplifies the construction of the intersection of the LSF with the element edges and does not affect the accuracy of the analysis, as linear interpolation functions are used for the state variable field. Further explanations about the level set field interpolation on the analysis mesh are provided in Section 5.

Contrary to classical approaches by Wang et al. (2003) or Allaire et al. (2004), where the level set is updated in the optimization process by solving Hamilton-Jacobi-type equations, the coefficients of the LSF are here defined as explicit functions of the design variables. They are updated by mathematical programming techniques driven by shape sensitivity computations.

It should be noted that, in this work, no filter is used to widen the zone of influence of the design variables and thus enhance the convergence of the optimization problem, as proposed in Kreissl and Maute (2012). Proceeding this way, the B-spline basis function support is not altered and its influence on the optimization process and the obtained designs can be emphasized and compared for different B-spline orders.

### 3 Seeding of inclusions

The design updates in level set topology optimization is solely driven by shape sensitivities (see van Dijk et al. (2013)). To allow for a sufficient freedom in the design, level set optimization techniques require either seeding inclusions in the initial design or introducing inclusions during the optimization process. In the context of solid/void problem, the inclusion represents a hole.

Seeding inclusions in the initial design domain is a commonly used strategy and has been successfully implemented in level set topology optimization (see for example Villanueva and Maute (2014)). Nonetheless, this strategy leads to several difficulties. To avoid a premature removal of holes and geometric features which may lead to suboptimal designs, the optimization process is started from a feasible configuration. In this case, finding a hole pattern that satisfies the design constraints is not a trivial task. Moreover, this strategy requires a sufficiently fine mesh able to resolve both the physics and the design of the initial layout, as the initial hole pattern can easily constitute the most geometrically complex configuration over the entire optimization process. Thus, as will be shown in Section 8, seeding initial holes limits the computational effectiveness of the proposed adaptive discretization strategy. Therefore, allowing for hole seeding during the optimization process proves to be advantageous in the context of mesh adaptivity.

Topological derivatives constitute a systematic approach to seed new holes in the design domain during the optimization process (see Novotny and Sokołowski (2013) for a detailed introduction). The basic concept of evaluating the influence, i.e., the sensitivity, of introducing an infinitesimal hole on the objective and constraint functions was introduced by Eschenauer et al. (1994). A finite-sized hole is inserted at a location where the topological derivative field associated to an objective or considered cost function is minimal. The shape of the new hole is then optimized along with existing domain boundaries. The reader is referred to Sigmund and Maute (2013) or Maute (2017) for further details. Although proven useful, topological derivatives only provide information about where to place an infinitesimal

hole and it remains unclear what the size and shape of a finite-sized hole should be.

Alternatively, hole seeding during the optimization process can be achieved by combining level set and density-based techniques, as explored recently in Kang and Wang (2013), Geiss et al. (2019b), and Burman et al. (2019) or Jansen (2019). Following the single-field approach of Barrera et al. (2019), an abstract design variable field,  $s(\mathbf{x})$ , is introduced with  $s \in \mathbb{R}, 0 \leq s(\mathbf{x}) \leq 1$ , to define both the level set field,  $\phi$ , and spatially variable material properties, such as the density  $\rho$  and the Young’s modulus  $E$ , using a SIMP interpolation scheme:

$$\phi(\mathbf{x}) = \phi_{sc} (\phi_{th} - s(\mathbf{x})), \tag{3}$$

$$\rho(\mathbf{x}) = \begin{cases} 0, & \forall s(\mathbf{x}) < \phi_{th}, \\ \rho_{sh} + (\rho_0 - \rho_{sh}) \frac{(s(\mathbf{x}) - \phi_{th})}{(1 - \phi_{th})}, & \forall s(\mathbf{x}) \geq \phi_{th}, \end{cases} \tag{4}$$

$$E(\mathbf{x}) = E_0 \rho^\beta, \tag{5}$$

where  $\phi_{sc}$  is a scaling parameter that accounts for the mesh size  $h$  and is set to  $\phi_{sc} \simeq 3h \dots 5h$ ,  $\phi_{th}$  is a threshold that defines the void/solid phases in terms of  $s(\mathbf{x})$  and  $\rho_{sh}$ , a parameter that controls the minimum density in the solid domain. The properties of the bulk material are denoted by  $\rho_0$  and  $E_0$ , and  $\beta$  is the SIMP exponent. The resulting interpolation scheme is illustrated in Fig. 2. Throughout the optimization process, the parameter  $\phi_{th}$  is kept constant and equal to  $\phi_{th} = 0.5$ , while  $\rho_{sh}$  is a continuation parameter and is gradually increased to  $\rho_{sh} = 1$  during the process. To avoid ill-conditioning, its initial value is set to  $\rho_{sh} \simeq 0.1 \dots 0.2$ , unless a smaller value is required to satisfy an initial mass or volume constraint. Contrary to classical density-based approaches, the density here is used primarily for hole seeding and for convergence acceleration. A 0 – 1 material distribution is achieved through continuation on the

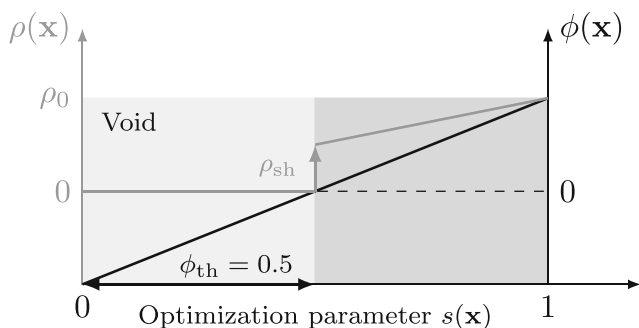


Fig. 2 Interpolation of the level set  $\phi(\mathbf{x})$  and the density  $\rho(\mathbf{x})$  for the combined level set/density scheme

parameter  $\rho_{sh}$ . A low SIMP exponent, e.g.,  $\beta = 2.0$ , is used to reduce the bias of the density method to rapidly separate the material distribution into a solid and void phase. For the proposed mesh adaptation strategy, we have observed that using a low SIMP exponent promotes the formation of fine features, especially when starting from coarse meshes.

In this paper, both approaches, i.e., the seeding of holes in the initial design and the combined level set/density scheme, are considered and compared.

### 4 Mesh adaptivity

This section focuses on the strategies used to perform hierarchical mesh refinement. First, we introduce the notion of a hierarchical mesh. We recall the basic concepts of mesh refinement and detail our implementation of the refinement strategy. Finally, the considered user-defined criteria, used to trigger mesh adaptation, are explained in detail.

#### 4.1 Hierarchical mesh

In this work, a regular background tensor grid is used to build hierarchical meshes. The elements of the background grid do not carry any notion of interpolation, i.e., an element of the background grid is not associated with B-spline bases or Lagrange nodes. Starting from a uniform background grid with a refinement level  $l_0$  and subdividing its elements, elements with a higher refinement level  $l > l_0$  are created.

Formally to define a hierarchical mesh of depth  $n$ , a sequence of subdomains  $\Omega^l$  is introduced:

$$\Omega^{n-1} \subseteq \Omega^{n-2} \subseteq \dots \subseteq \Omega^0 = \Omega, \tag{6}$$

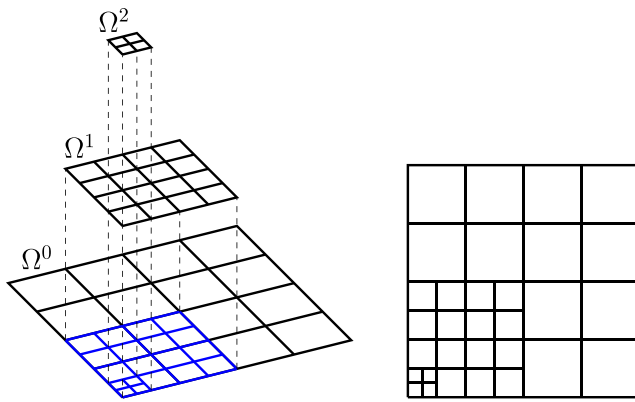
where each subdomain  $\Omega^l$  is the subregion of  $\Omega$  selected to be refined at the level  $l$  and is the union of elements on the level  $l - 1$ . The creation of such a refinement pattern is illustrated in Fig. 3.

In Section 5, these subdomains  $\Omega^l$ , refined to a level  $l$ , are used to create hierarchical B-spline bases to interpolate the design and state variable fields.

#### 4.2 Local refinement strategies

This subsection focuses on the refinement strategies used in this paper and explains their implementation. Before refinement is carried out by element subdivision, elements need to be flagged for adaptation. Elements can be flagged based on one or several user-defined criteria and on additional mesh regularity requirements.





**Fig. 3** Hierarchical refined mesh

A complete refinement step proceeds as stated in Algorithm 1. Initially, elements are flagged for refinement based on user-defined criteria, as further explained in Section 4.2.1. Additionally, elements are flagged within a so-called buffer region, around the previously flagged elements, so that the obtained refined mesh complies with specific regularity requirements, as further detailed in Section 4.2.2. Finally, a minimum refinement level  $l_{\min}$  is enforced and elements with  $l < l_{\min}$  are flagged and refined. To increase performance, flagged elements are collected in a refinement queue first and refined afterwards.

---

**Algorithm 1** Refinement algorithm.

---

- 1: Apply refinement strategy in Algorithm 2
  - 2: Collect flagged elements into queue for refinement
  - 3: **while** change in refinement queue **do**
  - 4:     **for** all elements in queue for refinement **do**
  - 5:         Apply refinement buffer in Algorithm 3
  - 6:         Collect flagged elements in queue for refinement
  - 7:     **end for**
  - 8: **end while**
  - 9: Refine elements in refinement queue
  - 10: **for** all elements **do**
  - 11:     **if** element refinement level  $l < l_{\min}$  **then**
  - 12:         Flag element for refinement
  - 13:         Refine element in refinement queue
  - 14:     **end if**
  - 15: **end for**
- 

In this paper, design and state variables are defined on a common mesh, although the interpolation order may differ. Thus, the discretization of both fields are adapted simultaneously. While the proposed framework allows for

separately refining state and design variable fields, this option is not considered here for the sake of simplicity.

After each adaptation of the mesh, the design variable field  $\phi$ , as described in (2), is mapped to the new refined mesh through an  $L^2$  projection. The B-spline coefficients  $\phi_{i,\text{new}}$  on the new mesh are obtained from the ones on the old mesh  $\phi_{i,\text{old}}$  by solving:

$$\int_{\Omega} \delta \phi_{\text{new}} (\phi_{\text{new}} - \phi_{\text{old}}) d\Omega = 0, \quad (7)$$

where

$$\begin{aligned} \phi_{\text{new}} &= \sum_i B_{i,\text{new}}(\mathbf{x}) \phi_{i,\text{new}}, \\ \phi_{\text{old}} &= \sum_i B_{i,\text{old}}(\mathbf{x}) \phi_{i,\text{old}}. \end{aligned} \quad (8)$$

This projection step requires an additional linear solve for a system whose size depends on the number of design variables of the new mesh.

#### 4.2.1 User-defined refinement criteria

Refinement can be triggered by one or several user-defined criteria, including geometry criteria or finite element error indicators, although the latter are not considered here. In this paper, refinement is applied after a given number of optimization steps, defined by the user, and using a purely geometric criterion, i.e., the distance to the solid-void interfaces.

During each refinement step, an element is flagged for refinement or for keeping its current refinement level by the Algorithm 2. Each element mesh is flagged for refinement depending on its nodal level set values  $\phi$ . The parameter  $\phi_{\text{bw}}$  defines the bandwidth around the interface which is refined. This allows fine-tuning the zone of refinement in addition to the buffer introduced in Section 4.2.2. Over the course of the optimization process, at the solid-void boundary, the mesh is refined up to a maximum refinement level defined for the interface  $l_{\text{ifc,max}}$ . Within the solid phase, the mesh is refined up to a maximum refinement level  $l_{\text{solid,max}}$ , that is usually equal or smaller than the interface one,  $l_{\text{solid,max}} \leq l_{\text{ifc,max}}$ . The void phase is never flagged for refinement. A minimum refinement level  $l_{\min} \geq 0$  is applied. As each refinement operation is started from the coarsest refinement level possible, an element that is not flagged will not be refined at all. At the beginning of the optimization,  $l_{\min}$  is generally set to  $l_{\min} > 0$ . By lowering  $l_{\min}$  throughout the optimization process, a coarsening effect is achieved.

**Algorithm 2** Refinement strategy.

---

```

1: Step 0. Set  $l_{\text{ifc}}$ ,  $l_{\text{solid}}$ , and  $l_{\text{void}}$  to an initial mesh refinement level  $l_0$ , where  $l_{\text{min}} \leq l_0 \leq \min(l_{\text{solid,max}}, l_{\text{ifc,max}})$ .
2: Step 1. Flag elements
3: for all elements do
4:   if  $-\phi_{\text{bw}} \leq \phi \leq +\phi_{\text{bw}}$  then
5:     if  $l_{\text{current}} < l_{\text{ifc}}$  then
6:       Refine
7:     else
8:       Keep
9:     end if
10:  end if
11:  if  $\phi < -\phi_{\text{bw}}$  then
12:    if  $l_{\text{current}} < l_{\text{solid}}$  then
13:      Refine
14:    else
15:      Keep
16:    end if
17:  end if
18:  if  $\phi > +\phi_{\text{bw}}$  then
19:    if  $l_{\text{current}} > l_{\text{void}}$  then
20:      Do nothing
21:    else
22:      Keep
23:    end if
24:  end if
25: end for
26: Step 2. Increase/reduce refinement levels
27:  $l_{\text{ifc}} = \min(l_{\text{ifc}} + 1, l_{\text{ifc,max}})$ 
28:  $l_{\text{solid}} = \min(l_{\text{solid}} + 1, l_{\text{solid,max}})$ 
29:  $l_{\text{void}} = \max(l_{\text{void}} - 1, l_{\text{min}})$ 

```

---

In the course of the optimization process, the shape of the solid-void interface changes and the interface may intersect elements previously not refined. In this study, the mesh is adapted to maintain a uniform refinement level for all intersected elements. Although not required by the proposed framework, this strategy is used here as it simplifies the XFEM enrichment (see Section 6.2). The same enrichment algorithm can be used for all intersected elements, as is the case for uniformly refined meshes. To this end, Algorithm 1 is executed, omitting Step 2 of Algorithm 2.

#### 4.2.2 Mesh regularity requirements

In this paper, we limit the size difference between adjacent elements in the refined mesh, i.e., a refinement of more than a factor 4 in 2D and 8 in 3D is not allowed. Although not mandatory (see studies by Jensen (2016) and Panesar et al. (2017)), this rule is adopted in most works dedicated to adaptivity as it promotes accurate analysis results.

This requirement can be achieved by enforcing a so-called buffer zone around elements primarily flagged for refinement. The width of the buffer zone  $d_{\text{buffer}}$  for a particular flagged element is calculated by multiplying its size with a user-defined buffer parameter  $b_{\text{buffer}}$ . The width of the buffer zone must be larger than the support size of the considered interpolation functions. Using B-splines, it means that  $b_{\text{buffer}} \geq p$ , where  $p$  is the polynomial degree of the used B-spline basis.

The mesh refinement procedure for building a buffer zone is summarized in Algorithm 3. Within a refined mesh, coarse elements are called parents; refined elements, children. The algorithm is applied to each flagged element and starts by determining its parent element. The refinement status of the parent's neighbors, i.e., elements lying within the buffer range of the considered parent, is checked. If these neighbors are neither refined nor flagged for refinement, the distance  $d_{\text{max}}$  between the considered parent element and its neighbors is calculated. If the distance  $d_{\text{max}}$  is smaller than the buffer size  $d_{\text{buffer}}$ , the neighbor elements are flagged for refinement. The algorithm is then applied recursively to all newly flagged neighbor elements, until no extra elements are flagged. An easy and efficient access to hierarchical mesh information, such as neighborhood relationships, is provided by using a quadtree and an octree data structure in 2D and 3D respectively.

**Algorithm 3** Refinement buffer algorithm.

---

```

1: Get parent
2: Get parent's neighbors in buffer range  $d_{\text{buffer}}$ 
3: if neighbor is not refined and not flagged for refinement then
4:   Calculate distance  $d_{\text{max}}$  between the flagged element and the neighbor
5:   if  $d_{\text{max}} < d_{\text{buffer}}$  then
6:     Flag neighbor for refinement
7:     Apply refinement buffer Algorithm 3 for neighbor
8:   end if
9: end if

```

---

## 5 Hierarchical B-splines

This section focuses on hierarchical B-splines for adaptively discretizing design variable and state variable fields. First the basic concepts of B-splines in one and multiple dimensions are recalled. Then, the principle of B-spline refinement and the construction of truncated and non-truncated hierarchical B-spline bases are briefly described.

### 5.1 Univariate B-spline basis functions

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

$$N_{i,0}(\xi) = \begin{cases} 1, & \text{if } \xi_i \leq \xi \leq \xi_{i+1}, \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

and using the Cox de Boor recursion formula for higher degrees  $p > 0$  (see de Boor (1972)):

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{10}$$

A knot is said to have a multiplicity  $k$  if it is repeated in the knot vector. The corresponding B-spline basis exhibits a  $C^{p-k}$  continuity at that specific knot, while it is  $C^\infty$  in between the knots. A knot span is defined as the half-open interval  $[\xi_i, \xi_{i+1})$ . Within this context, an element is defined as a nonempty knot span.

### 5.2 Tensor-product B-spline basis functions

Tensor-product B-spline basis functions  $B_{i,p}(\xi)$  are obtained by applying the tensor product to univariate B-spline basis functions. Denoting the parametric space dimension as  $d_p$ , a tensor-product B-spline basis is constructed starting from  $d_p$  knot vectors  $\Xi^m = \{\xi_1^m, \xi_2^m, \dots, \xi_{n_m+p_m+1}^m\}$  with  $p_m$  denoting the polynomial degree and  $n_m$  the number of basis functions in the parametric direction  $m = 1, \dots, d_p$ . A tensor-product B-spline basis function is generated from  $d_p$  univariate B-splines  $N_{i_m,p_m}^m(\xi^m)$  in each parametric direction  $m$  as:

$$B_{i,p}(\xi) = \prod_{m=1}^{d_p} N_{i_m,p_m}^m(\xi^m), \tag{11}$$

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

### 5.3 B-Spline refinement

Hierarchical refinement of uniform B-splines can be achieved by subdivision. A univariate B-spline basis function

can be expressed as a linear combination of  $p+2$  contracted, translated, and scaled copies of itself:

$$N_p(\xi) = 2^{-p} \sum_{j=0}^{p+1} \binom{p+1}{j} N_p(2\xi - j), \tag{12}$$

where the binomial coefficient is defined as:

$$\binom{p+1}{j} = \frac{(p+1)!}{j!(p+1-j)!}. \tag{13}$$

Figure 4 shows the refinement of a quadratic univariate B-spline basis function obtained by subdivision.

The extension of the subdivision property in (11) to tensor-product B-spline basis functions  $B_p$  is straightforward as they exhibit a tensor-product structure:

$$B_p(\xi) = \sum_{\mathbf{j}} \left( \prod_{m=1}^d 2^{-p_m} \binom{p_m+1}{j_m} N_{p_m}(2\xi_m - j_m) \right), \tag{14}$$

where the indices  $\mathbf{j} = \{j_1, \dots, j_{d_p}\}$  indicate the position in the tensor-product structure.

### 5.4 Hierarchical B-splines

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

$$\mathcal{V}^0 \subset \mathcal{V}^1 \subset \mathcal{V}^2 \subset \mathcal{V}^3 \subset \dots \tag{15}$$

Each B-spline space  $\mathcal{V}^l$  has a corresponding basis  $\mathcal{B}^l$ .

A hierarchical B-spline basis  $\mathcal{H}$  can be constructed recursively based on the sequence of B-spline bases  $\mathcal{B}^l$  that span the domains  $\Omega^l$ . 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

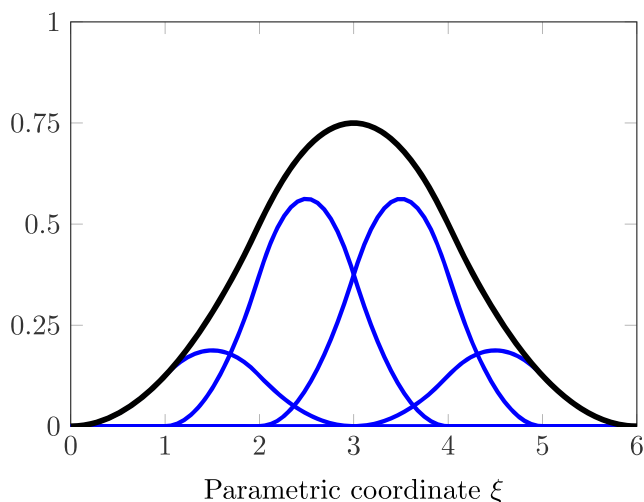


Fig. 4 Subdivision of a quadratic B-spline basis function (black) into  $p+2$  contracted B-spline basis functions of half the knot span width (blue)



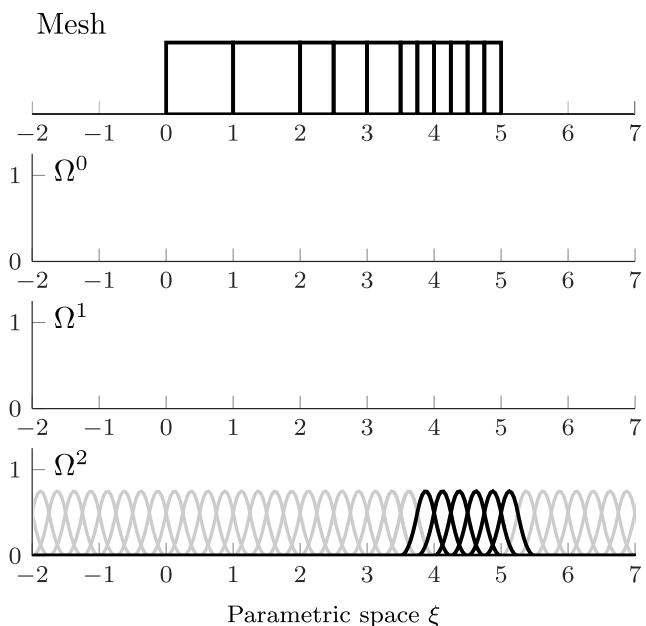
constructed by taking the union of all basis functions  $B$  in  $\mathcal{H}^l$  whose support is not fully enclosed in  $\Omega^{l+1}$  and all basis functions  $B$  in  $\mathcal{B}^{l+1}$  whose support lies in  $\Omega^{l+1}$ . Following Garau and Vázquez (2018), the recursive algorithm reads:

$$\begin{cases} \mathcal{H}^0 & =: \mathcal{B}^0 \\ \mathcal{H}^{l+1} & =: \{B \in \mathcal{H}^l \mid \text{supp}(B) \not\subseteq \Omega^{l+1}\} \cup \\ & \quad \{B \in \mathcal{B}^{l+1} \mid \text{supp}(B) \subseteq \Omega^{l+1}\}, \\ & \text{for } l = 0, \dots, n - 2, \end{cases} \quad (16)$$

where the index  $l$  gives 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.

A hierarchical B-spline basis  $\mathcal{H}$  is illustrated for a one-dimensional example in Fig. 5. The top row shows a one-dimensional hierarchical refined mesh. Following (16) a B-spline basis  $\mathcal{H}$  is created through an initialization step with all bases in the subdomain  $\Omega^0$  refined to a level  $l = 0$ . Recursively, all bases in the subdomain  $\Omega^{l+1}$  with higher refinement level  $l + 1$  are added, while existing basis functions of level  $l$  fully enclosed in  $\Omega^{l+1}$  are discarded.

Contrary to  $h$ -refinement in classical finite element where extra treatments, such as the introduction of multi-point constraints, are necessary, hanging nodes in hierarchical refined meshes are naturally handled by the B-spline basis.



**Fig. 5** Three levels of refinement on a given mesh. The refined mesh is depicted in the top figure. The three underlying figures present the three levels of refinement considered here. Within these three levels, the active B-spline basis  $\mathcal{H}$  is displayed in black, while the inactive B-spline basis functions are displayed in gray

### 5.5 Truncated B-splines

A major drawback of hierarchical B-spline bases is the loss of the PU property. The truncated hierarchical B-spline basis constitutes an extension of the hierarchical B-spline basis, aiming at recovering the PU principle and at reducing the number of overlapping functions on adjacent hierarchical levels (see Giannelli et al. (2012)). Considering a basis function  $B^l$ , part of  $\mathcal{B}^l$  and defined on the domain  $\Omega^l$ , its representation in terms of the finer basis of level  $l + 1$  is given as:

$$B^l = \sum_{B^{l+1} \in \mathcal{B}^{l+1}} c_{B^{l+1}}^{l+1} (B^l) B^{l+1}, \quad (17)$$

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

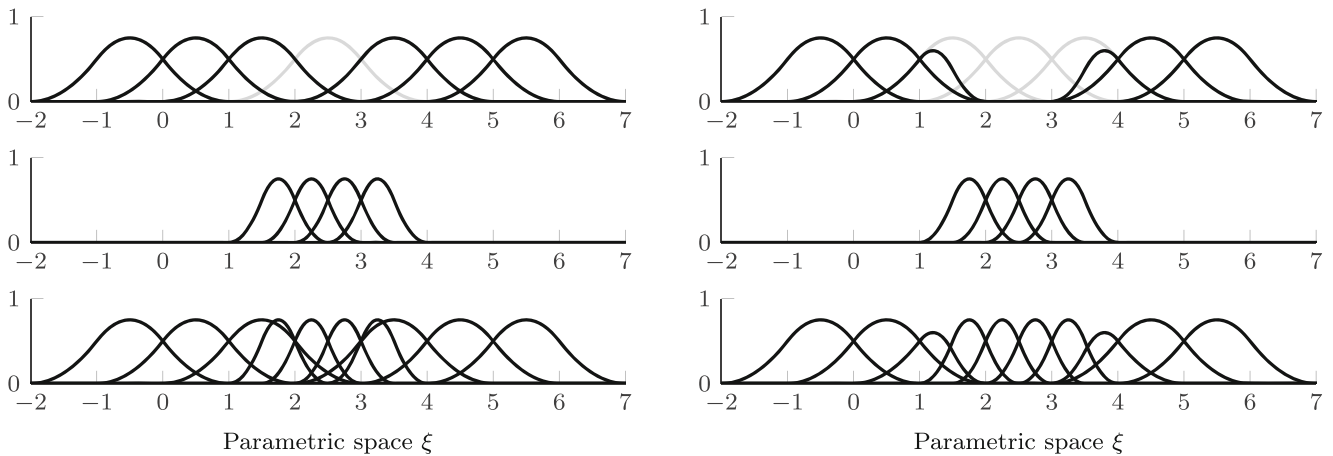
As described in Giannelli et al. (2012) and Garau and Vázquez (2018), the truncation of this basis function  $B^l$ , whose support overlaps with the support of finer basis functions  $B^{l+1}$ , part of  $\mathcal{B}^{l+1}$  and defined on  $\Omega^{l+1}$ , is expressed as:

$$\begin{aligned} \text{trunc}^{l+1}(B^l) &= \sum_{\substack{B^{l+1} \in \mathcal{B}^{l+1}, \\ \text{supp}(B^{l+1}) \not\subseteq \Omega^{l+1}}} c_{B^{l+1}}^{l+1} (B^l) B^{l+1}, \\ &= B^l - \sum_{\text{supp}(B^{l+1}) \subseteq \Omega^{l+1}} c_{B^{l+1}}^{l+1} (B^l) B^{l+1}. \end{aligned} \quad (18)$$

Similar to the creation of a hierarchical B-spline basis  $\mathcal{H}$ , a truncated hierarchical B-spline basis  $\mathcal{T}$  is constructed recursively, but by additionally applying the truncation in (18) at each iteration:

$$\begin{cases} \mathcal{T}^0 & =: \mathcal{B}^0 \\ \mathcal{T}^{l+1} & =: \{\text{trunc}^{l+1}(B) \mid B \text{ in } \mathcal{T}^l \wedge \text{supp}(B) \not\subseteq \Omega^{l+1}\} \\ & \quad \cup \{B \in \mathcal{B}^{l+1} \mid \text{supp}(B) \subseteq \Omega^{l+1}\}, \\ & \text{for } l = 0, \dots, n - 2. \end{cases} \quad (19)$$

The truncated basis  $\mathcal{T}$  spans the same space as the non-truncated basis  $\mathcal{H}$ , but its underlying functions exhibit the following properties. They are defined on smaller supports which result in sparser stiffness matrices in a finite element analysis. The PU property is restored which allows for the imposition of bounds on variables. In addition, truncated bases admit a strong stability property (see Giannelli et al. (2014)) which enables the construction of optimal multi-level solvers (see Hofreither et al. (2016)). However, this property is not exploited in this paper, as the displacements are interpolated by linear B-splines.

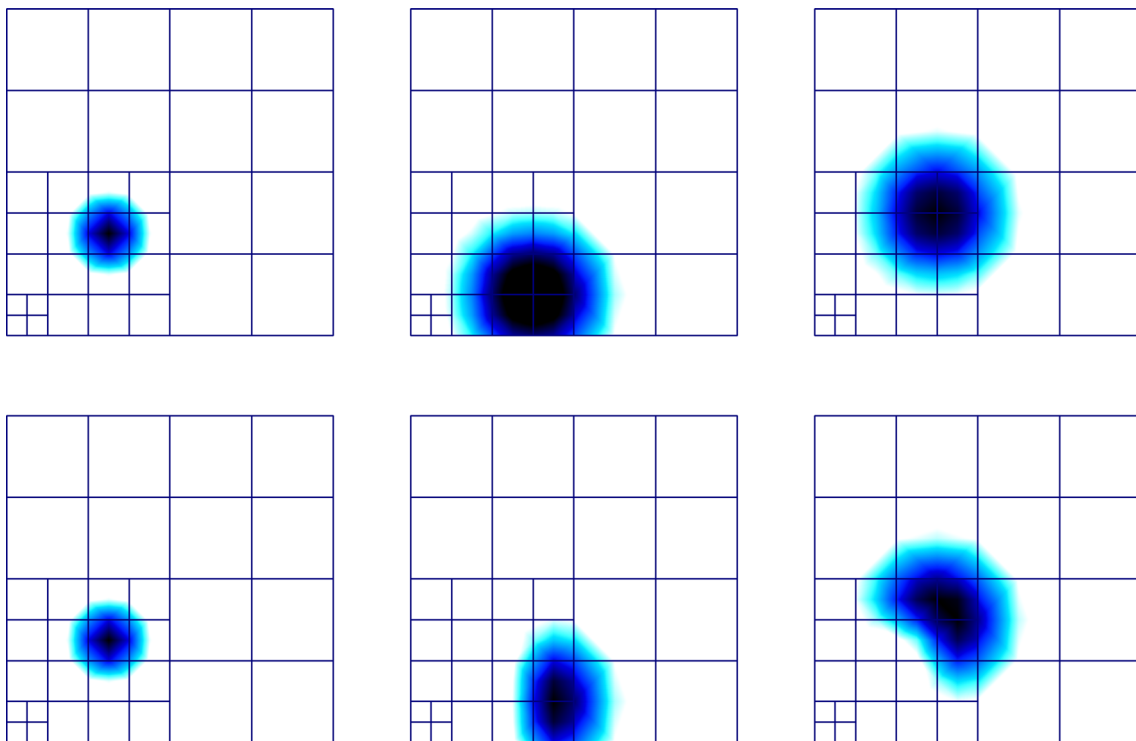


**Fig. 6** Comparison of univariate HB-spline (left) and THB-spline basis functions (right). The first and second levels correspond to  $\Omega^0$  and  $\Omega^1$  respectively, while the bottom level represents the combination of the functions on these two levels

The effect of the truncation is illustrated in Fig. 6. Univariate truncated and non-truncated bases  $\mathcal{T}$  and  $\mathcal{H}$  are juxtaposed for comparison. The comparison shows the reduced support of the truncated B-splines. The effect of truncation on the support of a multivariate, two-dimensional B-spline basis is shown in Fig. 7. Truncated multivariate B-splines present a characteristic kidney-shaped support zone.

## 6 Structural analysis

Whereas the proposed adaptive discretization scheme is not limited to any particular physics problem, in this paper, we consider solid-void problems as illustrated in Fig. 1, where the material subdomain  $A$  is filled with a linear elastic solid, while the material subdomain  $B$  is void. In this section, we present the variational form of the governing equations for



**Fig. 7** Comparison of multivariate HB-spline (top) and THB-spline (bottom) supports. While the non-truncated HB-spline basis functions show a uniform support zone, the support zone of THB-spline basis functions exhibits a characteristic kidney shape

the linear elastic analysis model. As the XFEM is used to discretize the state variable fields in the solid domain, the basic principles of the method are briefly recalled.

## 6.1 Governing equations

We solve for static equilibrium to enforce balance of linear momentum within the solid domain  $\Omega^A$ . In this work, the total residual  $\mathcal{R}$ , i.e., the weak form of the governing equations, consists of four terms which are discussed subsequently:

$$\mathcal{R} = \mathcal{R}_{\text{Lin}} + \mathcal{R}_{\text{Nitsche}} + \mathcal{R}_{\text{Ghost}} + \mathcal{R}_{\text{Spring}}. \quad (20)$$

The weak form of the linear elastic governing equation reads:

$$\mathcal{R}_{\text{Lin}} = \int_{\Omega^A} \delta \boldsymbol{\varepsilon} : \boldsymbol{\sigma} \, d\Omega - \int_{\Gamma_N^A} \delta \mathbf{u} \cdot \mathbf{t}_N \, d\Gamma, \quad (21)$$

where  $\mathbf{u}$  and  $\delta \mathbf{u}$  are the displacement field and the test function, respectively. Traction forces,  $\mathbf{t}_N$ , are applied on the Neumann boundary,  $\Gamma_N^A$ . The Cauchy stress tensor is denoted by  $\boldsymbol{\sigma} = \mathbf{D} \boldsymbol{\varepsilon}$  and is obtained by multiplication of the infinitesimal strain tensor  $\boldsymbol{\varepsilon} = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T)$  with the fourth-order constitutive tensor  $\mathbf{D}$ , here for isotropic linear elasticity.

To weakly enforce prescribed displacements on Dirichlet boundaries, the static equilibrium in (21) is augmented with Nitsche's method (see Nitsche (1971)):

$$\begin{aligned} \mathcal{R}_{\text{Nitsche}} = & - \int_{\Gamma^{AB}} \delta \mathbf{u} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}_\Gamma) \, d\Gamma \\ & + \int_{\Gamma^{AB}} \delta (\boldsymbol{\sigma} \cdot \mathbf{n}_\Gamma) \cdot (\mathbf{u} - \mathbf{u}_D) \, d\Gamma \\ & + \gamma_N \int_{\Gamma^{AB}} \delta \mathbf{u} \cdot (\mathbf{u} - \mathbf{u}_D) \, d\Gamma, \end{aligned} \quad (22)$$

where  $\mathbf{u}_D$  are the displacement imposed on the Dirichlet boundary,  $\Gamma_D^A$ . The parameter  $\gamma_N$  is chosen to achieve a certain accuracy in satisfying the boundary conditions and is a multiple of the ratio  $E/h$ , where  $E$  is the Young's modulus of the considered material and  $h$  is the edge length of the intersected elements.

Using immersed boundary techniques (see Section 6.2), numerical instabilities arise when either the contributions of the DOFs interpolating the displacement field on the residual vanish and/or these contributions become linearly dependent. These issues typically arise when the level set field intersects elements such that small material subdomains emerge. This results in an ill-conditioning of the equation system and inaccurate prediction of displacement gradients along the interface. The face-oriented ghost stabilization proposed by Burman and Hansbo (2014) is used in this work to mitigate this issue. Using a virtual work-based formulation, the jump

in the displacements gradients is penalized across the faces belonging to intersected elements by augmenting the residual equations with:

$$\mathcal{R}_{\text{Ghost}} = h \gamma_G \sum_{F \in \mathcal{F}_{\text{cut}}} \int_F \llbracket \delta \boldsymbol{\varepsilon} \cdot \mathbf{n}_F \rrbracket \llbracket \boldsymbol{\sigma} \cdot \mathbf{n}_F \rrbracket \, d\Gamma, \quad (23)$$

where  $F$  is a specific face belonging to  $\mathcal{F}_{\text{cut}}$ , the set of element faces cut by the interface, and  $\mathbf{n}_F$  is the outward normal to face  $F$ . The jump operator  $\llbracket \bullet \rrbracket$  computes quantities between adjacent elements  $A$  and  $B$  as  $\llbracket \bullet \rrbracket = \bullet^A - \bullet^B$ . The ghost penalty parameter is denoted  $\gamma_G$  and typically takes its value in the range  $0.1 \dots 0.001$ . In contrast to the displacement-based formulation of Burman and Hansbo (2014), (23) allows for different materials in adjacent elements.

During the optimization process, isolated islands of material can emerge and develop in the design domain. This leads to a singular system of equations, as the rigid body modes associated to these free-floating material islands are not suppressed or constrained. To prevent this issue, we adopt the selective structural spring approach proposed and successfully implemented in Villanueva and Maute (2017), Geiss and Maute (2018), and Geiss et al. (2019b). An additional stiffness term is added to the weak form of the governing equations:

$$\mathcal{R}_{\text{Spring}} = \int_{\Omega^A} \gamma_S k_S \delta \mathbf{u} \cdot \mathbf{u} \, d\Omega, \quad (24)$$

where the parameter  $\gamma_S$  is set to a value ranging from 0 to 1 depending on the solution of an additional diffusion problem, so that a fictitious spring stiffness  $k_S$  is only applied to solid subdomains disconnected from any mechanical boundary conditions. The spring stiffness  $k_S$  is set to  $E/h^2$ , where  $E$  is Young's modulus of the considered material and  $h$  the element edge width.

## 6.2 The extended finite element method

The XFEM was developed by Moës et al. (1999) to model crack propagation without remeshing. The method allows capturing discontinuous or singular behaviors within a mesh element by adding specific enrichment functions to the classical finite element approximation. Here, following the work by Terada et al. (2003), Hansbo and Hansbo (2004), and Makhija and Maute (2014), a generalized Heaviside enrichment strategy with multiple enrichment levels is used. This particular enrichment enhances the classical finite element interpolation with additional shape functions which then results in independent interpolation into disconnected subdomains.

Considering a two-phase problem, the displacement field  $\mathbf{u}(\mathbf{x})$  is approximated with the Heaviside enrichment:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{m=1}^M \left( H(-\phi(\mathbf{x})) \sum_{i \in I^*} B_i(\mathbf{x}) u_{im}^A \delta_{ml}^{Ai} + H(\phi(\mathbf{x})) \sum_{i \in I^*} B_i(\mathbf{x}) u_{im}^B \delta_{ml}^{Bi} \right), \tag{25}$$

where  $I^*$  is the set of all the B-spline coefficients in the analysis mesh,  $B_i(\mathbf{x})$  is the B-spline basis function associated with the  $i^{th}$  coefficient, and  $u_{im}^I$  is the vector of displacement DOF associated with the  $i^{th}$  coefficient for material phase  $I = A, B$ . The number of active enrichment levels is denoted  $M$  and the Kronecker delta  $\delta_{ml}^{Ii}$  is used to select the active enrichment level  $m$  for the  $i^{th}$  coefficient and material phase  $I$ . This selection ensures the satisfaction of the PU principle (see Babuška and Melenk (1997)), as only one set of DOFs is used to interpolate the solution at a given point  $\mathbf{x}$ . For simplicity, the interpolation of the displacement field is restricted to first-order functions. The Heaviside function  $H$  is defined as:

$$H(z) = \begin{cases} 1, & \text{if } z > 0, \\ 0, & \text{if } z \leq 0. \end{cases} \tag{26}$$

Using the Heaviside enrichment, the integration of the weak form of the governing equations has to be performed separately on each material phase. Therefore, intersected elements are decomposed into integration subdomains, i.e., into triangles in two dimensions and into tetrahedra in three dimensions. For further details on the decomposition approach, the reader is referred to Villanueva and Maute (2014).

### 7 Optimization problem

In this paper, we focus on minimum compliance designs considering mass either via a constraint or a component of the objective. The optimization problem is formulated as follows:

$$\begin{aligned} \min_{0 \leq \mathbf{s} \leq 1} \quad & \mathcal{Z}(\mathbf{s}, \mathbf{u}(\mathbf{s})) + c_p \mathcal{P}_p(\mathbf{s})/\mathcal{P}_0 \\ & + c_\phi (\mathcal{P}_\phi(\phi(\mathbf{s})) + \mathcal{P}_{\nabla\phi}(\nabla\phi(\mathbf{s})))/\mathcal{P}_0 \tag{27} \\ \text{s.t.} \quad & \mathcal{M}^A(\mathbf{s})/\mathcal{M}(\mathbf{s}) - c_m \leq 0. \end{aligned}$$

The functional  $\mathcal{Z}$  is a weighted sum of the compliance and the mass, both evaluated over the solid domain:

$$\mathcal{Z}(\mathbf{s}, \mathbf{u}(\mathbf{s})) = w_s \frac{\mathcal{S}(\mathbf{s}, \mathbf{u}(\mathbf{s}))}{\mathcal{S}_0} + w_m \frac{\mathcal{M}^A(\mathbf{s})}{\mathcal{M}_0}, \tag{28}$$

where  $\mathcal{S}$  is the strain energy,  $\mathcal{S}_0$  is the initial strain energy value,  $\mathcal{M}^A$  the mass of material phase A,  $\mathcal{M}_0$  is the initial mass value, and  $w_s$  and  $w_m$  are the weighting factors. The perimeter associated to the domain boundary

is denoted by  $\mathcal{P}_p$  and its initial value is  $\mathcal{P}_0$ . The penalty functions  $\mathcal{P}_\phi$  and  $\mathcal{P}_{\nabla\phi}$  are used to regularize the level set field around and away from the interface and are further discussed in Section 7.1. The parameters  $c_p$  and  $c_\phi$  are the penalties associated with the perimeter and the regularization respectively. The mass constraint ensures that the ratio of material phase A,  $\mathcal{M}^A$ , and the total mass,  $\mathcal{M}$ , is smaller or equal to  $c_m$ . Note that enforcing the bounds on the design variable  $\mathbf{s}$  is conveniently achieved using THB-splines as they form a convex hull, while it requires extra handling when using HB-splines.

On purpose, we do not consider any means to control the feature size. One goal of this paper is to study the influence of the proposed adaptive hierarchical B-spline discretization on the optimization results, including the ability to resolve fine features. To allow the emergence of such features, no feature size control is imposed.

In this work, we solve the discretized optimization problem in the reduced space of only the design variables by a gradient-based algorithm. For each candidate design generated in the optimization process, the state variables are determined by solving the discretized governing equations (see Section 6).

### 7.1 Level set regularization scheme

To ensure that the spatial gradient near the solid-void interface is approximately uniform and matches a target gradient norm over the span  $[-\phi_{vt}, \phi_{vt}]$ , a level set regularization scheme is used. The scheme also enforces convergence to a target positive  $\phi_{up}$  or negative  $\phi_{low}$  value away from the interface, as depicted in Fig. 8.

Such a scheme requires the identification of the vicinity of the solid-void interface. The level set values are often used for this purpose. However, unless the LSF is constructed as an approximation or as an exact signed distance field, this approach lacks robustness and may lead to spurious oscillations in the level set field (see Geiss et al. (2019a)). In this study, we do not compute a signed distance field. Instead, we identify the proximity of a point to the solid-void interface by building a so-called *level of*

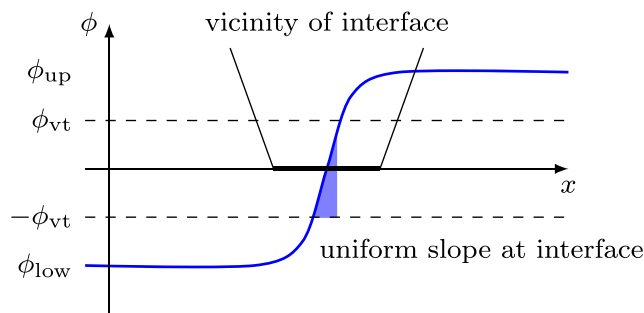


Fig. 8 Target level set function

neighborhood (LoN)  $\mathcal{I}(\mathbf{x})$  of a point with respect to a point belonging to an intersected element. The LoN is evaluated at the nodes of the considered mesh, providing the LoN value  $\mathcal{I}_i$  at each node  $i$ . The approach is illustrated in Fig. 9. The process starts with identifying intersected elements and setting the LoN  $\mathcal{I}$  of all nodes belonging to these elements to one; otherwise,  $\mathcal{I}$  is set to zero. In a recursive loop over all elements and nodes, first the elements having a node with a LoN value larger than 0 are flagged and then the LoN values of nodes belonging to a flagged element are increased by 1. The region with LoN values larger than 0 widens with the number of times the loop,  $N_{\text{LoN}}$ , is executed. Thus, the maximum LoN value  $\mathcal{I}_{\text{max}}$  equals  $N_{\text{LoN}}$ . The use of these nodal LoN values is discussed below.

To promote the convergence of the level set field to a target field, two penalty functions,  $\mathcal{P}_\phi$  and  $\mathcal{P}_{\nabla\phi}$ , are added to the optimization problem objective in (27). The first term penalizes the difference between the level set value  $\phi$  and, depending on its sign, the lower or upper target value  $\phi_{\text{target}}$  away from the interface:

$$\mathcal{P}_\phi = \int_{\Omega} \alpha_1 (1 - w) \left( \frac{\phi(\mathbf{s})}{\phi_{\text{target}}} - \text{sign}(\phi) \right)^2 d\Omega, \quad (29)$$

where  $\phi_{\text{target}}$  is set as a multiple of the element edge size of the initial mesh  $h_{\text{init}}$ , i.e., the mesh with the initial refinement level.

The second term penalizes both the difference between the norm of the spatial gradient of the level set field  $\nabla\phi$  and a target gradient norm  $\nabla\phi_{\text{target}}$  near the interface and the norm of the spatial gradient of the level set field  $\nabla\phi$  away from the interface:

$$\begin{aligned} \mathcal{P}_{\nabla\phi} = & \int_{\Omega} \alpha_2 w \left( \frac{\|\nabla\phi(\mathbf{s})\|}{\|\nabla\phi_{\text{target}}\|} - 1.0 \right)^2 d\Omega \\ & + \int_{\Omega} \alpha_3 (1 - w) \|\nabla\phi\|^2 d\Omega, \end{aligned} \quad (30)$$

where  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  are weighting factors. The parameter  $w$  is a measure of the distance of a point to the solid-void interface as:

$$w = e^{-\gamma_I(\mathcal{I}(\mathbf{x})/\mathcal{I}_{\text{max}}-1)^2}, \quad (31)$$

where  $\mathcal{I}(\mathbf{x})$  is the local LoN value, interpolated by the element shape functions using the nodal  $\mathcal{I}_i$  values. The parameter  $\gamma_I$  determines the region around and away from the interface in which the integrals in (29) and (30) are evaluated. As the value of  $\gamma_I$  is increased, the area considered to be in the vicinity of the interface, i.e., the area where the target slope  $\|\nabla\phi_{\text{target}}\|$  is promoted, shrinks. The parameters  $\phi_{\text{target}}$ ,  $\nabla\phi_{\text{target}}$ , and  $\gamma_I$  are user-defined.

### 7.2 Sensitivity analysis

In this study, we compute the design sensitivities by the adjoint approach. Let us consider an objective or constraint function  $\mathcal{F}(\mathbf{u}(\mathbf{s}), \mathbf{s})$  dependent on the design variables. The derivative of this response function with respect to a design variable  $s_i$  is computed as follows:

$$\frac{d\mathcal{F}(\mathbf{u}(\mathbf{s}))}{ds_i} = \frac{\partial\mathcal{F}}{\partial s_i} + \frac{\partial\mathcal{F}}{\partial\mathbf{u}} \frac{d\mathbf{u}}{ds_i}, \quad (32)$$

where the first term of the right-hand side accounts for the explicit dependency on the design variables and the second term for the implicit dependency on the design variables through the state variables. The implicit term is evaluated by the adjoint approach:

$$\frac{\partial\mathcal{F}}{\partial\mathbf{u}} \frac{d\mathbf{u}}{ds_i} = -\boldsymbol{\lambda}^T \frac{\partial\mathcal{R}}{\partial s_i}, \quad (33)$$

where  $\mathcal{R}$  is the residual defined in (20) for the forward analysis and  $\boldsymbol{\lambda}$  are the adjoint responses, evaluated through:

$$\boldsymbol{\lambda} = \frac{\partial\mathcal{F}}{\partial\mathbf{u}} \left[ \frac{\partial\mathcal{R}}{\partial\mathbf{u}} \right]^{-1}. \quad (34)$$

The derivatives with respect to the design variables are evaluated following the semi-analytical approach presented

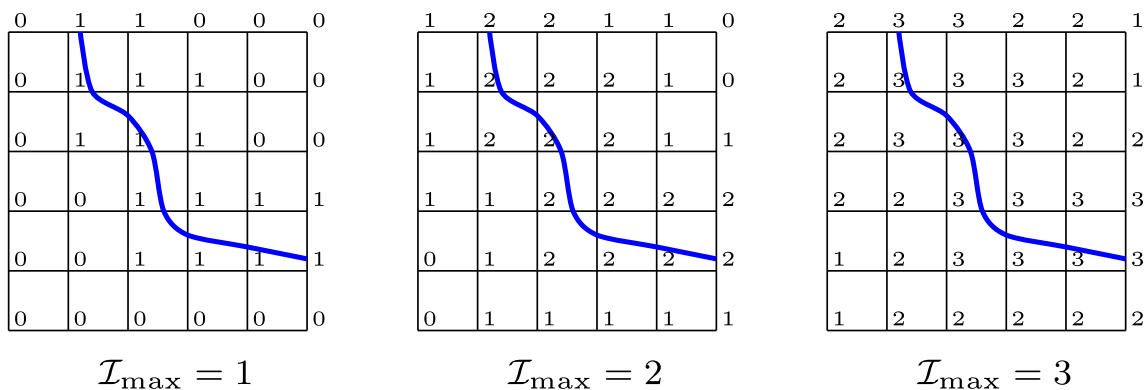


Fig. 9 Definition of the level of neighborhood (LoN)



in Sharma et al. (2017). It should be noted that all the stabilization parameters introduced for the forward analysis in Sections 6.1 and 7.1 are accounted for during the sensitivity analysis.

As truncated B-spline basis functions have smaller support than non-truncated ones (see Fig. 6 and 7), the zone of influence associated with a given truncated B-spline coefficient is smaller. Therefore, the perturbation of a truncated B-spline coefficient influences a smaller number of nodal level set values on the integration mesh and a reduced number of derivatives with respect to the level set nodal values needs to be evaluated per B-spline coefficient. Depending on the implementation of the sensitivity analysis, this property may reduce the computational cost.

When computing the design sensitivities with a LSM, only the intersected elements need to be considered for evaluating (34), as only the finite element residuals of these elements depend on the level set field. However, when the seeding approach presented in Section 3 is used, the residual contributions of non-intersected elements within the solid domain also depend on the design variables. Thus, in this case, the partial derivative  $\frac{\partial \mathcal{R}}{\partial \mathbf{u}}$  needs to be integrated over all elements in the solid domain.

## 8 Numerical examples

We study the proposed level set topology optimization with hierarchical mesh refinement with 2D and 3D examples, focusing on minimum compliance designs considering mass either via a constraint or a component of the objective as defined in (27).

The state variable field, i.e., the displacement field, is discretized in space with bilinear Lagrange quadrangular elements for 2D, and with trilinear Lagrange hexahedral elements for 3D design domains. The design variable field, i.e., the level set field, is discretized with HB- and THB-splines. The design variables are the B-splines coefficients and are used to interpolate nodal values on the Lagrange analysis mesh.

The optimization problems are solved by GCMMA from Svanberg (2002) and the required sensitivity analysis is performed following the adjoint approach described in Section 7.2. The parameters for the initial, lower, and upper asymptote adaptation in GCMMA are set to 0.05, 0.65, and 1.05, respectively. The optimization problem is considered converged if the absolute change of the objective function relative to the mean of the objective function in the five previous optimization steps drops below  $10^{-5}$  and the constraint is satisfied.

The refinement strategy described in Section 4.2 is adopted and the maximum refinement level for interface and

solid elements is set to  $l_{\text{ifc,max}} = l_{\text{solid,max}} = l_{\text{max}} = 4$ . The minimum refinement level is set to  $l_{\text{min}} = 0$ , the size of the coarsest mesh, the initial mesh refinement level, and the number of iterations between two mesh adaptations are given with each example.

The system of discretized governing equations and adjoint sensitivity equations are solved by the direct solver PARDISO for the 2D problems (see Kourounis et al. (2018)), and by a GMRES algorithm for 3D problems, preconditioned by an algebraic multi-grid solver (see Gee et al. (2006)). A relative drop of the linear residual of  $10^{-12}$  is required. This strict tolerance is imposed to reduce the influence of the iterative solver on the optimization results.

For the first two examples, we provide detailed comparisons between designs generated on uniformly and adaptively refined meshes in terms of performance, achieved geometries, and computational cost. Furthermore, we investigate the influence of the B-spline interpolation by varying the interpolation order.

The effect of applying or omitting truncation for the B-splines is investigated for 2D examples and presented in Appendix A. The numerical studies suggest that using THB-splines does not affect the geometric complexity, the performance, or the computational efficiency gains achieved with the proposed adaptive strategy. THB-Splines offer several advantageous properties; they form a PU, a desirable property when solving finite element problems. They also lead to an improved conditioning of the system of equations and offer a convenient way to impose bounds on design variables, i.e., an operation largely applied when solving optimization problems. Therefore, all subsequent optimization problems are solved using THB-splines.

Providing meaningful computational efficiency measures for the adaptive simulations is not trivial as various factors influence the computational cost. A simple wall-clock time comparison is provided for the 2D examples and takes into account both the time spent in the forward and sensitivity analyses. It is given as the ratio between the total runtime for the most refined uniform mesh and the total runtime for an adaptive mesh:

$$E_{\text{time}} = \frac{\sum_{k=1}^{N_{\text{opt}}} t_{\text{uniform}}^{(k)}}{\sum_{k=1}^{N_{\text{opt}}} t_{\text{adaptive}}^{(k)}}, \quad (35)$$

where  $t_{\text{uniform}}^{(k)}$  and  $t_{\text{adaptive}}^{(k)}$  are the runtimes for the uniform and adaptive meshes at the optimization iteration  $k$ .  $N_{\text{opt}}$  is the number of optimization steps until convergence and is set to  $N_{\text{opt}} = \min(N_{\text{opt,uniform}}, N_{\text{opt,adaptive}})$ .

Providing wall-clock times for all of the optimization examples presented in the paper would not be meaningful,

as different problem configurations of the same example were run on different hardware due to limited availability to dedicated computing resources. Furthermore, different linear solvers and software implementations lead to different dependencies of the computational time on the number of finite element DOFs. Therefore, in addition to a runtime comparison, if available, we measure the computational cost in terms of the evolution of the number of unconstrained DOFs of the finite element models during the optimization process.

An efficiency factor  $E_{xfem}$  is defined as the ratio between the total number of unconstrained DOFs in the XFEM model for the most refined uniform mesh and the total number of unconstrained DOFs in the XFEM model for an adaptive mesh. This measure is indicative of the computational gains achieved for XFEM solid-void problems and is expressed as follows:

$$E_{xfem} = \frac{\sum_{k=1}^{N_{opt}} \left( \#DOFs_{uniform,xfem}^{(k)} \right)^{n_s}}{\sum_{k=1}^{N_{opt}} \left( \#DOFs_{adaptive}^{(k)} \right)^{n_s}}, \tag{36}$$

where  $\#DOFs_{uniform,xfem}^{(k)}$  and  $\#DOFs_{adaptive}^{(k)}$  are the numbers of unconstrained DOFs in the XFEM models for the uniform and the adaptive meshes at the optimization iteration  $k$ . The parameter  $n_s$  is an exponent that can be fitted to relate the number of DOFs to the computational effort in terms of floating-point operations or wall-clock time (see Woźniak et al. (2014)), for example when a sparse direct linear solver is used and is set to  $n_s = 1$  for the 2D and  $n_s = 4/3$  for the 3D cases.

Alternatively, a measure of the peak resource requirement  $R_{xfem}$  is defined as the ratio between the maximum number of unconstrained DOFs in the XFEM model for the uniform mesh with highest refinement level and the maximum number of unconstrained DOFs in the XFEM model for the adaptive mesh:

$$R_{xfem} = \frac{\max \left( \#DOFs_{uniform,xfem} \right)^{n_s}}{\max \left( \#DOFs_{adaptive} \right)^{n_s}}. \tag{37}$$

It should be noted that for an XFEM model, the number of unconstrained DOFs is changing in the course of the optimization process, even on a uniformly refined mesh. For the solid-void problems studied in this paper and because the void domain is omitted in the analysis, the XFEM model reduces the number of DOFs compared with a classical FEM model. To account for these computational gains, we

define an efficiency factor  $E_{fem}$ , indicative for solid-solid XFEM or solid-void FEM problems, as:

$$E_{fem} = \frac{\sum_{k=1}^{N_{opt}} \left( \#DOFs_{uniform,fem}^{(k)} \right)^{n_s}}{\sum_{k=1}^{N_{opt}} \left( \#DOFs_{adaptive}^{(k)} \right)^{n_s}}, \tag{38}$$

where  $\#DOFs_{uniform,fem}^{(k)}$  is an approximation of the number of unconstrained DOFs for the uniform mesh with highest refinement level at the optimization iteration  $k$ , assuming that all DOFs are active as is the case for a classical FEM model.

A corresponding measure of the peak resource need  $R_{fem}$  is defined as follows:

$$R_{fem} = \frac{\max \left( \#DOFs_{uniform,fem} \right)^{n_s}}{\max \left( \#DOFs_{adaptive} \right)^{n_s}}. \tag{39}$$

The measures given by (36), (38), (37), and (39) for estimating the gains in computational efficiency are solely based on the costs for solving the linear finite element system. It does not account for the costs for building the XFEM model for initializing the linear solver, for the  $L^2$  projection of the design variable fields (see Section 4.2), for computing the right-hand side of the adjoint system in (33), and for the optimization algorithm. The latter computational costs are, however, small typically when compared with building and solving the system of governing equations.

### 8.1 Two-dimensional beam

As illustrated in Fig. 10, the first problem setup consists of a solid-void 2D beam with a length  $L = 6.0$  and a height  $l = 1.0$ . The solid is described by a linear elastic model and an isotropic constitutive behavior, with Young’s modulus  $E = 1.0$  and Poisson’s ratio  $\nu = 0.3$ . The beam is supported on its two lower corners over a length  $l_s = 0.025$  and a pressure  $p = 1.0$  is applied in the middle of its top span over a length  $l_p = 2 \times 0.025$ . The data are provided in self-consistent units.

The structure is designed for minimum compliance with a mass constraint of 40% of the total mass of the design domain. Taking advantage of the symmetry of the problem,

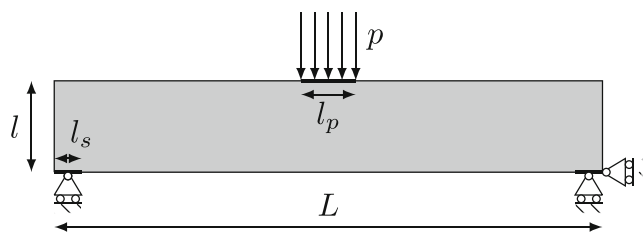


Fig. 10 Two-dimensional beam

only one-half of the design domain is considered. Table 1 summarizes the problem parameters.

The compliance minimization problem, defined in (27), is solved on uniformly and adaptively refined meshes, considering five levels of refinement  $l_{\text{refine}} = 0, 1, 2, 3,$  and  $4,$  corresponding to  $30 \times 10, 60 \times 20, 120 \times 40, 240 \times 80,$  and  $480 \times 160$  element meshes. The influence of the B-spline interpolation is investigated by solving the same design problem with linear, quadratic, and cubic B-spline functions. These influences are studied separately considering the two approaches for introducing inclusions, i.e., the initial hole seeding and the combined level set/density scheme.

### 8.1.1 Design with initial hole seeding

In this subsection, the design domain is initially seeded with holes, so that the mass constraint is satisfied at the beginning of the optimization process. The initial configuration is depicted in Fig. 11. The complexity of such a hole pattern requires a rather fine initial mesh and cannot be represented with the coarsest refinement levels  $l_{\text{ref}} = 0, 1.$  Thus, the optimization process needs to start from a mesh with  $l_{\text{ref}} > 1.$

Figure 12 shows the optimized designs generated on uniformly refined meshes with  $l_{\text{ref}} = 2, 3,$  and  $4$  using THB-splines and for different B-spline orders, along with the associated mechanical performance in terms of strain energy. The solid phase is depicted in black and the meshes are omitted as they are uniform. The designs generated on uniform meshes for different B-spline orders become increasingly similar as the mesh is refined. The

**Table 1** Parameter list for the 2D beam optimization problem

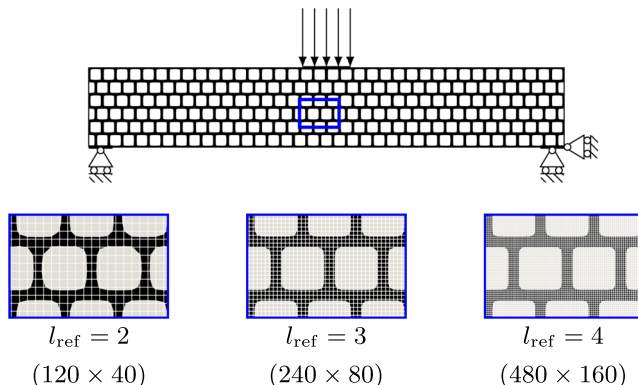
Parameter	Value
$w_s$	0.9
$w_m$	0.0
$c_p$	0.025
$c_\phi$	0.5
$c_v$	0.4
$\mathcal{S}_0$	59.12
$\mathcal{P}_0$	6.0
$\phi_{\text{sc}}$	$5h_{\text{init}}$
$V_{\text{max}}$	6.0
$\gamma_N$	$100E/h$
$\gamma_G$	0.005
$\phi_{\text{target}}$	$1.5h_{\text{init}}$
$\nabla\phi_{\text{target}}$	0.75
$\mathcal{I}_{\text{max}}$	1
$\gamma_I$	4.61
$\alpha_1 = \alpha_2 = \alpha_3$	0.5

performances of the designs are also rather similar. As expected, the results show that refining the mesh allows for a higher geometric complexity, including thinner structural members.

Figure 13 shows the optimized designs generated on adaptively refined meshes using linear, quadratic, and cubic THB-splines. For each layout, the solid phase is depicted in gray and the refined mesh is shown in the void phase. The computational gains achieved with the adaptive strategy are given in Table 2.

Applying hierarchical mesh refinement, the optimization process is started with an initial uniform mesh with a refinement level  $l_{\text{ref}}^0 = 2$  and is refined up to a refinement level  $l_{\text{ref,max}} = 4$  and coarsened to a refinement level  $l_{\text{min}} = 0.$  The minimum and maximum refinement levels are updated after every 10, 25, or 50 iterations. Additionally, the mesh is adapted to maintain a uniform refinement level in all intersected elements, as mentioned in Section 4.2.1. First, the designs demonstrate the ability of the method to recover similar layouts to the ones obtained on uniform meshes with higher mesh refinement. No significant loss is observed regarding the level of details or the complexity of the geometry achieved. The strain energy values are quite similar regardless of the use of uniform or adaptive meshes. The remaining differences in the strain energy values are not only due to differences in the designs, but also arise from differences in the discretization of the finite element model.

The results in Fig. 13 show, however, that the adaptive designs are sensitive to the number of iterations after which the refinement is applied. More frequent refinement operations, specifically early in the optimization process, allow for the emergence of finer features in the design. In terms of computational efficiency, the cost reduction with the adaptive strategy remains limited as initial configurations require rather fine meshes, i.e.,  $l_{\text{ref}} > 1.$  The runtime ratios and the efficiency factors in Table 2 show that the achieved gain slightly increases with the B-spline



**Fig. 11** Initial hole seeding for the 2D beam. Zoom on the hole pattern representation for the refinement levels  $l_{\text{ref}} = 2, 3,$  and  $4$

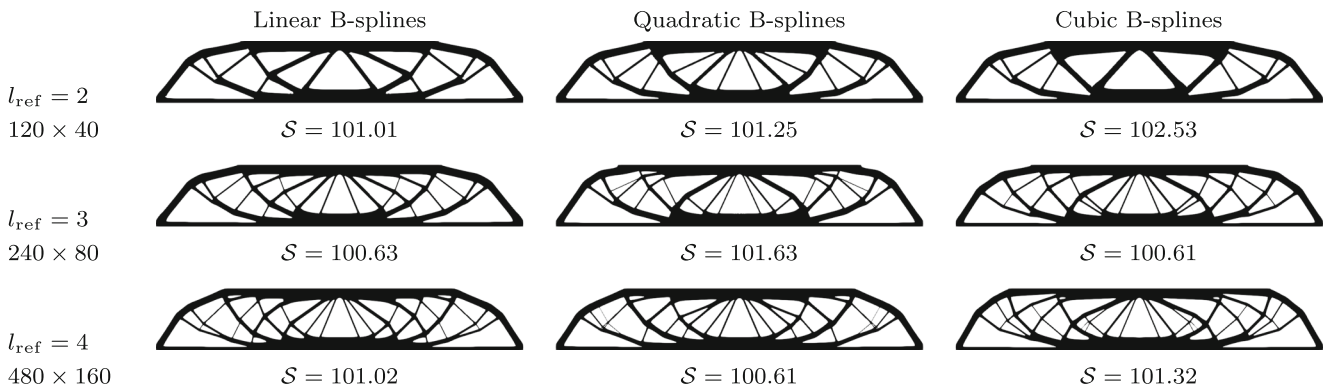


Fig. 12 2D beam using uniformly refined meshes with THB-splines and initial hole seeding

order. The largest factors, in terms of runtime, efficiency, and peak need, are achieved when the number of iterations before refinement is increased. Performing mesh refinement and updating the minimum and maximum refinement levels every 50 iterations lead to the largest computational gains. Using XFEM alone reduces the computational cost by about 50%. The adaptation strategy allows for a reduction of the computational cost by more than 3.5 over the uniform FEM model and by more than 1.75 over the uniform XFEM model. The peak resource needs  $R_{xfem}$  and  $R_{fem}$  follow the same trend.

Considering the refined meshes in Fig. 13, it is noticeable that the refined bandwidth around the solid-void interfaces increases in size with the order of the B-spline interpolation. This is a direct consequence of the refined buffer zone described in Section 4.2.2, whose size depends on the interpolation order. For both, the uniform and adaptive designs the influence of the B-spline interpolation order on the designs is mainly noticeable when working with lower order B-splines. Using a linear interpolation allows for the emergence of finer members and details in the structure, at the increased risk of converging to a local minimum with inferior performance. Higher order designs tend to be

smoother and to eliminate thin members. For this 2D case, it is noteworthy that working with linear B-splines does not lead to any smoothness issues or irregular shapes which might be caused by a spurious interplay between geometry and finite element prediction.

To further investigate the influence of mesh adaptivity, the optimization problem is solved starting from a uniform mesh with the highest refinement level, here  $l_{ref,max} = 4$  and using THB-splines. The mesh adaptation operation, triggered here every 25 iterations, involves coarsening of the mesh in void regions. The generated designs are shown in Fig. 14 along with the ones generated on the finest uniform mesh  $l_{ref} = 4$ . They slightly differ from the layouts obtained on uniform meshes, but present quite similar performance in terms of strain energy. These differences might result from the restart of the optimization process, as the mesh is adapted. After refinement, the level set field is mapped onto the newly generated mesh, which might result in a slight perturbation of the field values. Furthermore, the GCMMA is restarted with uniform lower and upper asymptotes for all variables which may alter the evolution of the design. Additionally, these results show that the obtained designs are dependent on the initial mesh refinement  $l_{ref}^0$ .

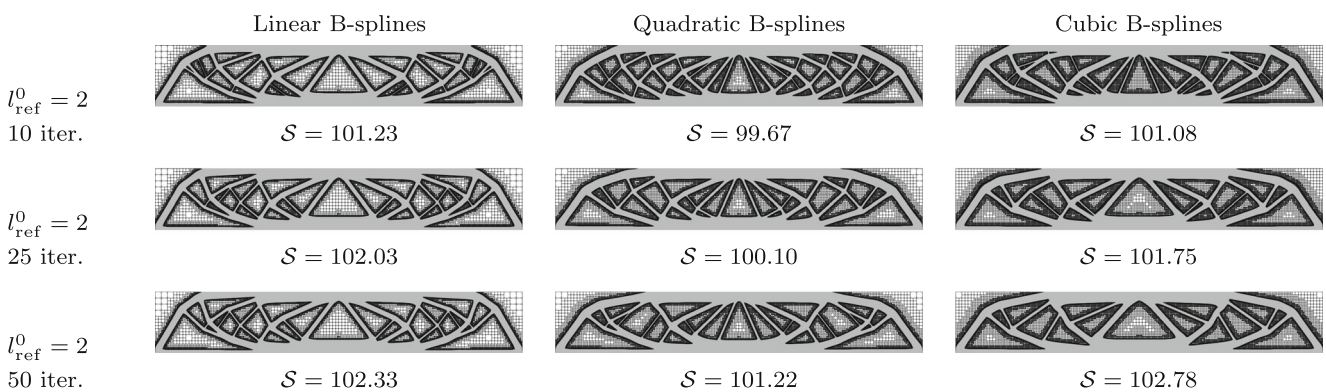


Fig. 13 2D beam using adaptively refined meshes with THB-splines and initial hole seeding. Initial refinement level of  $l_{ref}^0 = 2$  and maximum refinement level of  $l_{ref,max} = 4$

**Table 2** Performance in terms of computational cost for designs in Fig. 13

	Mesh update after 10 iter.			Mesh update after 25 iter.			Mesh update after 50 iter.		
	Lin.	Quad.	Cub.	Lin.	Quad.	Cub.	Lin.	Quad.	Cub.
$N_{\text{opt}}$	1648	1158	840	1460	1158	1254	1219	1158	1137
$E_{\text{time}}$	1.77	1.66	1.69	1.87	1.84	1.83	2.05	2.01	2.28
$E_{\text{xfem}}$	1.85	1.72	1.78	1.93	1.85	1.97	2.08	2.03	2.23
$R_{\text{xfem}}$	1.35	1.29	1.26	1.65	1.70	1.72	1.81	1.99	2.11
$E_{\text{fem}}$	3.81	3.56	3.61	3.94	3.81	4.05	4.20	4.19	4.58
$R_{\text{fem}}$	2.38	2.31	2.22	2.92	3.03	3.05	3.21	3.56	3.74

The efficiency factors are given in Table 3. They show that the initial refinement level affects the computational cost. In this case, the mesh adaptivity does not provide an actual computational gain with respect to an XFEM model, as  $E_{\text{xfem}}$  and  $R_{\text{xfem}}$  are about equal to 1.

### 8.1.2 Design with simultaneous hole seeding

In this section, the combined level set/density scheme, presented in Section 3, is used to nucleate holes during the optimization process. The combined scheme uses a SIMP exponent  $\beta = 2.0$ . The density shift parameter  $\rho_{\text{sh}}$  is initially set to 0.2 and is increased every 25 optimization steps. In this case, the optimization process can be started with rather coarse meshes. The minimum and maximum refinement levels are updated every 25 iterations. All other parameters are kept the same as in Section 8.1.1. Numerical studies have shown that the chosen strategy for updating the density shift parameter and the minimum/maximum refinement levels leads to satisfactory results. Refining this strategy or using alternate approaches may lead to improved performance and should be explored in future studies.

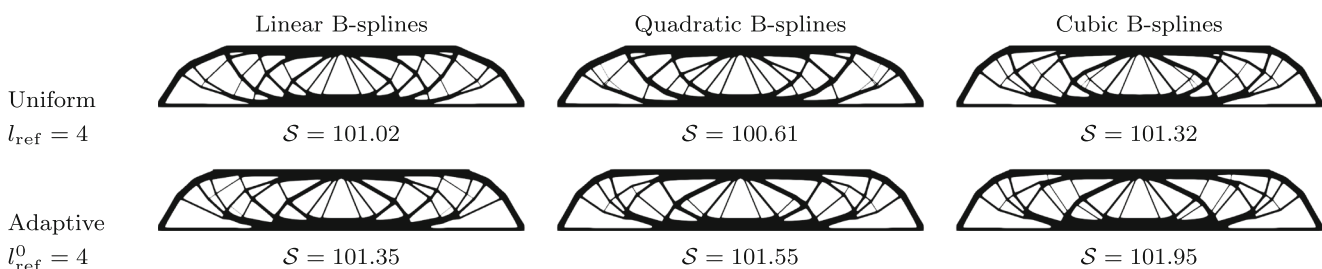
Figure 15 shows the optimized designs generated on uniformly refined meshes with  $l_{\text{ref}} = 1, 2, 3$ , and 4 using THB-splines and considering different B-spline orders, along with the associated performance in terms of strain energy. The solid phase is depicted in black and the meshes are omitted as they are uniform. For coarse meshes, the

designs exhibit different performances. The differences in the strain energy measure disappear as the meshes are refined. As expected, the complexity of the designs increases with mesh refinement. Remaining differences in both the geometry and the performance are likely caused by the non-convexity of the optimization problem.

Figure 16 shows the optimized designs generated on adaptively refined meshes with initial refinement level  $l_{\text{ref}}^0 = 1, 2$ , and 3 and using linear, quadratic, and cubic THB-splines. The performance in terms of strain energy values is provided for each design. For the adaptive designs, the mesh refinement is depicted in the void phase and the solid phase is represented in gray. The gains in terms of computational cost are given in Table 4.

Using the adaptive strategy, the results clearly show that, starting from coarser meshes, it is possible to recover designs rather similar, in terms of geometry and performance, to the ones obtained with finer uniform meshes, at least up to one or two refinement level higher. It is also noticeable that the layouts obtained with the refinement strategy depend on the initial mesh refinement. Finer structural members are more likely to develop and be maintained through the optimization process as the initial mesh is refined.

The runtime ratios and the efficiency factors are given in Table 4. They show that more significant gains are obtained when starting from coarser meshes, with lower refinement levels. The adaptation strategy allows for a



**Fig. 14** 2D beam using adaptively refined meshes with THB-splines and initial hole seeding. Initial refinement level of  $l_{\text{ref}}^0 = 4$  and maximum refinement level of  $l_{\text{ref,max}} = 4$ . Only coarsening is applied to adapt the mesh



**Table 3** Performance in terms of computational cost for designs in Fig. 14

	$l_{\text{ref}}^0 = 4$		
	Lin.	Quad.	Cub.
$N_{\text{opt}}$	1911	1158	1131
$E_{\text{xfem}}$	0.99	1.00	1.00
$R_{\text{xfem}}$	1.00	1.00	1.00
$E_{\text{fem}}$	2.05	2.05	2.04
$R_{\text{fem}}$	1.77	1.78	1.77

reduction of the computational cost by more than a factor of 3 over the uniform FEM model and by more than 1.75 over the uniform XFEM model. The achieved runtime and efficiency factors starting from the same initial refinement with different B-splines order match closely.

As a direct consequence of the buffer zone definition (see Section 4.2.2), the refined meshes in Fig. 16 exhibit a wider refined bandwidth as the interpolation order increases from linear to cubic. As observed in the previous section and in Figs. 15 and 16, linear B-splines can resolve finer structural members, while keeping a smooth description of the geometry at least in this 2D setting. Designs obtained with quadratic and cubic B-splines are smoother and similar.

A comparison of the results from Sections 8.1.1 and 8.1.2 suggests that the ability to nucleate holes during the optimization process is crucial to increase the computational cost reduction achieved with mesh adaptation. Creating an initial hole pattern requires the use of rather fine initial meshes, which limits the computational cost savings. Comparing the efficiency ratios in Tables 2 and 4 shows that starting from a  $120 \times 40$  initial mesh and refining every 25

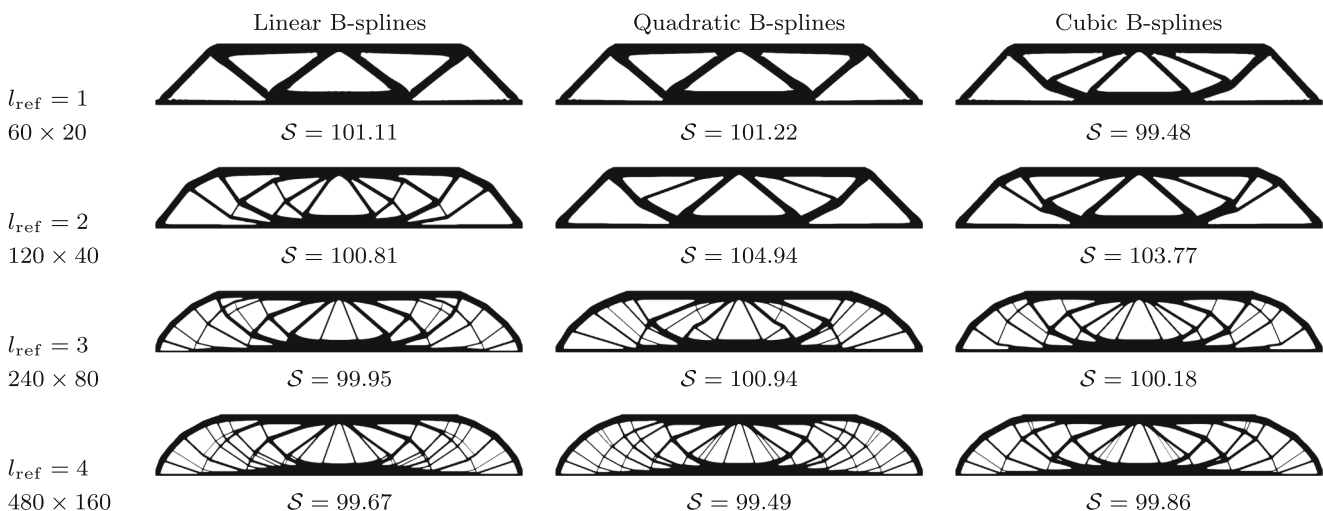
iterations, an efficiency factor  $E_{\text{xfem}}$  of about 2.0 is achieved for the initial seeding approach, while a ratio of about 2.5 is obtained for the combined scheme. Moreover, although a dependency on the initial mesh refinement is observed in both cases, the combined level set/density scheme seems to mitigate this dependency.

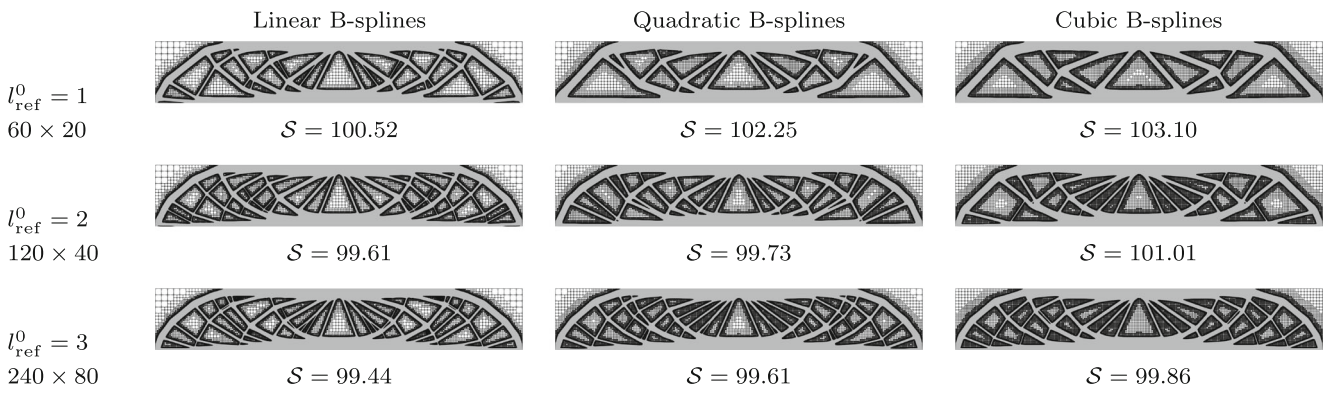
## 8.2 Three-dimensional beam

In this section, the previous 2D beam example is extended to 3D. This example allows studying the influence of the mesh adaptivity on 3D designs in terms of geometry, performance, and computational cost. In particular, we assess the ability of our approach to generate well-known three-dimensional optimal members such as shear webs that usually develop when solving optimization problems on fine uniform meshes. For 3D designs, the dependency on the initial hole pattern is mitigated by the ability to develop holes in the third dimension, which might counterbalance the performance issues encountered when using the initial seeding approach with the refinement strategy.

Figure 17 illustrates the setup for the solid-void 3D beam with a length  $L = 6.0$ , a height  $l = 1.0$ , and a thickness  $h = 1.0$ . The structural response is modeled by an isotropic linear elastic model with Young's modulus  $E = 1.0$  and Poisson's ratio  $\nu = 0.3$ . The structure is supported on its two lower sides over a length  $l_s = 0.025$  and a pressure  $p = 1.0$  is applied in the middle of its top span over an area  $l_p \times h = 2 \times 0.025 \times 1.0$ . The values above are provided in self-consistent units.

We optimize the beam for minimum compliance under a mass constraint of 10% of the total mass of the design domain. Taking advantage of the symmetry, we only study a quarter of the design domain. A list of problem parameters is given in Table 5.

**Fig. 15** 2D beam using uniformly refined meshes with THB-splines and simultaneous hole seeding



**Fig. 16** 2D beam using adaptively refined meshes with THB-splines and simultaneous hole seeding. Initial refinement levels of  $l_{ref}^0 = 1, 2,$  and  $3$  and maximum refinement level of  $l_{ref,max} = 4$

As for the 2D case, the initial hole seeding and the combined level set/density schemes are studied. The optimization problem in (27) is solved on uniformly and adaptively refined meshes considering four levels of refinements,  $l_{ref} = 0, 1, 2, 3$ , corresponding to  $18 \times 6 \times 3$ ,  $36 \times 12 \times 6$ ,  $72 \times 24 \times 12$ , and  $144 \times 48 \times 24$  element meshes. The influence of the B-spline interpolation order is investigated by solving the same problem with linear, quadratic, and cubic B-spline functions. All the simulations are performed with THB-splines.

**8.2.1 Design with initial hole seeding**

The initial seeding of holes in the design domain is presented in Fig. 18. This configuration satisfies the mass constraint, but requires a rather fine initial mesh to resolve the hole pattern, i.e.,  $l_{ref} > 1$ .

The optimized designs generated on uniformly and adaptively refined meshes are given in Figs. 19 and 20, respectively, along with their performance in terms of strain energy. Additionally, we depict vertical cross-sections of the designs in blue, to visualize the layouts' internal material arrangement. The efficiency factors are summarized in Table 6.

In Fig. 19, uniform meshes with levels of refinement  $l_{ref} = 2$  and  $3$  are considered, as coarser meshes cannot resolve the initial hole pattern. The design geometries and their performance in terms of strain energy are quite similar for linear B-splines. Each optimized layout is characterized by two shear webs and a truss-type structure that develops under the loaded surface. For higher B-spline orders, similar designs are observed only when using a finer mesh, i.e.,  $l_{ref} = 3$ . For coarser meshes with refinement level  $l_{ref} = 2$ , a single shear web develops in the center of the design. This behavior suggests that the obtained design depends on both the B-spline order and the refinement level. As the mesh is refined, smaller members are included in the layouts, which exhibit smoother surfaces. This can be observed in particular for the shear webs that become thin and smooth with mesh refinement.

Applying the refinement strategy every 25 iterations, the optimization process is started on a uniform mesh with a refinement level  $l_{ref}^0 = 2$  (see Fig. 20). In the course of the optimization process, the mesh is locally refined up to level  $l_{ref,max} = 3$  or coarsened to a level  $l_{min} = 0$ . The adaptive strategy generates designs that display a slightly higher geometric complexity and smoothness than the designs on corresponding initial uniform meshes. However, they are

**Table 4** Performance in terms of computational cost for designs in Fig. 16

	$l_{ref}^0 = 1$			$l_{ref}^0 = 2$			$l_{ref}^0 = 3$		
	Lin.	Quad.	Cub.	Lin.	Quad.	Cub.	Lin.	Quad.	Cub.
$N_{opt}$	671	550	547	941	539	433	941	484	454
$E_{time}$	2.57	3.24	3.31	1.79	2.11	2.05	1.53	1.60	1.58
$E_{xfem}$	2.91	3.58	3.69	2.15	2.54	2.70	1.66	1.79	1.82
$R_{xfem}$	4.28	4.98	5.04	3.13	3.16	3.42	1.76	1.81	1.83
$E_{fem}$	5.40	6.52	6.80	4.13	4.61	4.80	3.19	3.20	3.27
$R_{fem}$	4.28	4.98	5.04	3.13	3.16	3.42	1.76	1.81	1.83

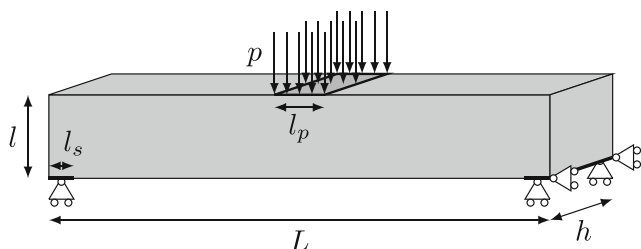


Fig. 17 Three-dimensional beam

visibly dependent on the initial mesh refinement as the obtained design topologies, i.e., two shear webs for linear B-splines versus one shear web for higher order B-splines, are similar to the ones generated on a uniform mesh with  $l_{ref} = 2$ . The efficiency ratios are given in Table 6. They show that a moderate reduction of the computational cost is achieved as the optimization process with adaptivity is started on a rather fine mesh, a reduction by a factor of about 4 over the uniform FEM model and by about 1.5 over the uniform XFEM model.

It is interesting to note that, as for the 2D case, linear B-splines support the development of smaller structural features such as the beams observed for both the uniform and the adaptive designs. At the same time, using these low-order B-splines leads to spurious oscillations of the level set field, which result in layouts with rough surfaces. Higher order B-splines promote the generation of smoother designs, as is the case for both quadratic and cubic B-splines.

### 8.2.2 Design with simultaneous hole seeding

In this section, holes are nucleated in the design domain during the optimization process using the combined level

Table 5 Parameter list for the 3D beam optimization problem

Parameter	Value
$w_s$	9.0
$c_p$	1.0
$c_\phi$	10.0
$c_v$	0.4
$\mathcal{P}_0$	10.0
$\mathcal{S}_0$	59.12
$\phi_{sc}$	$3h_{init}$
$V_{max}$	3
$\gamma_N$	$100E/h$
$\gamma_G$	0.01
$\phi_{target}$	$1.5h_{init}$
$\nabla\phi_{target}$	0.75
$\mathcal{I}_{max}$	1
$\gamma_I$	4.61
$\alpha_1 = \alpha_2 = \alpha_3$	0.5

Table 6 Performance in terms of computational cost for designs in Fig. 20

	$l_{ref}^0 = 2$		
	Lin.	Quad.	Cub.
$N_{opt}$	501	520	520
$E_{xfem}$	1.26	1.64	1.43
$R_{xfem}$	2.07	3.43	2.32
$E_{fem}$	3.90	5.44	4.64
$R_{fem}$	2.72	4.30	2.91

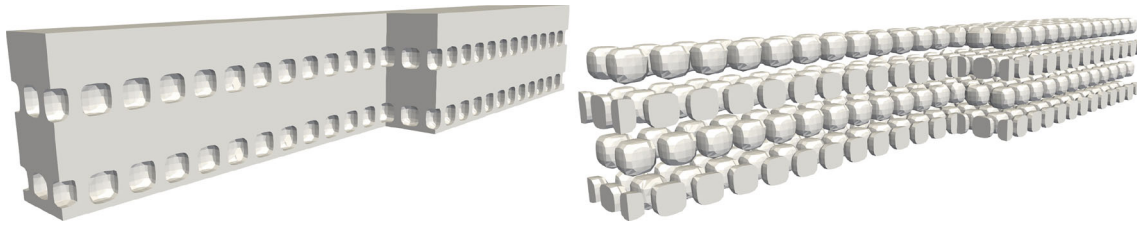
set/density scheme presented in Section 3. For the combined scheme, the SIMP exponent  $\beta$  is set to 2.0. The density shift parameter  $\rho_{sh}$  is initially set to 0.2 and is updated every 25 optimization steps. Contrary to Section 8.2.1, coarser meshes can be used as a starting point for the adaptive optimization, since the initial design geometry is not complex.

The designs obtained on uniformly and adaptively refined meshes are shown in Figs. 21 and 22 with their performance in terms of strain energy. Additionally, we depict vertical cross-sections of the designs in blue, to visualize the layouts' internal material arrangement. Table 7 gives the design performance in terms of computational cost.

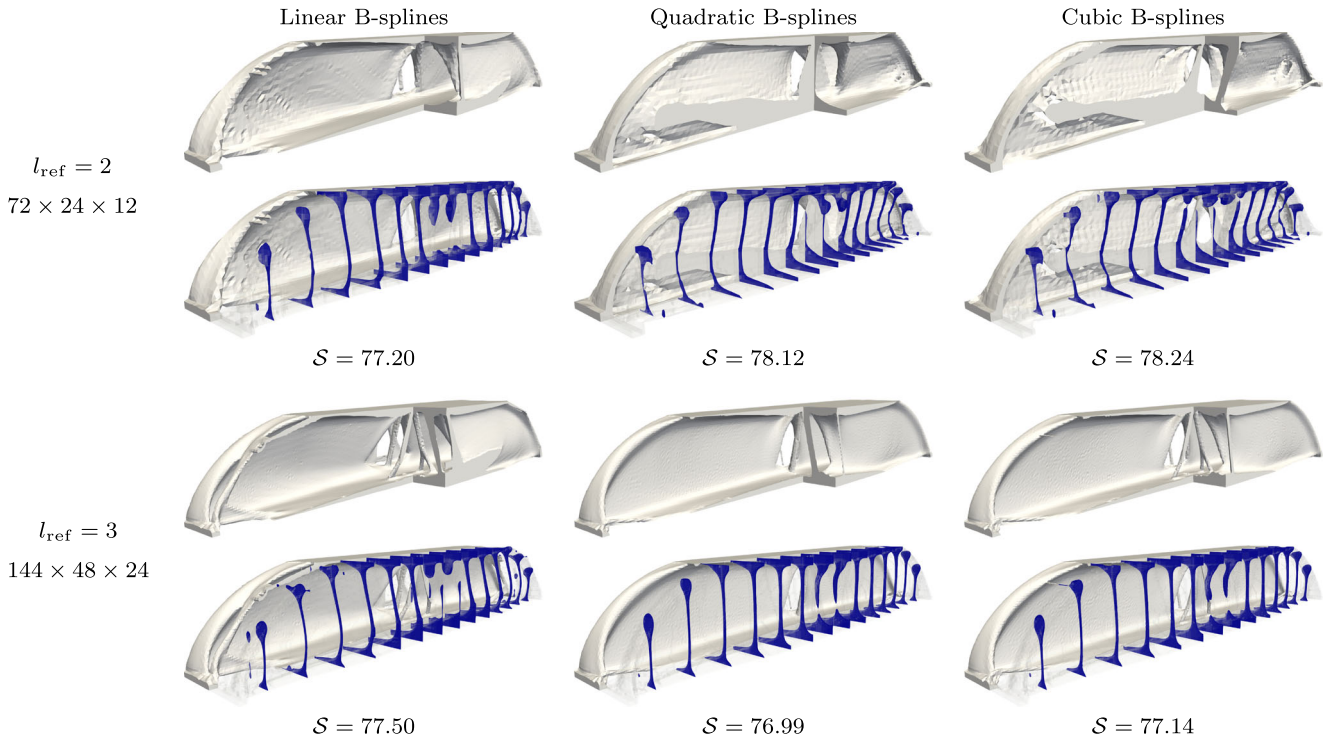
In Fig. 21, three levels of refinement are considered  $l_{ref} = 1, 2, \text{ and } 3$ . The uniform design geometries are quite similar, as each layout is characterized by two shear webs and a truss-type structure that develops under the loaded surface. Also, the performance in terms of strain energy is quite similar for the designs obtained on meshes with refinement levels  $l_{ref} = 2$  and 3. These results demonstrate that the combined level set/density scheme reduces the dependency on the initial design that was observed with the initial hole seeding approach. Using extremely coarse meshes with  $l_{ref} = 1$  leads to spurious oscillations of the level set field

Table 7 Performance in terms of computational cost for designs in Fig. 22

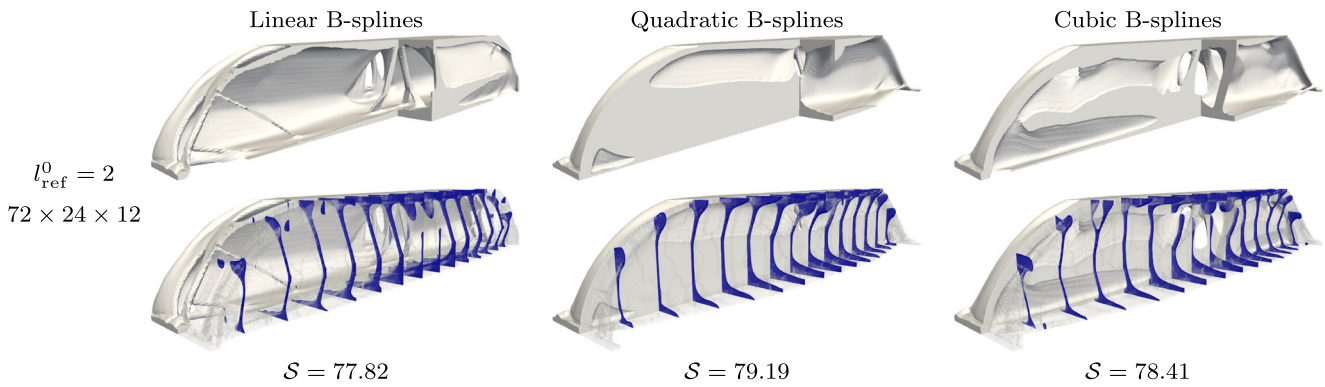
	$l_{ref}^0 = 1$		
	Lin.	Quad.	Cub.
$N_{opt}$	292	209	275
$E_{xfem}$	2.87	4.08	3.09
$R_{xfem}$	3.78	3.75	3.92
$E_{fem}$	6.70	8.78	7.56
$R_{fem}$	3.78	3.75	3.92



**Fig. 18** Initial hole seeding for the 3D beam. Representation of the solid on the left and the void on the right



**Fig. 19** 3D beam using uniformly refined meshes with THB-splines and initial hole seeding



**Fig. 20** 3D beam using adaptively refined meshes with THB-splines and initial hole seeding. Initial refinement levels of  $l_{\text{ref}}^0 = 2$  and maximum refinement level of  $l_{\text{ref}, \text{max}} = 3$

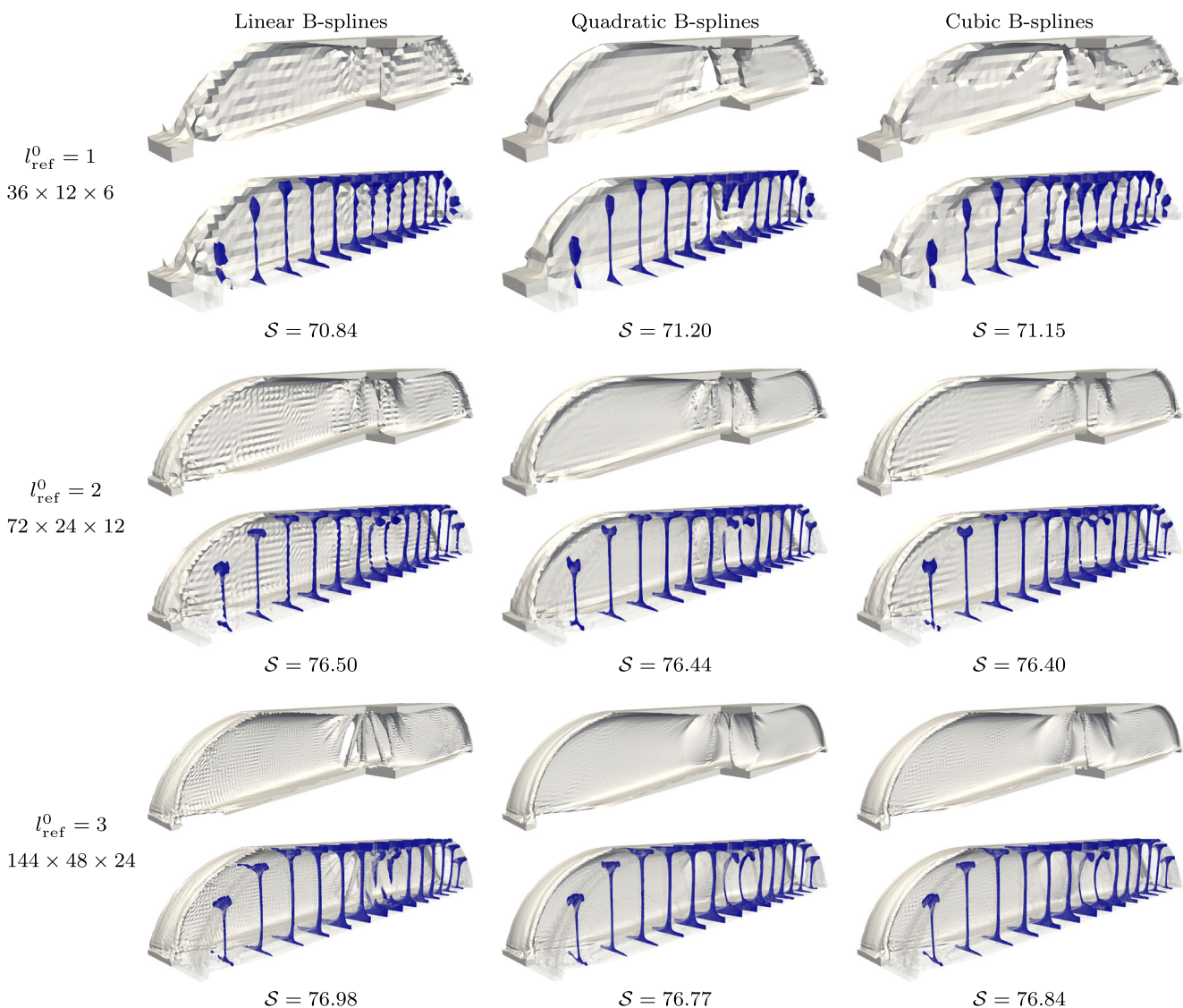


and thus a rough solid/void interface surface. This behavior is more strongly pronounced when using linear B-splines.

Working with an adaptive mesh, the optimization process is started on a uniformly refined mesh with the lowest level of refinement  $l_{\text{ref}}^0 = 1$  and is refined up to  $l_{\text{ref,max}} = 3$  or coarsened to  $l_{\text{min}} = 0$  (see Fig. 22). The results demonstrate the ability of the adaptive optimization approach to create designs similar to the ones obtained on uniform meshes with a higher level of refinement. The designs also compare well in terms of strain energy performance. The refinement strategy leads to a significant reduction of the computational cost, as the optimization is started on a coarser mesh. Table 7 shows high efficiency ratios, i.e., a reduction by a factor of about 7 over the uniform FEM model and by about 3 over the uniform XFEM model.

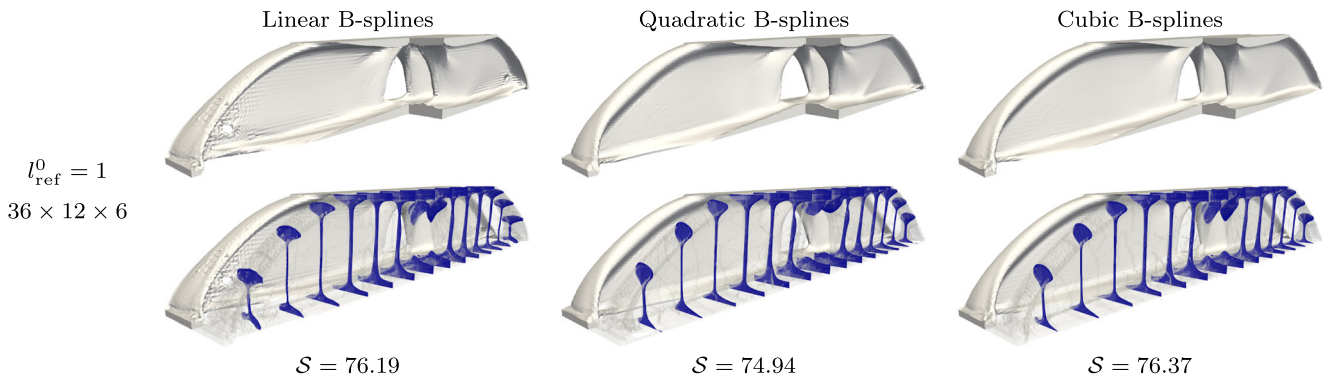
As observed in the previous section, using low-order B-splines promotes the development of thinner features, which is not observed with higher polynomial orders. However, linear B-spline designs lack smoothness, even as the mesh is refined.

The effect of hole seeding during the optimization process is not as pronounced as in 2D. In 3D, holes can be created more easily by effectively changing the thickness, i.e., changing the shape in the third dimension. The main advantage of the combined scheme lies in the ability to use coarser meshes as a starting point for the adaptive topology optimization, which leads to larger efficiency ratios  $E_{\text{xfem}}$ , i.e., around 3 for the combined scheme versus around 1.5 with the initial seeding strategy when compared with uniform XFEM meshes.



**Fig. 21** 3D beam using uniformly refined meshes with THB-splines and simultaneous hole seeding



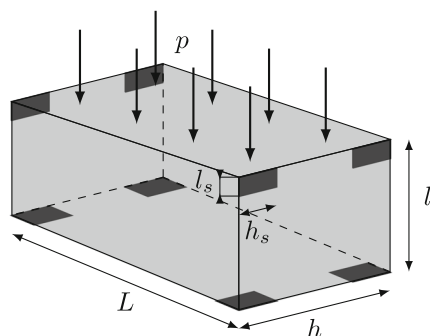


**Fig. 22** 3D beam using adaptively refined meshes with simultaneous hole seeding. Initial refinement levels of  $l_{\text{ref}}^0 = 1$  and maximum refinement level of  $l_{\text{ref,max}} = 3$

Both the 2D and 3D beam examples show that using linear B-splines allows for the development of thinner structural members. However, in the 3D case, the latter details develop at the cost of spurious oscillations in the level set field, leading to a rough representation of the surface. Higher order B-splines promote the development of smoother designs. In 2D and in 3D, the designs generated with quadratic and cubic B-splines match closely. Therefore, all upcoming simulations are carried out with quadratic B-splines. They lead to smooth designs without reducing the complexity of the geometry achieved. Moreover, they present smaller support and a lower level of complexity than cubic B-splines, which leads to simpler implementation and a reduced overall computational cost.

### 8.3 Short beam under uniform pressure

This final example aims at demonstrating the capabilities of the proposed mesh adaptivity to generate geometrically complex designs with limited computational resources. The optimization problem formulation is modified to generate designs with increasingly lower volume fractions and show that mesh adaptation allows for the development of extremely thin structural members.



**Fig. 23** Three-dimensional short beam under uniform pressure

We consider a short beam subject to a uniform pressure on its top. The design domain with boundary conditions is shown in Fig. 23 and is defined by a rectangular cuboid with a length  $L = 2.0$ , a height  $l = 0.6$ , and a thickness  $h = 1.0$ . The structural response is described by an isotropic linear elastic model with Young's modulus  $E = 6.9 \cdot 10^{10}$  and Poisson's ratio  $\nu = 0.3$ . The presence of material is enforced on the top of the table, where a uniform pressure load  $p = 0.67 \cdot 10^6$  is applied. The structure is supported on the side at the top corners, and on the bottom at each corner with  $l_s = 0.01$  and  $h_s = 0.2$ .

We optimize the structure for minimum compliance and mass; no constraints are imposed. Applying the adaptive refinement strategy, we focus on the influence of the mass

**Table 8** Parameter list for the 3D short beam under uniform pressure design optimization problem

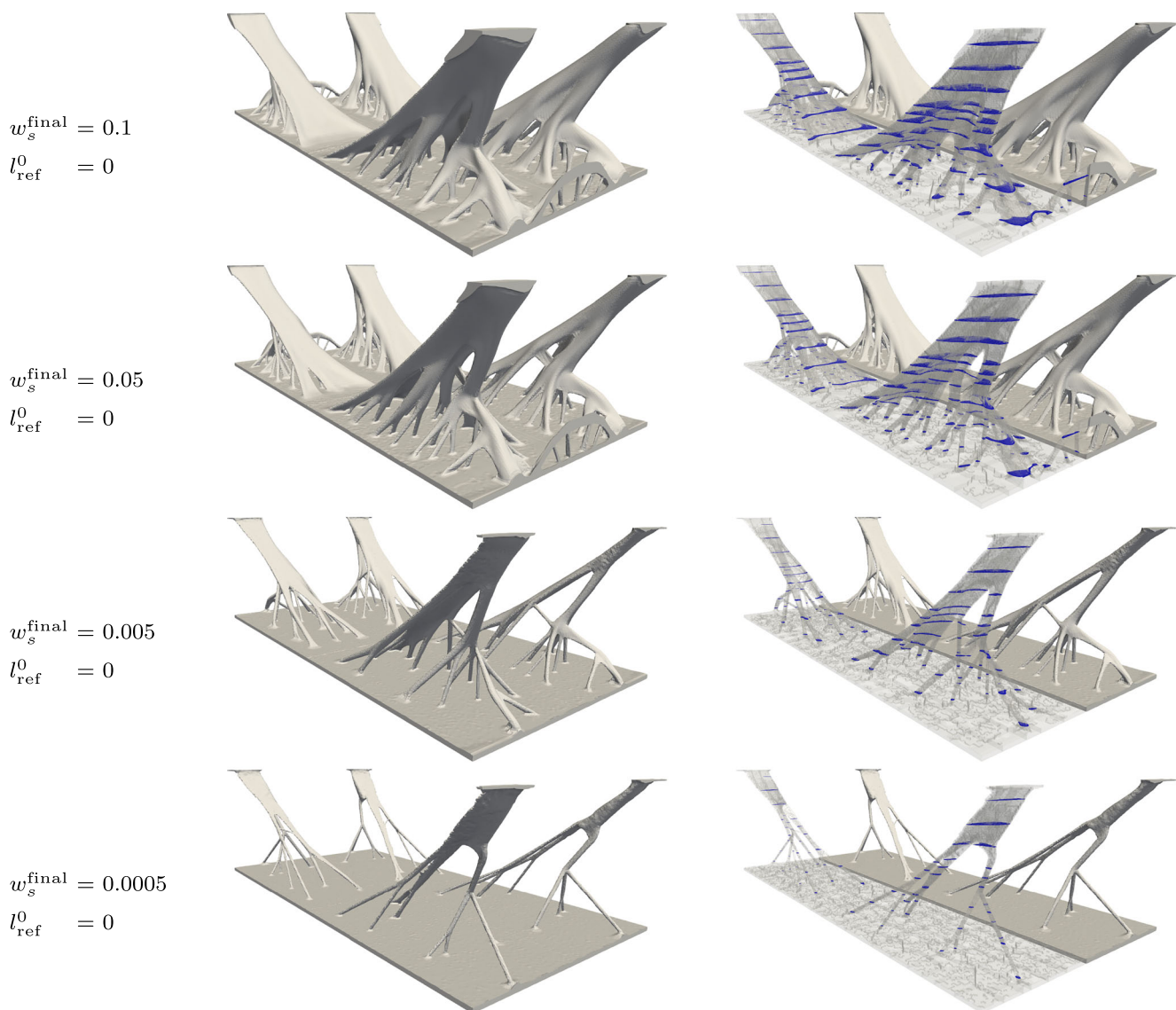
Parameter	Value
$w_s^0$	0.5
$w_m$	10.0
$c_p$	0.05
$c_\phi$	5.0
$c_v$	1.0
$S^0$	61.93
$\mathcal{M}^0$	0.3
$\mathcal{P}^0$	3
$\phi_{\text{sc}}$	$3 h_{\text{init}}$
$\gamma_N$	$100 E/h$
$\gamma_G$	0.01
$\phi_{\text{target}}$	$1.5 h_{\text{init}}$
$\nabla \phi_{\text{target}}$	0.75
$\mathcal{I}_{\text{max}}$	1
$\gamma_I$	4.61
$\alpha_1 = \alpha_2 = \alpha_3$	0.5

of the designs on the achieved computational cost reduction. Taking advantage of the symmetry, we only study a quarter of the design domain. The problem parameters are given in Table 8 in self-consistent units.

We investigate the effect of the designs mass on the adaptive strategy. To generate designs with various strain energy/mass ratios, we vary the compliance weighting parameter  $w_s$  value. To prevent an overly aggressive removal of mass early on in the optimization process, we apply a continuation on  $w_s$ , which is initially set to  $w_s^0 = 0.5$  and is gradually decreased during the optimization process to the prescribed values  $w_s^{\text{final}} = 0.1, 0.05, 0.005,$  and  $0.0005$ . We should note that the strategy for reducing

$w_s$  impacts the computational costs. Large  $w_s$  values put more emphasis on the strain energy. Thus, stiffer structures, using more mass, are generated, which in turn requires more elements to be resolved at a particular refinement level.

Based on the observations made for the 2D and 3D beam problems in Sections 8.1 and 8.2, we study this problem with the level set/density combined scheme only, as it allows to initiate the design process on a rather coarse mesh. The SIMP exponent  $\beta$  is set to 1.5. The density shift parameter  $\rho_{\text{sh}}$  is initially set to 0.2 and is updated every 25 optimization steps. Furthermore, the design variable field is interpolated with quadratic THB-splines. We consider four levels of refinement  $l_{\text{ref}} = 0, 1, 2,$  and  $3$ , corresponding to  $24 \times 14 \times$



**Fig. 24** Three-dimensional short beam under uniform pressure design using adaptively refined mesh with quadratic THB-splines and simultaneous hole seeding. Initial refinement level of  $l_{\text{ref}}^0 = 0$  and maximum refinement level of  $l_{\text{ref,max}} = 3$

**Table 9** Performance in terms of strain energy, mass, and volume fraction for designs in Fig. 24

$w_s^{\text{final}}$	Strain energy	Mass	Volume fraction
0.1	349.38	0.0232	5.57
0.05	469.24	0.0189	4.14
0.005	1442.14	0.0115	1.68
0.0005	4421.07	0.0098	1.19

12,  $48 \times 28 \times 24$ ,  $96 \times 56 \times 48$ , and  $192 \times 112 \times 96$  element meshes. The problem is solved using adaptively refined meshes, starting from an initial mesh with a refinement level  $l_{\text{ref}}^0 = 0$  and allowing refinement up to  $l_{\text{ref,max}} = 3$  and coarsening to  $l_{\text{min}} = 0$ . Solving the problem on uniformly refined meshes is omitted.

The designs generated on adaptively refined meshes with decreasing volume fractions are shown in Fig. 24. Additionally, we depict horizontal cross-sections of the designs in blue, to visualize the layouts' internal material arrangement. The performance in terms of strain energy, mass, and the volume ratios is given in Table 9. As the mass of the design decreases, very thin structural members are generated. These designs demonstrate the ability of the adaptive mesh strategy to resolve fine members and details starting from rather coarse meshes.

Table 10 gives the iteration  $k_{\text{peak}}$  at which the peak number of free DOFs is reached, the peak number of free DOFs and the number of free DOFs at the end of the optimization process. The peak number of DOFs is similar for each problem setup and is recorded around iteration 95, i.e., after the second refinement step. The final number of unconstrained DOFs shows that as the mass of the designs becomes smaller, the associated number of free DOFs in the system drops, leading to a reduced computational time.

Table 10 also provides the efficiency factors evaluated with  $N_{\text{opt}}$  being the number of iterations to convergence. Here, we only consider the efficiency factors with respect to an equivalent FEM model, i.e.,  $E_{\text{fem}}$  and  $R_{\text{fem}}$  considering

a uniformly refined mesh with  $l_{\text{ref}} = 3$ . The efficiency ratio  $E_{\text{fem}}$  increases as the mass of the designs decreases, i.e., the lower the volume fraction, the larger the computational saving. Since all cases start with  $w_s = 0.5$  and since the peak DOFs count is recorded around iteration 95 for each design, the  $R_{\text{fem}}$  values are almost the same for all cases. The strategy for reducing  $w_s$  could be improved to increase the computational efficiency due to mesh adaptation.

## 9 Conclusions

This paper presents an adaptive mesh refinement strategy using hierarchical B-splines to perform level set topology optimization. The geometry of the design is described by a LSF and the analysis is performed using the XFEM. The problem of simultaneously achieving a precise description of the geometry and an accurate evaluation of the physical responses at an acceptable computational cost is addressed by hierarchical mesh refinement, leading to an adaptive discretization of the design and state variable fields. After a user-specified number of optimization steps, local refinement is performed based on a user-defined geometric criterion, here the proximity to the solid/void interface. A sufficient freedom in the design is ensured by considering an initial hole seeding approach and a hole nucleation approach based on a combined level set/density scheme. The proposed method is applied to 2D and 3D structural solid-void topology optimization problems considering mass and strain energy. Optimized designs are generated on uniformly and adaptively refined meshes and compared in terms of geometric complexity, design performance, and computational cost.

The numerical experiments suggest that higher order B-splines promote the development of smooth designs and limit the need for filtering techniques required with low-order interpolation functions. Contrary to using coarser meshes, higher order B-splines do not limit the ability to represent fine members and thus do not restrict the design

**Table 10** Performance in terms of computational cost for designs in Fig. 24

	$w_s^{\text{final}} = 0.1$	$w_s^{\text{final}} = 0.05$	$w_s^{\text{final}} = 0.005$	$w_s^{\text{final}} = 0.0005$
$k_{\text{peak}}$	95	94	94	95
Peak # DOFs	1910815	1852155	1867655	1836490
End # DOFs	1289080	1128070	758425	758425
$N_{\text{opt}}$	218	220	252	273
$E_{\text{fem}}$	23.16	25.56	31.01	32.73
$R_{\text{fem}}$	9.79	10.21	10.10	10.32

space. Moreover, introducing truncation allows imposing accurately bounds on the design variables.

Additionally, the numerical simulations show that the ability to nucleate holes during the optimization is crucial for increasing the computational advantage provided by mesh adaptation. The initial hole seeding approach limits the adaptive strategy performance, as rather fine meshes are required to resolve complex initial hole patterns.

Although the choice of the initial refinement influences the optimized designs, the numerical studies demonstrate that the adaptive strategy allows for the development of complex features, such as thin shear webs in 3D. Moreover, the adaptive designs exhibit similar performance when compared with the uniform ones. The evaluation of efficiency factors in terms of the number of unconstrained DOFs shows that the adaptive strategy provides an improvement in terms of computational cost. In general, the efficiency factors increase both with the order of the B-spline interpolation, with the coarseness of the initial mesh and, for 3D problems, with a drop in the volume fraction of the optimized designs.

Future work will focus on the following: (i) exploring alternative refinement strategies to further increase the accuracy and computational efficiency, and (ii) addressing problems with more complex physics. With regard to the refinement strategy, the method could be extended to handle the design and state variable fields separately. Other refinement criteria should be considered, such as finite element error indicators. Additionally, the B-spline interpolation should be extended to higher order B-splines for the state variable field. Finally, as more complex physics are considered, allowing for a more dynamic refinement approach could be crucial to resolve, for example, transient problems.

**Acknowledgments** The first, third, fourth, and fifth authors received the support for this work from the Defense Advanced Research Projects Agency (DARPA) under the TRADES program (agreement HR0011-17-2-0022).

**Compliance with ethical standards** The opinions and conclusions presented in this paper are those of the authors and do not necessarily reflect the views of the sponsoring organizations.

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

## Appendix Hierarchical B-splines versus truncated hierarchical B-splines

The influence of the truncation operation on the B-splines is investigated by solving the 2D beam problem, presented in Section 8.1, with HB-splines and THB-splines on an

adaptive mesh with an initial refinement level  $l_{\text{ref}}^0 = 2$  and applying the refinement operation every 25 iterations first for the initial hole seeding approach and then for the level set/density scheme.

It should be noted that extra treatment is required to impose bounds on the design variables when working with HB-splines, as they do not constitute a PU. Bounds are enforced by clipping the variable values with the upper or lower allowed values. The clipping operation is only applied to the combined level set/density scheme as the density values should remain between 0 and 1.

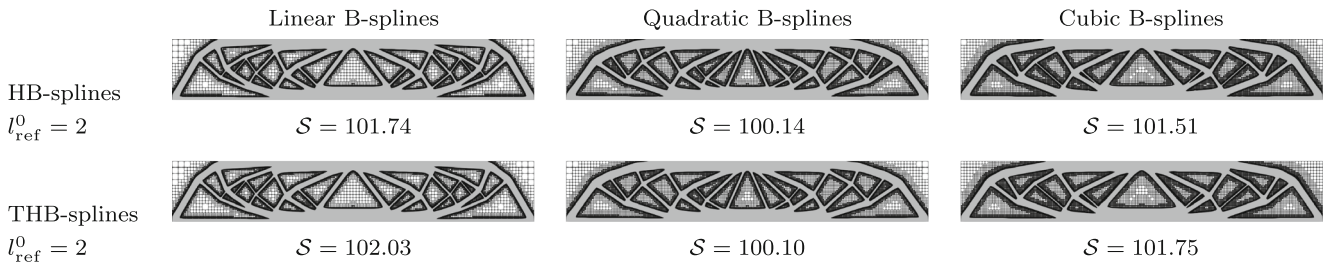
### A.1 2D beam with initial hole seeding

The problem setting is identical to the one presented in Section 8.1.1, but with a comparison of HB-splines and THB-splines. The obtained designs and corresponding strain energy values are given in Fig. 25. The linear, quadratic, and cubic designs only differ slightly. The remaining differences between the designs can be explained by the support size of the HB- and THB-splines that differs, as shown in Fig. 7. This is further supported by the maximum stencil size recorded for each design variables, i.e., the maximum number of design coefficients affected by the change in a specific design coefficient. The stencil sizes for the HB-splines are 8, 16, and 33 for the linear, quadratic, and cubic orders, against 2, 9, and 16 for the THB-splines. The runtime ratios and the efficiency factors are given in Table 11 and match closely for HB- and THB-splines.

### A.2 2D beam with simultaneous hole seeding

The problem setting is identical to the one presented in Section 8.1.2 but with a comparison of HB-splines and THB-splines. The obtained designs and corresponding final strain energy values are given in Fig. 26. The designs differ more significantly for HB- and THB-splines than when working with level set only (see Fig. 25). These differences can be partly explained by the maximum stencil size difference between HB- and THB-splines, i.e., 7, 16, and 37 for linear, quadratic, and cubic HB-splines against 2, 9, and 16 for the THB-splines. On top of the stencil size mismatch, these differences in the final designs can be explained by the clipping operation applied to the density values to keep them between 0 and 1 when working with HB-splines. Clipping yields non-differentiability with respect to the clipped values which can influence the optimization process. The runtime ratios and the computational gain factors are given in Table 12 and match closely for both HB- and THB-splines.

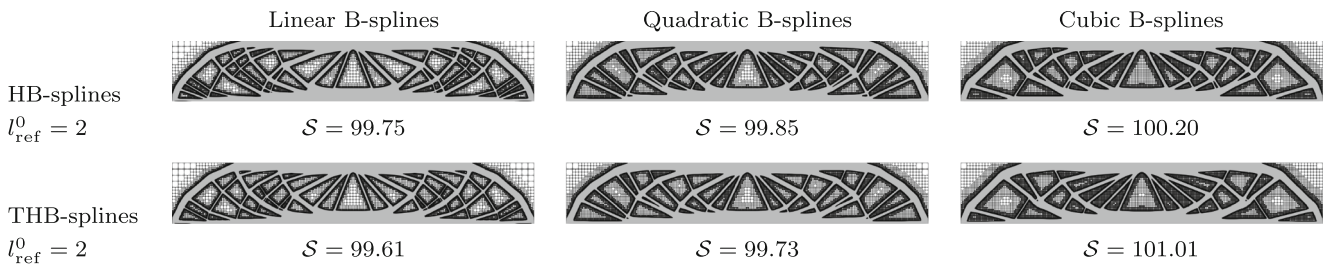




**Fig. 25** 2D beam using adaptively refined meshes with initial hole seeding. Initial refinement level of  $l_{\text{ref}}^0 = 2$ , maximum refinement level of  $l_{\text{ref,max}} = 4$ . Comparison of HB- and THB-splines

**Table 11** Performance in terms of computational cost for designs in Fig. 25

	HB-splines			THB-splines		
	Lin.	Quad.	Cub.	Lin.	Quad.	Cub.
$N_{\text{opt}}$	1385	1158	1254	1460	1158	1254
$E_{\text{time}}$	1.82	1.78	1.80	1.87	1.84	1.83
$E_{\text{xfem}}$	1.93	1.83	1.97	1.93	1.85	1.97
$R_{\text{xfem}}$	1.65	1.70	1.74	1.65	1.70	1.72
$E_{\text{fem}}$	3.93	3.78	4.06	3.94	3.81	4.05
$R_{\text{fem}}$	2.92	3.04	3.08	2.92	3.03	3.05



**Fig. 26** 2D beam using adaptively refined meshes with simultaneous hole seeding. Initial refinement level of  $l_{\text{ref}}^0 = 2$  and maximum refinement level of  $l_{\text{ref,max}} = 4$ . Comparison of HB- and THB-splines

**Table 12** Performance in terms of computational cost for designs in Fig. 26

	HB-splines			THB-splines		
	Lin.	Quad.	Cub.	Lin.	Quad.	Cub.
$N_{\text{opt}}$	941	405	542	941	539	433
$E_{\text{time}}$	1.78	2.13	1.88	1.79	2.11	2.05
$E_{\text{xfem}}$	2.16	2.71	2.52	2.15	2.54	2.70
$R_{\text{xfem}}$	3.31	3.15	3.29	3.13	3.16	3.42
$E_{\text{fem}}$	4.14	4.72	4.65	4.13	4.61	4.80
$R_{\text{fem}}$	3.31	3.15	3.29	3.13	3.16	3.42



**Replication of results** Upon request, the authors will provide the full set of input parameters for each topology optimization problems presented in the paper. Additionally, the initial meshes used for the designs with uniform refinement and the succession of meshes created through the optimization processes for the designs with adaptive refinement will be provided upon request.

## References

- Allaire G, Jouve F, Toader A-M (2004) Structural optimization using sensitivity analysis and a level-set method. *J Comput Phys* 194(1):363–393. <https://doi.org/10.1016/j.jcp.2003.09.032>
- Babuška I, Melenk JM (1997) The partition of unity method. *Int J Numer Methods Eng* 40(4):727–758. [https://doi.org/10.1002/\(SICI\)1097-0207\(19970228\)40:4<727::AID-NME86>3.0.CO;2-N](https://doi.org/10.1002/(SICI)1097-0207(19970228)40:4<727::AID-NME86>3.0.CO;2-N)
- Bandara K, Rüberg T, Cirak F (2016) Shape optimisation with multiresolution subdivision surfaces and immersed finite elements. *Comput Methods Appl Mech Eng* 300:510–539. <https://doi.org/10.1016/j.cma.2015.11.015>
- Barrera JL, Geiss MJ, Maute K (2019) Hole seeding in level set topology optimization via density fields. arXiv:1909.10703
- Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. *Comput Methods Appl Mech Eng* 71(2):197–224. [https://doi.org/10.1016/0045-7825\(88\)90086-2](https://doi.org/10.1016/0045-7825(88)90086-2)
- Bennet JA, Botkin ME (1985) Structural shape optimization with geometric description and adaptive mesh refinement. *AIAA J* 23(3):458–464. <https://doi.org/10.2514/3.8935>
- Bruggi M, Verani M (2011) A fully adaptive topology optimization algorithm with goal-oriented error control. *Computers & Structures* 89(15):1481–1493. <https://doi.org/10.1016/j.compstruc.2011.05.003>
- Burman E, Hansbo P (2014) Fictitious domain methods using cut elements: III. A stabilized Nitsche method for Stokes' problem. *ESAIM: Mathematical Modelling and Numerical Analysis* 48(3):859–874. <https://doi.org/10.1051/m2an/2013123>
- Burman E, Elfverson D, Hansbo P, Larson MG, Larsson K (2019) Cut topology optimization for linear elasticity with coupling to parametric nondesign domain regions. *Comput Methods Appl Mech Eng* 350:462–479. <https://doi.org/10.1016/j.cma.2019.03.016>
- Costa Jr JCA, Alves MK (2003) Layout optimization with h-adaptivity of structures. *Int J Numer Methods Eng* 58(1):83–102. <https://doi.org/10.1002/nme.759>
- Cottrell JA, Hughes TJR, Bazilevs Y (2009) *Isogeometric analysis: toward integration of CAD and FEA*. Wiley Publishing, 1st edn. ISBN 0470748737, 9780470748732
- de Boor C (1972) On calculating with B-splines. *Journal of Approximation Theory* 6(1):50–62. [https://doi.org/10.1016/0021-9045\(72\)90080-9](https://doi.org/10.1016/0021-9045(72)90080-9)
- Deaton JD, Grandhi RV (2014) A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Struct Multidiscip Optim* 49(1):1–38. <https://doi.org/10.1007/s00158-013-0956-z>
- DeDè L, Borden MJ, Hughes TJR (2012) Isogeometric analysis for topology optimization with a phase field model. *Archives of Computational Methods in Engineering* 19(3):427–465. <https://doi.org/10.1007/s11831-012-9075-z>
- Eschenauer HA, Kobelev VV, Schumacher A (1994) Bubble method for topology and shape optimization of structures. *Struct Optim* 8(1):42–51. <https://doi.org/10.1007/BF01742933>
- Evans JA, Bazilevs Y, Babuška I, Hughes TJR (2009) n-widths, sup-infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering* 198(21):1726–1741. <https://doi.org/j.cma.2009.01.021>
- Forsey DR, Bartels RH (1988) Hierarchical B-spline refinement. *SIGGRAPH Comput Graph* 22(4):205–212. <https://doi.org/10.1145/378456.378512>
- 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>
- Gee MW, Siefert CM, Hu JJ, Tuminaro RS, Sala MG (2006) ML 5.0 smoothed aggregation user's guide. Technical Report SAND2006-2649 Sandia National Laboratories
- Geiss MJ, Maute K (2018) Topology optimization of active structures using a higher-order level-set-XFEM-density approach. In: 2018 Multidisciplinary Analysis and Optimization Conference. <https://doi.org/10.2514/6.2018-4053>
- Geiss MJ, Barrera JL, Boddetti N, Maute K (2019a) A regularization scheme for explicit level-set XFEM topology optimization. *Frontiers of Mechanical Engineering* 14(2):153–170. <https://doi.org/10.1007/s11465-019-0533-2>
- Geiss MJ, Boddetti N, Weeger O, Maute K, Dunn ML (2019b) Combined level-set-XFEM-density topology optimization of four-dimensional printed structures undergoing large deformation. *J Mech Des* 141(5):051405–051405–14. <https://doi.org/10.1115/1.4041945>
- Giannelli C, Jüttler B, Speleers H (2012) THB-splines: the truncated basis for hierarchical splines. *Computer Aided Geometric Design* 29(7):485–498. <https://doi.org/10.1016/j.cagd.2012.03.025>. *Geometric Modeling and Processing 2012*
- Giannelli C, Jüttler B, Speleers H (2014) Strongly stable bases for adaptively refined multilevel spline spaces. *Adv Comput Math* 40(2):459–490. <https://doi.org/10.1007/s10444-013-9315-2>
- Guest JK, Smith Genut LC (2010) Reducing dimensionality in topology optimization using adaptive design variable fields. *Int J Numer Methods Eng* 81(8):1019–1045. <https://doi.org/10.1002/nme.2724>
- Hansbo A, Hansbo P (2004) A finite element method for the simulation of strong and weak discontinuities in solid mechanics. *Comput Methods Appl Mech Eng* 193(33):3523–3540. <https://doi.org/10.1016/j.cma.2003.12.041>
- Hofreither C, Jüttler B, Kiss G, Zulehner W (2016) Multigrad methods for isogeometric analysis with THB-splines. *Comput Methods Appl Mech Eng* 308:96–112. <https://doi.org/10.1016/j.cma.2016.05.005>
- Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput Methods Appl Mech Eng* 194(39):4135–4195. <https://doi.org/10.1016/j.cma.2004.10.008>
- Hughes TJR, Reali A, Sangalli G (2008) Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: comparison of p-method finite elements with k-method NURBS. *Comput Methods Appl Mech Eng* 197(49):4104–4124. <https://doi.org/10.1016/j.cma.2008.04.006>
- Hughes TJR, Evans JA, Reali A (2014) Finite element and NURBS approximations of eigenvalue, boundary-value, and initial-value problems. *Comput Methods Appl Mech Eng* 272:290–320. <https://doi.org/10.1016/j.cma.2013.11.012>
- Jahangiry HA, Tavakkoli SM (2017) An isogeometrical approach to structural level set topology optimization. *Comput Methods Appl Mech Eng* 319:240–257. <https://doi.org/10.1016/j.cma.2017.02.005>
- Jansen M (2019) Explicit level set and density methods for topology optimization with equivalent minimum length scale constraints. *Struct Multidiscip Optim* 59(5):1775–1788. <https://doi.org/10.1007/s00158-018-2162-5>

- Jensen KE (2016) Anisotropic mesh adaptation and topology optimization in three dimensions. *J Mech Des* 138:061401–1–061401–8. <https://doi.org/10.1115/1.4032266>
- Kang Z, Wang Y (2013) Integrated topology optimization with embedded movable holes based on combined description by material density and level sets. *Comput Methods Appl Mech Eng* 255:1–13. <https://doi.org/10.1016/j.cma.2012.11.006>
- Kikuchi N, Kyoon YC, Torigaki T, Taylor JE (1986) Adaptive finite element methods for shape optimization of linearly elastic structures. *Comput Methods Appl Mech Eng* 57(1):67–89. [https://doi.org/10.1016/0045-7825\(86\)90071-X](https://doi.org/10.1016/0045-7825(86)90071-X)
- Kourounis D, Fuchs A, Schenk O (2018) Towards the next generation of multiperiod optimal power flow solvers. *IEEE Transactions on Power Systems* 33(4):4005–4014. <https://doi.org/10.1109/TPWRS.2017.2789187>
- Kreissl S, Maute K (2012) Level set based fluid topology optimization using the extended finite element method. *Struct Multidiscip Optim* 46(3):311–326. <https://doi.org/10.1007/s00158-012-0782-8>
- Krysl P, Grinspun E, Schröder P (2003) Natural hierarchical refinement for finite element methods. *Int J Numer Methods Eng* 56(8):1109–1124. <https://doi.org/10.1002/nme.601>
- Lieu QX, Lee J (2017) A multi-resolution approach for multi-material topology optimization based on isogeometric analysis. *Comput Methods Appl Mech Eng* 323:272–302. <https://doi.org/10.1016/j.cma.2017.05.009>
- Makhija D, Maute K (2014) Numerical instabilities in level set topology optimization with the extended finite element method. *Struct Multidiscip Optim* 49(2):185–197. <https://doi.org/10.1007/s00158-013-0982-x>
- Maute K (2017) Topology optimization, pages 1–34. American Cancer Society. ISBN 9781119176817. <https://doi.org/10.1002/9781119176817.ecm2108>
- Maute K, Ramm E (1995) Adaptive topology optimization. *Struct Opt* 10(2):100–112. <https://doi.org/10.1007/BF01743537>
- Maute K, Schwarz S, Ramm E (1998) Adaptive topology optimization of elastoplastic structures. *Struct Opt* 15(2):81–91. <https://doi.org/10.1007/BF01278493>
- Moës N, Dolbow J, Belytschko T (1999) A finite element method for crack growth without remeshing. *Int J Numer Methods Eng* 46(1):131–150. [https://doi.org/10.1002/\(SICI\)1097-0207\(19990910\)46:1<131::AID-NME726>3.0.CO;2-J](https://doi.org/10.1002/(SICI)1097-0207(19990910)46:1<131::AID-NME726>3.0.CO;2-J)
- Nana A, Cuillère JC, Francois V (2016) Towards adaptive topology optimization. *Adv Eng Softw* 100:290–307. <https://doi.org/10.1016/j.advengsoft.2016.08.005>
- Nguyen-Xuan H (2017) A polytree-based adaptive polygonal finite element method for topology optimization. *Int J Numer Methods Eng* 110(10):972–1000. <https://doi.org/10.1002/nme.5448>
- 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(1):9–15. <https://doi.org/10.1007/BF02995904>
- Novotny AA, Sokołowski J (2013) Topological derivatives in shape optimization. Springer, Berlin. ISBN 978-3-642-35245-4. <https://doi.org/10.1007/978-3-642-35245-4>
- Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J Comput Phys* 79(1):12–49. [https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2)
- Panesar A, Brackett D, Ashcroft I, Wildman R, Hague R (2017) Hierarchical remeshing strategies with mesh mapping for topology optimisation. *Int J Numer Methods Eng* 111(7):676–700. <https://doi.org/10.1002/nme.5488>
- Qian X (2013) Topology optimization in B-spline space. *Comput Methods Appl Mech Eng* 265:15–35. <https://doi.org/10.1016/j.cma.2013.06.001>
- Salazar de Troya MA, Tortorelli DA (2018) Adaptive mesh refinement in stress-constrained topology optimization. *Struct Multidiscip Optim* 58(6):2369–2386. <https://doi.org/10.1007/s00158-018-2084-2>
- 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. *Computer Methods in Applied Mechanics and Engineering* 249–252:116–150. <https://doi.org/10.1016/j.cma.2012.03.017>. Higher order finite element and isogeometric methods
- Schleupen A, Maute K, Ramm E (2000) Adaptive FE-procedures in shape optimization. *Struct Multidiscip Optim* 19(4):282–302. <https://doi.org/10.1007/s001580050125>
- Sharma A, Villanueva H, Maute K (2017) On shape sensitivities with Heaviside-enriched XFEM. *Struct Multidiscip Optim* 55(2):385–408. <https://doi.org/10.1007/s00158-016-1640-x>
- Sigmund O, Maute K (2013) Topology optimization approaches—a comparative review. *Struct Multidiscip Optim* 48(6):1031–1055. <https://doi.org/10.1007/s00158-013-0978-6>
- Stainko R (2006) An adaptive multilevel approach to the minimal compliance problem in topology optimization. *Commun Numer Methods Eng* 22(2):109–118. <https://doi.org/10.1002/cnm.800>
- Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM J Optim* 12(2):555–572. [https://doi.org/10.1016/0021-9045\(72\)90080-9](https://doi.org/10.1016/0021-9045(72)90080-9)
- Terada K, Asai M, Yamagishi M (2003) Finite cover method for linear and non-linear analyses of heterogeneous solids. *Int J Numer Methods Eng* 58(9):1321–1346. <https://doi.org/10.1002/nme.820>
- van Dijk NP, Maute K, Langelaar M, van Keulen F (2013) Level-set methods for structural topology optimization: a review. *Struct Multidiscip Optim* 48(3):437–472. <https://doi.org/10.1007/s00158-013-0912-y>
- Villanueva CH, Maute K (2014) Density and level set-XFEM schemes for topology optimization of 3-d structures. *Comput Mech* 54(1):133–150. <https://doi.org/10.1007/s00466-014-1027-z>
- Villanueva CH, Maute K (2017) CutFEM topology optimization of 3D laminar incompressible flow problems. *Comput Methods Appl Mech Eng* 320:444–473. <https://doi.org/10.1016/j.cma.2017.03.007>
- Wallin M, Ristinmaa M, Askfelt H (2012) Optimal topologies derived from a phase-field method. *Struct Multidiscip Optim* 45(2):171–183. <https://doi.org/10.1007/s00158-011-0688-x>
- Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. *Comput Methods Appl Mech Eng* 192(1):227–246. [https://doi.org/10.1016/S0045-7825\(02\)00559-5](https://doi.org/10.1016/S0045-7825(02)00559-5)
- Wang MY, Zong H, Ma Q, Tian Y, Zhou M (2019) Cellular level set in B-splines (CLIBS): a method for modeling and topology optimization of cellular structures. *Comput Methods Appl Mech Eng* 349:378–404. <https://doi.org/10.1016/j.cma.2019.02.026>
- Wang S, de Sturler E, Paulino GH (2010) Dynamic adaptive mesh refinement for topology optimization. *arXiv:1009.4975*
- Wang Y, Benson DJ (2016) Isogeometric analysis for parameterized LSM-based structural topology optimization. *Comput Mech* 57(1):19–35. <https://doi.org/10.1007/s00466-015-1219-1>
- Wang Y, Xu H, Pasini D (2017) Multiscale isogeometric topology optimization for lattice materials. *Comput Methods Appl Mech Eng* 316:568–585. <https://doi.org/10.1016/j.cma.2016.08.015>
- Wang Y, Wang Z, Xia Z, Poh LH (2018) Structural design optimization using isogeometric analysis: a comprehensive review.

- Computer Modeling in Engineering & Sciences 117(3):455–507. <https://doi.org/10.31614/cmcs.2018.04603>
- Wang Y-Q, He J-J, Luo Z, Kang Z (2013) An adaptive method for high-resolution topology design. *Acta Mech Sinica* 29(6):840–850. <https://doi.org/10.1007/s10409-013-0084-4>
- Wang Y-Q, Kang Z, He Q (2014) Adaptive topology optimization with independent error control for separated displacement and density fields. *Computers & Structures* 135:50–61. <https://doi.org/10.1016/j.compstruc.2014.01.008>
- Woźniak M, Kuźnik K, Paszyński M, Calo VM, Pardo D (2014) Computational cost estimates for parallel shared memory isogeometric multi-frontal solvers. *Computers & Mathematics with Applications* 67(10):1864–1883. <https://doi.org/10.1016/j.camwa.2014.03.017>
- Xie X, Wang S, Wang Y, Jiang N, Zhao W, Xu M (2020) Truncated hierarchical B-spline-based topology optimization. *Struct Multidiscip Optim* 62:83–105. <https://doi.org/10.1007/s00158-019-02476-4>
- Yserentant H (1986) On the multi-level splitting of finite element spaces. *Numer Math* 49(4):379–412. <https://doi.org/10.1007/BF01389538>
- Zhang W, Li D, Zhou J, Du Z, Li B, Guo X (2018) A moving morphable void (MMV)-based explicit approach for topology optimization considering stress constraints. *Comput Methods Appl Mech Eng* 334:381–413. <https://doi.org/10.1016/j.cma.2018.01.050>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.