Johann Radon Institute for
Computational and Applied Mathematics
Austrian Academy of Sciences (ÖAW)

Coupling Adaptively Refined Multi-patch Spline Discretizations via Boundary Compatibility

B. Jüttler, S.K. Kleiss

RICAM-Report 2016-37

Coupling Adaptively Refined Multi-patch Spline Discretizations via Boundary Compatibility

Bert Jüttler[a], Stefan K. Kleiss[b,*]

[a] *Institute of Applied Geometry, Johannes Kepler University, Linz, Austria*
[b] *Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences*

Abstract

The present paper studies adaptive refinement on multi-patch structures via the gluing approach. More precisely, we investigate the applicability of the gluing construction in isogeometric analysis on multi-patch domains with adaptively refined spline spaces. We will see that this is closely related to the concept of boundary compatibility of an adaptive spline construction.

Given a spline basis (or, more generally, a generating system if linear independence is not guaranteed) on a d-dimensional box domain, there are two possibilities for constructing the spline basis on the domain boundary. Firstly, one can simply restrict the basis functions to the boundary. Secondly, one may restrict the underlying mesh to the boundary and construct the spline basis on the resulting mesh. The two constructions do not necessarily produce the same set of functions. If they do, then the spline bases are said to be compatible.

We study this property for hierarchical (HB-) and truncated hierarchical B-splines (THB-splines) and identify sufficient conditions. These conditions are weaker for THB- than for HB-splines. Finally we demonstrate the importance of boundary compatibility for geometric modeling and for adaptive refinement in isogeometric analysis, in particular when considering multi-patch domains.

Keywords: isogeometric analysis, adaptive refinement, THB-splines, boundary compatibility, multi-patch discretizations

1. Introduction

The framework of Isogeometric Analysis (IGA) [20, 41] aims at closing the gap between geometry representations in Computer-Aided Design (CAD) and classical Finite Element Methods (FEM). The spaces of spline functions that define the geometry simultaneously provide the basis for the discretization. Hence, data transformations becomes unneccessary and the advantageous properties of spline functions, such as the built-in

*Corresponding author, RICAM, Austrian Academy of Sciences, c/o JKU Linz, Altenbergerstr. 69, 4040 Linz, Austria

Email address: stefan.kleiss@ricam.oeaw.ac.at (Stefan K. Kleiss)

smoothness and the various options for refinement (h-, p- or k-refinement, see [9, 20, 41]) can be exploited in the numerical computation. The power of this approach is reflected by the growing number of publications investigating its application in various fields, where isogeometric methods have been shown to be a valuable alternative to the classical, well-established finite-element techniques, see, e.g., [4, 6, 7, 22, 63, 69] and the recent survey [11] as well as the references therein.

In its most basic form, IGA relies on *tensor-product B-splines* which preclude the possibility of local refinement. Any refinement, i.e., the insertion of a knot, affects the entire parameter domain and thus introduces superfluous degrees of freedom. Adaptive numerical methods, which iteratively improve accuracy of the numerical solution while keeping the number of degrees of freedom as small as possible, require local refinement schemes. This observation has triggered the exploration of numerous generalizations of the tensor-product framework.

T-splines, introduced in [59, 60], are based on local knot vectors and allow meshes with T-junctions. They were first applied in [5, 25] and analyzed and further developed in [8, 10, 50, 51, 58]. Additional constraints are required to ensure good mathematical properties, such as linear independence and the partition of unity property [57].

The concept of hierarchical B-splines (HB-splines) was presented in [28, 46] and studied the isogeometric framework in [64]. Hierarchical B-splines, however, do not preserve the partition of unity-property of tensor-product B-splines. This property can be restored by normalizing the hierarchical splines, see [62], or by applying a truncation procedure which leads to Truncated Hierarchical B-splines (THB-splines). THB-splines were introduced in [31] and further explored in [29, 30, 32, 35, 44]. Their extension to Catmull-Clark surfaces has been presented recently [66].

Further spline constructions that allow local refinement include polynomial splines over hierarchical T-meshes (PHT-splines), see [23, 49, 54, 65, 67], and locally refined splines (LR-splines), see [14, 24, 42, 61]. See also [43, 48].

Typically, the constructions of d-variate splines assume that the domain is a d-dimensional box in $\mathbb{R}^d$. The resulting spline spaces, however, are not suitable for representing more complicated physical domains. As a natural solution to this problem, multi-patch representations have been employed in IGA. Multi-patch structures can also be the result of domain segmentation methods which split large objects into smaller topological hexahedra or quadrilaterals that admit spline representations, see [16, 52, 55]. Apart from geometric aspects, multi-patch representations are also employed when different materials or models are used in different areas of the considered object, see, e.g., [21, 38].

Many techniques are available for performing numerical computation on multi-patch structures in IGA. Methods which have been derived from classical, overlapping and non-overlapping domain decomposition methods were studied in, e.g., [12, 13, 15, 36, 37, 38, 45]. A Nitsche-type formulation was discussed in [3], and the discontinuous Galerkin methods studied in [39, 40, 47] allow structures which may have small gaps and/or overlaps between patches, i.e., which are not geometrically conforming.

If the interfaces of the multi-patch structure are suitable and the global system has

a reasonable size, it is also possible to use a simple all-at-once-approach, where a global discretization is constructed by *gluing* local basis functions. In this approach, any two basis functions from different patches, which are non-vanishing and take identical values on an interface, are identified with the same degree of freedom. This can be interpreted as constructing one global basis function which is defined patch-wise and which is globally continuous. Extensions of this gluing approach cover situations where one side of an interface is the refinement of the other one (similar to hanging nodes in classical finite elements), or where higher smoothness is enforced via geometric continuity, see [17, 19, 21].

The present paper studies *adaptive refinement* on *multi-patch structures* via the *gluing approach*. More precisely, we investigate the applicability of the gluing construction in the framework of adaptive spline refinement on multi-patch domains, which is discussed in Section 5. We will see that this is closely related to the concept of *boundary compatibility* of an adaptive spline construction.

Consider a given construction of (adaptively refined) spline spaces, which is applied to d-dimensional domain. There are two possibilities for constructing a spline basis on the domain boundary. Firstly, one can restrict the basis functions to the boundary, which corresponds to a simple evaluation of the "full", d-variate splines, at the domain boundary. Secondly, the mesh can be restricted to the domain boundary and new splines can be constructed from the restricted mesh. If these two constructions produce the same set of functions, we say that the basis possesses the property of *boundary compatibility*.

We discuss the property of boundary compatibility of generating systems from an abstract point of view in Section 2, where we also show that tensor-product B-splines possess this property. We present conditions which are sufficient for boundary compatibility of HB-splines and THB-splines in Sections 3 and 4, respectively. Section 5 is dedicated to multi-patch structures, and based on the previous theoretical findings we formulate conditions which ensure that the patch-wise defined bases of a multi-patch THB-/HB-spline structure can be coupled by invoking a gluing construction. These conditions, however, may be violated in the course of adaptive refinement, and we present an algorithm that restores them in Section 6.

2. Boundary compatibility

We introduce the notion of boundary compatibility of spline spaces and discuss it for tensor-product splines and adaptively refined spline constructions.

2.1. Compatibility property

Given a parameter domain, which is often assumed to be the d-dimensional unit cube

$$\Omega = [0,1]^d,$$

spline spaces in isogeometric analysis and geometric modeling are typically constructed by specifying a finite basis $\mathcal{X}$ (or, more generally, a generating system if linear independence is not guaranteed) that spans a finite-dimensional subspace of $C(\Omega)$. More

general axis-aligned boxes can be transformed into the unit cube by a simple affine transformation.

We will consider $\mathcal{X}$ as a *vector* of functions, i.e., $\mathcal{X} \in C(\Omega)^{|\mathcal{X}|}$, where $|\mathcal{X}|$ denotes the number of functions in $\mathcal{X}$. By a slight abuse of notation, we will write $\varphi \in \mathcal{X}$ if one of the elements of the vector is equal to the function φ.

Clearly, the boundary of the d-dimensional unit cube Ω consists of $2d$ sides, which are $(d-1)$-dimensional boxes. Without loss of generality we consider the side

$$\Sigma = \{(0, \xi_2, \ldots, \xi_d),\ \xi_i \in [0,1] \text{ for } i = 2, \ldots, d\} \subset \overline{\Omega},$$

which is obtained by intersecting the boundary of the cube with the plane $\xi_1 = 0$. In order to keep the notation simple we will identify Σ with the unit cube $[0,1]^{d-1}$, even though it is simultaneously considered as a subset of $\overline{\Omega} = [0,1]^d$.

Typically, the spline basis $\mathcal{X}$ on Ω is defined by specifying a polynomial degree p and by choosing a mesh $\mathcal{M}$ (which is often a subdivision of the domain into axis-aligned boxes, possibly with different multiplicities for the interfaces). On the one hand, restricting the spline basis on Ω to the side Σ gives the generating system $\mathcal{X}|_\Sigma$ that spans the restricted spline space. On the other hand, the restriction $\mathcal{M}|_\Sigma$ of the mesh to the side Σ results in another mesh, which defines another spline basis $\mathcal{Y}$ on the side. The generating system $\mathcal{Y}$ is represented by another vector of functions, i.e., $\mathcal{Y} \in C(\Sigma)^{|\mathcal{Y}|}$.

We study the compatibility of these two spline constructions. More precisely, it is desirable to obtain (essentially) a one-to-one correspondence between the non-zero functions in $\mathcal{X}|_\Sigma$ and $\mathcal{Y}$.

Given two functions $\varphi \in \mathcal{X}$ and $\psi \in \mathcal{Y}$ satisfying

$$\varphi|_\Sigma = \psi,$$

we say that φ is an *extension* of ψ, and ψ is the *restriction* of φ. The compatibility of the spline constructions of the domain is formalized by introducing the notion of compatibility.

Definition 1. Consider two generating systems $\mathcal{X} \in C(\Omega)^{|\mathcal{X}|}$ and $\mathcal{Y} \in C(\Sigma)^{|\mathcal{Y}|}$ that are defined on the unit cube Ω and on its side Σ, respectively. These two systems are said to be *compatible*, if both the *unique restriction condition*

$$\forall \varphi \in \mathcal{X}: \quad \varphi|_\Sigma \neq 0 \ \Rightarrow\ \exists! \psi \in \mathcal{Y}:\ \varphi|_\Sigma = \psi \tag{1}$$

and the *unique extension condition*

$$\forall \psi \in \mathcal{Y}: \quad \psi \neq 0 \ \Rightarrow\ \exists! \varphi \in \mathcal{X}:\ \varphi|_\Sigma = \psi \tag{2}$$

are satisfied.

Given a vector of functions $\mathcal{V}$, we denote with $\mathcal{V}_0$ the sub-vector which is formed by its non-zero elements. We consider the situation of the previous definition and provide a characterization of compatible generating systems.

Lemma 2. *The two generating systems $\mathcal{X} \in C(\Omega)^{|\mathcal{X}|}$ and $\mathcal{Y} \in C(\Sigma)^{|\mathcal{Y}|}$ are compatible if and only if*

(i) there are no repeated entries in $\mathcal{Y}_0$ and

(ii) there exists a permutation matrix P such that

$$(\mathcal{X}|_\Sigma)_0 = P\mathcal{Y}_0.$$

The second condition is equivalent to the fact that the two vectors are related by a bijection that identifies elements taking identical values on the side.

Proof. One the one hand, if the two generating systems are compatible, the unique restriction condition implies that there are no repeated entries in $\mathcal{Y}_0$. The unique extension condition thus establishes the existence of a bijection between $\mathcal{Y}_0$ and a subset of $(\mathcal{X}|_\Sigma)_0$. The latter subset is the entire set $(\mathcal{X}|_\Sigma)_0$, due to the unique restriction condition. Consequently we have a bijection between $(\mathcal{X}|_\Sigma)_0$ and $\mathcal{Y}_0$, which can expressed by a permutation.

On the other hand, the two conditions (i) and (ii) imply both the unique restriction and the unique extension condition. The existence of the restrictions and extensions is obvious, and their uniqueness follows from (i). □

We will say that a construction of spline bases possesses the property of *boundary compatibility* if it produces generating systems that are compatible in the sense of Definition 1.

2.2. Tensor-product splines

Tensor-product B-splines are the simplest instance of a boundary compatible spline construction on the domain Ω and its side Σ.

On Ω, these splines are obtained by choosing d degrees and d knot vectors (one per coordinate direction). The Cartesian product of the knot vectors defines the mesh $\mathcal{M}$. The mesh decomposes the domain Ω into elements (cells), and the set of elements will be denoted by $\mathcal{L}(\Omega)$.

Each knot vector has an associated set of univariate B-splines. Their products are the tensor-product B-splines. More precisely, we obtain tensor-product B-splines

$$\mathcal{B} = (\beta_{\boldsymbol{i}})_{\boldsymbol{i}\in\mathcal{I}},$$

with the index set

$$\mathcal{I} = \bigtimes_{k=1}^{d} \{1, \dots, n^k\},$$

where n^k denotes the number of B-splines in the k-th coordinate. These functions are products

$$\beta_{\boldsymbol{i}}(\xi_1, \xi_2, \dots, \xi_d) = \prod_{k=1}^{d} b_{k,i_k}(\xi_k), \quad \boldsymbol{i} = (i_1, \dots, i_d) \in \mathcal{I}$$

of d univariate B-splines b_{k,i_k}.

The tensor-product splines on Σ use the same knot vectors except for the first one, which is omitted. The construction of the tensor-product B-splines on the side is analogous to the full-dimensional case.

In detail, we obtain tensor-product B-splines

$$\mathcal{C} = (\gamma_{\boldsymbol{j}})_{\boldsymbol{j}\in\mathcal{J}},$$

with the index set

$$\mathcal{J} = \bigtimes_{k=2}^{d} \{1, \dots, n^k\}.$$

These are functions are products

$$\gamma_{\boldsymbol{j}}(\xi_2, \dots, \xi_d) = \prod_{k=2}^{d} b_{k,j_k}(\xi_k), \quad \boldsymbol{j} = (j_2, \dots, j_d) \in \mathcal{J}$$

of $d-1$ univariate B-splines b_{k,j_k}.

It is rather obvious that tensor-product splines possess the compatibility property under natural assumptions. The tensor-product spline bases $\mathcal{B}$ and $\mathcal{C}$ form the vectors $\mathcal{X}$ and $\mathcal{Y}$ in this situation.

Proposition 3. *The tensor-product B-splines on Ω and Σ are compatible if the boundary knots 0 and 1 have multiplicity $p+1$ in all univariate knot vectors.*

Proof. The existence of the restriction and extension follows directly from the tensor-product structure of the involved bases. The uniqueness is implied by the linear independence of tensor-product splines. □

Note that the conditions of the Proposition imply compatibility with respect to all $2d$ sides of the cube Ω. Also note that the indices of functions in $\mathcal{B}$ which do not vanish on Σ are given by

$$\mathcal{I}_\Sigma = \{\boldsymbol{i} \in \mathcal{I} : \ \beta_{\boldsymbol{i}}|_\Sigma \neq 0\} = \{(1, \boldsymbol{j}) : \ \boldsymbol{j} \in \mathcal{J}\}, \tag{3}$$

and that the extension of a B-spline $\gamma_{\boldsymbol{j}}$ is given by $\beta_{(1,\boldsymbol{j})}$, i.e.,

$$\beta_{(1,\boldsymbol{j})}|_\Sigma = \gamma_{\boldsymbol{j}}, \quad \forall \boldsymbol{j} \in \mathcal{J}. \tag{4}$$

2.3. Spline constructions providing local adaptivity

As mentioned in the introduction, spline constructions that allow local refinement include T-splines, LR-splines, HB-splines, THB-splines, and PHT-splines.

- T-splines possess the property of boundary compatibility. Only T-splines whose local knot vector includes the boundary knot with multiplicity $p+1$ do not vanish at the domain boundary. The basis functions at the domain boundary are thus determined only by the mesh on the respective side.

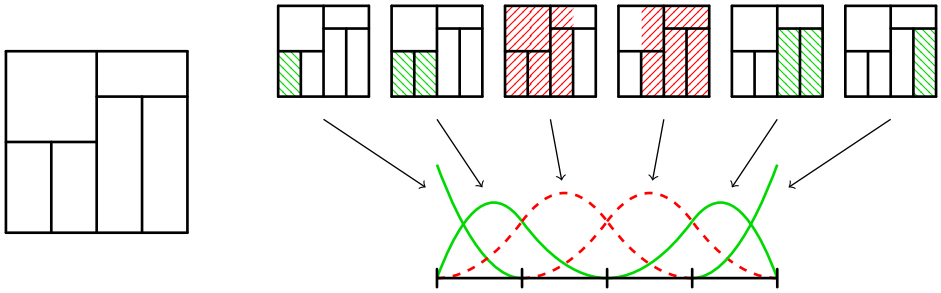


Figure 1: Counterexample for boundary compatibility of (bi-quadratic) LR-splines. The dashed univariate splines do not have extensions in the two-variate basis.

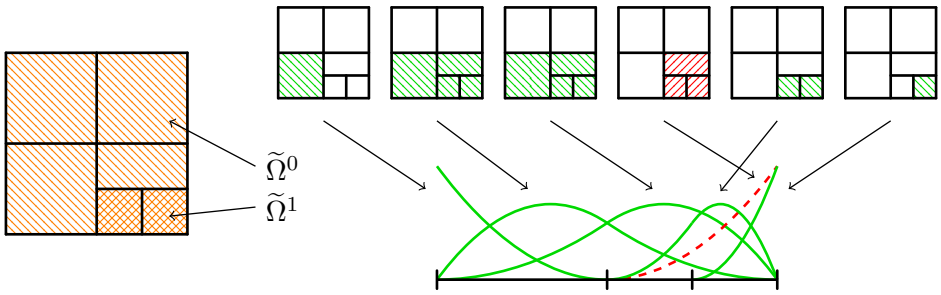


Figure 2: Counterexample for boundary compatibility of (bi-quadratic) HB-splines. The dashed spline is a restriction of a two-dimensional function, but it is not a univariate HB-spline (see Section 3 for details).

- We note that LR-splines do not possess the property of boundary compatibility in general. A counterexample is illustrated in Figure 1. The restriction of the two-dimensional mesh (shown on the left) to its southern boundary results in the knot vector displayed on the right. Among the six univariate basis functions on this knot vector, the two functions that are plotted with dashed, red lines, do not have a bivariate extension.

- Also, HB-splines and THB-splines are not boundary compatible in general. Figure 2 shows an example of a hierarchical mesh (see Section 3 for definition and details) which defines non-boundary compatible HB-splines. The function plotted with a dashed, red line is a restriction of a two-dimensional spline, but it is not a univariate HB-spline.

- The boundary compatibility of PHT-splines depends on the specific spline space, but it is clearly satisfied for $\mathbb{S}(3,3,1,1)$ and $\mathbb{S}(3,3,3,1,1,1)$, the space of piecewise bicubic, globally C^1-continuous polynomials.

Sufficient conditions ensuring the compatibility of hierarchical splines will be derived in the next two sections.

3. Hierarchical B-splines

We recall the framework of hierarchical spline refinement and derive conditions that guarantee the property of boundary compatibility.

3.1. Hierarchies of tensor-product splines

We will use the hierarchical approach to adaptive refinement, which is based on a nested sequence of spline spaces. More precisely, we consider a sequence of levels $\ell = 0, \dots, N$. For each level we choose d polynomial degrees and d knot vectors (one per coordinate direction). The Cartesian product of the knot vectors defines the mesh $\mathcal{M}^\ell$, which subdivides the domain Ω into a set of elements $\mathcal{L}^\ell(\Omega)$. The hierarchy of these meshes is denoted by $\widehat{\mathcal{M}} = (\mathcal{M}^\ell)_{\ell=0,\dots,N}$.

The degrees and knot vectors define tensor-product B-splines

$$\mathcal{B}^\ell = (\beta^\ell_{\boldsymbol{i}})_{\boldsymbol{i}\in\mathcal{I}^\ell},$$

with index sets $\mathcal{I}^\ell$. The sequence of spline bases will be denoted as

$$\widehat{\mathcal{B}} = (\mathcal{B}^\ell)_{\ell=0,\dots,N}.$$

We assume that the knots and degrees are chosen such that the B-splines span nested spline spaces,

$$\operatorname{span}\mathcal{B}^0 \subset \operatorname{span}\mathcal{B}^1 \subset \ldots \subset \operatorname{span}\mathcal{B}^N. \tag{5}$$

This is the case if and only if the degrees are non-decreasing, the knots of a given level are present at all higher levels also, and their multiplicity increases at least by the difference of the degrees.

For each level ℓ, the tensor-product splines on the side Σ use the same knot vectors except for the first one, which is omitted, and the construction of the boundary tensor-product B-splines is again analogous to the full-dimensional case. We obtain another sequence of tensor-product B-splines

$$\widehat{\mathcal{C}} = (\mathcal{C}^\ell)_{\ell=0,\dots,N}, \quad \mathcal{C}^\ell = (\gamma^\ell_{\boldsymbol{j}})_{\boldsymbol{j}\in\mathcal{J}^\ell},$$

with index sets $\mathcal{J}^\ell$. If the full hierarchy consists of nested spaces (5), then this property is inherited by the tensor-product splines on Σ,

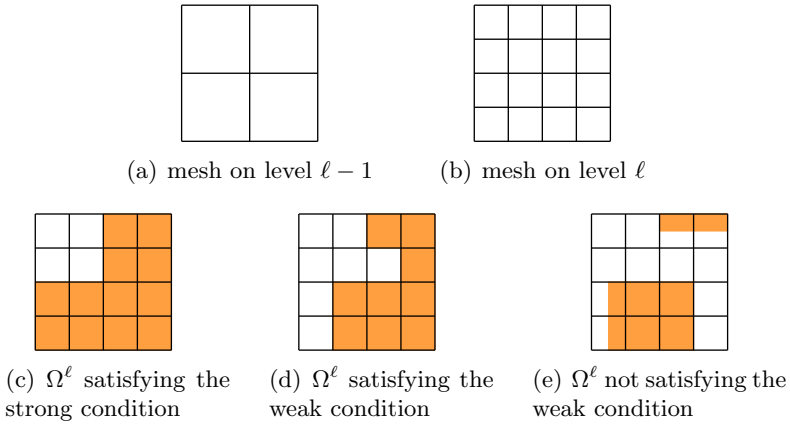
$$\operatorname{span}\mathcal{C}^0 \subset \operatorname{span}\mathcal{C}^1 \subset \ldots \subset \operatorname{span}\mathcal{C}^N.$$

Finally we note that Proposition 3 ensures the compatibility of the spline bases $\mathcal{B}^\ell$ and $\mathcal{C}^\ell$ of each level. In particular we have index sets

$$\mathcal{I}^\ell_\Sigma = \{\boldsymbol{i}\in\mathcal{I}^\ell : \ \beta^\ell_{\boldsymbol{i}}|_\Sigma \neq 0\} = \{(1,\boldsymbol{j}) : \ \boldsymbol{j}\in\mathcal{J}^\ell\}, \tag{6}$$

and the extension of a B-spline $\gamma^\ell_{\boldsymbol{j}}$ is given by $\beta^\ell_{(1,\boldsymbol{j})}$, i.e.,

$$\beta^\ell_{(1,\boldsymbol{j})}|_\Sigma = \gamma^\ell_{\boldsymbol{j}}, \quad \forall \boldsymbol{j}\in\mathcal{J}^\ell. \tag{7}$$



(a) mesh on level $\ell - 1$ (b) mesh on level ℓ

(c) Ω^ℓ satisfying the strong condition (d) Ω^ℓ satisfying the weak condition (e) Ω^ℓ not satisfying the weak condition

Figure 3: Different choices of the subdomain Ω^ℓ illustrate the weak/strong condition, see Definition 4.

3.2. Subdomain hierarchies

For each level ℓ we consider a subdomain Ω^ℓ, which is assumed to be the closure of an open subset of $[0,1]^d$. In addition it is required that these subdomains form an inversely nested sequence

$$\Omega = \Omega^0 \supseteq \Omega^1 \supseteq \ldots \supseteq \Omega^N.$$

This finite sequence of subdomains is denoted by

$$\widehat{\Omega} = (\Omega^\ell)_{\ell=0,\ldots,N}. \tag{8}$$

To unify the notation, we define $\Omega^\ell = \emptyset$ for $\ell > N$.

In principle it is possible to consider an arbitrary sequence of nested subdomains. We restrict ourselves to subdomains that are obtained by collecting elements. More precisely, each subdomain corresponds to a set of elements $\mathcal{E}^\ell$ via

$$\Omega^\ell = \overline{\bigcup_{e \in \mathcal{E}^\ell} e}.$$

The following two notions, which characterize two important special cases, have been introduced in [64]:

Definition 4. The subdomain hierarchy $\widehat{\Omega}$ fulfills the *weak condition*, if each Ω^ℓ is composed of elements which are defined by the mesh $\mathcal{M}^\ell$ of the *same* level, i.e., $\mathcal{E}^\ell \subseteq \mathcal{L}^\ell$, $\ell = 1, \ldots, N$. They are said to satisfy the *strong condition* if the sets $\mathcal{E}^\ell$ consist of elements of the *previous* level $\ell - 1$, i.e., $\mathcal{E}^\ell \subseteq \mathcal{L}^{\ell-1}$.

Clearly the strong condition implies the weak one. An illustration is provided in Figure 3, where a hierarchical mesh is shown with 2×2 cells on level $\ell - 1$ and 4×4 cells on level ℓ (see Figures 3(a) and 3(b)). The three choices of the subdomain Ω^ℓ depicted in Figures 3(c), 3(d), and 3(e) satisfy the strong (and thus also the weak condition), only the weak condition, and neither condition, respectively.

The intersections of the side Σ with the subdomains Ω^ℓ define the associated *side subdomains*

$$\Sigma^\ell = \Sigma \cap \Omega^\ell.$$

These intersections define another inversely nested sequence,

$$\Sigma = \Sigma^0 \supseteq \Sigma^1 \supseteq \ldots \supseteq \Sigma^N.$$

This finite sequence of subdomains is denoted by

$$\widehat{\Sigma} = (\Sigma^\ell)_{\ell=0,\ldots,N}. \tag{9}$$

If the subdomains Ω^ℓ satisfy the weak or the strong condition then this is inherited by the associated side subdomains, respectively.

We conclude this section with a simple observation about supports of B-splines:

Lemma 5. *We consider a B-spline $\gamma_{\boldsymbol{j}}^\ell \in \mathcal{C}^\ell$ and its unique extension $\beta_{(1,\boldsymbol{j})}^\ell \in \mathcal{B}^\ell$.*

(i) Assume that Ω^ℓ satisfies the weak condition. The support of the extension $\beta_{(1,\boldsymbol{j})}^\ell$ is contained in the subdomain Ω^ℓ of the same level if and only if the support of $\gamma_{\boldsymbol{j}}^\ell$ is contained in the associated side subdomain Σ^ℓ.

(ii) Assume that Ω^ℓ satisfies the strong condition. The support of the extension $\beta_{(1,\boldsymbol{j})}^\ell$ is contained in the subdomain $\Omega^{\ell+1}$ of the next finer level if and only if the support of $\gamma_{\boldsymbol{j}}^\ell$ is contained in the associated side subdomain $\Sigma^{\ell+1}$.

Proof. (i) The implication $\operatorname{supp}\gamma_{\boldsymbol{j}}^\ell \not\subseteq \Sigma^\ell \Rightarrow \operatorname{supp}\beta_{(1,\boldsymbol{j})}^\ell \not\subseteq \Omega^\ell$ is trivial. For the implication $\operatorname{supp}\gamma_{\boldsymbol{j}}^\ell \subseteq \Sigma^\ell \Rightarrow \operatorname{supp}\beta_{(1,\boldsymbol{j})}^\ell \subseteq \Omega^\ell$, see Figure 4 for an illustration. The domain $\Omega^\ell \subset \Omega$ is indicated by the colored area on the right in Figure 4, and the domain $\Sigma^\ell = \Omega^\ell \cap \Sigma$ by the colored area on the left. The red line on the left marks the support of a basis function $\gamma_{\boldsymbol{j}}^\ell$.

Since Ω^ℓ consists of non-empty cells of level ℓ, it is always possible to extend Σ^ℓ into Ω^ℓ by a one-cell-layer, which is in indicated by the area shaded red on the right in Figure 4. The extension of $\gamma_{\boldsymbol{j}}^\ell$ is given by $\beta_{(1,\boldsymbol{j})}^\ell$ (see (7)), and due to the tensor-product structure of the B-splines, the support of $\beta_{\boldsymbol{i}}^\ell$ is

$$\operatorname{supp}\beta_{\boldsymbol{i}}^\ell = \operatorname{supp} b_{1,1}^\ell \times \operatorname{supp}\gamma_{\boldsymbol{j}}^\ell.$$

Recall that the B-splines are defined on open knot vectors, i.e., the support of the univariate basis function $b_{1,1}^\ell$ is contained in the first non-empty knot span of the first coordinate direction. Thus, $\operatorname{supp}\beta_{(1,\boldsymbol{j})}^\ell$, which is indicated by the red rectangle on the right in Figure 4, is always contained in the one-cell-layer and therefore also in Ω^ℓ.

The second statement (ii) follows with the same arguments as the first one, using the fact that $\Omega^{\ell+1}$ consists of non-empty cells of level ℓ due to the strong condition. □

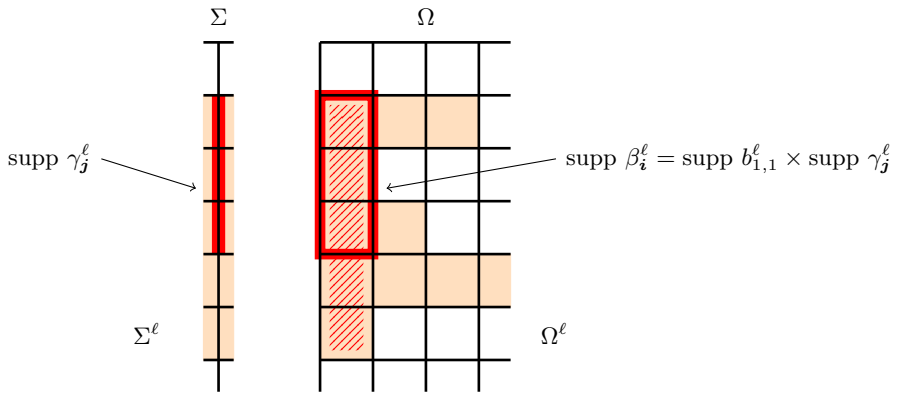


Figure 4: The support of a B-spline $\gamma_{\boldsymbol{j}}^{\ell}$ (left) and of its extension (right). The subdomains Σ^{ℓ} and Ω^{ℓ} are shaded orange.

3.3. Boundary Compatibility

We will now use the results from the previous sections to discuss compatibility of HB-splines. Recall that Figure 2 shows an example where a subdomain does not satisfy the strong condition and where the resulting HB-splines are not boundary compatible. In this section, we will show that the strong condition is sufficient for compatibility of HB-splines.

On each level ℓ, we select the basis functions from $\mathcal{B}^{\ell}$, whose support is contained in Ω^{ℓ}, but not in $\Omega^{\ell+1}$,

$$\mathcal{S}^{\ell}(\widehat{\mathcal{B}},\widehat{\Omega}) = (\beta_{\boldsymbol{i}}^{\ell})_{\boldsymbol{i}\in\mathcal{I}^{\ell},\ \operatorname{supp}\beta_{\boldsymbol{i}}^{\ell}\subseteq\Omega^{\ell}\wedge\operatorname{supp}\beta_{\boldsymbol{i}}^{\ell}\not\subseteq\Omega^{\ell+1}}.$$

Applying the same mechanism to the B-splines on the side defines the selected B-splines on Σ,

$$\mathcal{S}^{\ell}(\widehat{\mathcal{C}},\widehat{\Sigma}) = (\gamma_{\boldsymbol{j}}^{\ell})_{\boldsymbol{j}\in\mathcal{J}^{\ell},\ \operatorname{supp}\gamma_{\boldsymbol{j}}^{\ell}\subseteq\Sigma^{\ell}\wedge\operatorname{supp}\gamma_{\boldsymbol{j}}^{\ell}\not\subseteq\Sigma^{\ell+1}}.$$

These functions will be denoted as the *selected* ones.

Lemma 6. *The selected functions $\mathcal{S}^{\ell}(\widehat{\mathcal{B}},\widehat{\Omega})$ and $\mathcal{S}^{\ell}(\widehat{\mathcal{C}},\widehat{\Sigma})$ are compatible if the domain hierarchy $\widehat{\Omega}$ satisfies the strong condition.*

Proof. This follows immediately by combining Proposition 3 and Lemma 5. It should be noted that both parts of the Lemma apply since the strong condition implies the weak one. □

We define the hierarchical B-splines (HB-splines) by collecting the selected functions from all levels,

$$\mathcal{H}(\widehat{\mathcal{B}},\widehat{\Omega}) = \bigcup_{\ell=0,\dots,N} \mathcal{S}^{\ell}(\widehat{\mathcal{B}},\widehat{\Omega}) \quad \text{and} \quad \mathcal{H}(\widehat{\mathcal{C}},\widehat{\Sigma}) = \bigcup_{\ell=0,\dots,N} \mathcal{S}^{\ell}(\widehat{\mathcal{C}},\widehat{\Sigma}),$$

see [31, 32, 64] and the references therein for further information. Among other useful properties it can be shown that these sets are linearly independent. Consequently we obtain two bases of spline spaces, which are defined on the domain Ω and on its side Σ.

Theorem 7. *The bases $\mathcal{H}(\widehat{\mathcal{B}},\widehat{\Omega})$ and $\mathcal{H}(\widehat{\mathcal{C}},\widehat{\Sigma})$ of HB-splines on Ω and Σ are compatible if the subdomain hierarchy $\widehat{\Omega}$ fulfills the strong condition.*

Proof. The existence of restrictions and extensions follows from the compatibility of the selected functions $\mathcal{S}^\ell(\widehat{\mathcal{B}},\widehat{\Omega})$ and $\mathcal{S}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma})$ within each level.

On the one hand, the restriction is also unique. Indeed, if a function

$$\beta_{\boldsymbol{i}}^\ell \in \mathcal{S}^\ell(\widehat{\mathcal{B}},\widehat{\Omega}),$$

possessed two restrictions

$$\gamma_{\boldsymbol{j}}^\ell \in \mathcal{S}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma}) \quad \text{and} \quad \gamma_{\boldsymbol{j}'}^{\ell'} \in \mathcal{S}^{\ell'}(\widehat{\mathcal{C}},\widehat{\Sigma}),$$

this would contradict the linear indepedence of $\mathcal{H}(\widehat{\mathcal{C}},\widehat{\Sigma})$.

On the other hand, the extension is also unique. Assume there exists a function

$$\gamma_{\boldsymbol{j}}^\ell \in \mathcal{S}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma})$$

with two extensions

$$\beta_{\boldsymbol{i}}^\ell \in \mathcal{S}^\ell(\widehat{\mathcal{B}},\widehat{\Omega}) \quad \text{and} \quad \beta_{\boldsymbol{i}'}^{\ell'} \in \mathcal{S}^{\ell'}(\widehat{\mathcal{B}},\widehat{\Omega}).$$

We then have $\ell \neq \ell'$, due to Lemma 6. These functions possess two restrictions

$$\gamma_{\boldsymbol{j}}^\ell \in \mathcal{S}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma}) \quad \text{and} \quad \gamma_{\boldsymbol{j}'}^{\ell'} \in \mathcal{S}^{\ell'}(\widehat{\mathcal{C}},\widehat{\Sigma})$$

which are identical, in contradiction to the linear indepedence of $\mathcal{H}(\widehat{\mathcal{C}},\widehat{\Sigma})$. □

4. Truncated hierarchical B-splines

We extend the results of the previous section to THB-splines, which are obtained by invoking the truncation mechanism. In addition we present an application to geometric modeling.

4.1. Active B-splines and truncation

In this section we consider the *partial subdomain hierarchies*,

$$\widehat{\Omega}^\ell = (\Omega^k)_{k=0,\dots,\ell} \quad \text{and} \quad \widehat{\Sigma}^\ell = (\Sigma^k)_{k=0,\dots,\ell},$$

which are obtained by considering the subdomains of the first ℓ levels, cf. (8) and (9).

The starting point for the definition of truncated hierarchical B-splines is provided by the *vectors of active B-splines on* Ω,

$$\mathcal{A}^\ell(\widehat{\mathcal{B}},\widehat{\Omega}^\ell) = (\beta_{\boldsymbol{i}}^\ell)_{\boldsymbol{i}\in\mathcal{I}^\ell,\,\operatorname{supp}\beta_{\boldsymbol{i}}^\ell\subseteq\Omega^\ell},$$

which are defined for each level ℓ. These splines are "active" in the sense that they remain in the spline space. Even if the truncation operation, which will be specified

below, truncates a certain function to zero, it can still be represented by the basis functions of higher levels.

Applying the same mechanism to the side defines the *vector of active B-splines on* Σ,

$$\mathcal{A}^\ell(\widehat{\mathcal{C}}, \widehat{\Sigma}^\ell) = (\gamma_{\boldsymbol{j}}^\ell)_{\boldsymbol{j}\in\mathcal{J}^\ell,\, \operatorname{supp}\gamma_{\boldsymbol{j}}^\ell\subseteq\Sigma^\ell}.$$

The active B-splines are compatible under mild assumptions.

Lemma 8. *The vectors of active B-splines* $\mathcal{A}^\ell(\widehat{\mathcal{B}}, \widehat{\Omega}^\ell)$ *and* $\mathcal{A}^\ell(\widehat{\mathcal{C}}, \widehat{\Sigma}^\ell)$ *are compatible if the subdomain hierarchy* $\widehat{\Omega}$ *satisfies the weak condition.*

Proof. This is implied by Proposition 3 and Lemma 5. □

Due to the nestedness of the function sets $\mathcal{B}^\ell$ and $\mathcal{C}^\ell$, the functions

$$\varphi \in \operatorname{span}\mathcal{B}^{\ell-1} \quad \text{and} \quad \psi \in \operatorname{span}\mathcal{C}^{\ell-1}$$

can be represented as linear combinations of functions from the next higher level,

$$\varphi = \sum_{\boldsymbol{i}\in\mathcal{I}^\ell} f_{\boldsymbol{i}}^\ell \beta_{\boldsymbol{i}}^\ell \quad \text{and} \quad \psi = \sum_{\boldsymbol{j}\in\mathcal{J}^\ell} g_{\boldsymbol{j}}^\ell \gamma_{\boldsymbol{j}}^\ell, \quad f_{\boldsymbol{i}}^\ell, g_{\boldsymbol{j}}^\ell \in \mathbb{R}, \tag{10}$$

respectively. Applying *truncation with respect to level* ℓ to these functions means that we omit the contributions of all basis functions whose support is contained in Ω^ℓ and Σ^ℓ, respectively,

$$\operatorname{trunc}^\ell \varphi = \sum_{\substack{\boldsymbol{i}\in\mathcal{I}^\ell,\\ \beta_{\boldsymbol{i}}^\ell\notin\mathcal{A}^\ell(\widehat{\mathcal{B}},\widehat{\Omega}^\ell)}} f_{\boldsymbol{i}}^\ell \beta_{\boldsymbol{i}}^\ell \quad \text{and} \quad \operatorname{trunc}^\ell \psi = \sum_{\substack{\boldsymbol{j}\in\mathcal{J}^\ell,\\ \gamma_{\boldsymbol{j}}^\ell\notin\mathcal{A}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma}^\ell)}} g_{\boldsymbol{j}}^\ell \gamma_{\boldsymbol{j}}^\ell. \tag{11}$$

We use the same symbol for truncation on the domain Ω and on the side Σ.

The truncation with respect to level ℓ keeps the values of a function on $\Omega \setminus \Omega^\ell$ and $\Sigma \setminus \Sigma^\ell$ unchanged,

$$(\operatorname{trunc}^\ell \varphi)|_{\Omega\setminus\Omega^\ell} = \varphi|_{\Omega\setminus\Omega^\ell} \quad \text{and} \quad (\operatorname{trunc}^\ell \psi)|_{\Sigma\setminus\Sigma^\ell} = \psi|_{\Sigma\setminus\Sigma^\ell}. \tag{12}$$

Moreover the truncation with respect to the next level annihilates any active function that is not selected at the current level,

$$\begin{aligned} &\operatorname{trunc}^{\ell+1} \varphi = 0 \quad \text{if} \quad \varphi \in \mathcal{A}^\ell(\widehat{\mathcal{B}}, \widehat{\Omega}^\ell) \setminus \mathcal{S}^\ell(\widehat{\mathcal{B}}, \widehat{\Omega}) \quad \text{and} \\ &\operatorname{trunc}^{\ell+1} \psi = 0 \quad \text{if} \quad \psi \in \mathcal{A}^\ell(\widehat{\mathcal{C}}, \widehat{\Sigma}^\ell) \setminus \mathcal{S}^\ell(\widehat{\mathcal{C}}, \widehat{\Sigma}). \end{aligned} \tag{13}$$

Finally we note that truncation commutes with restriction to the side under mild assumptions:

Lemma 9. *Consider* $\varphi \in \operatorname{span}\mathcal{B}^{\ell-1}$. *Then,*

$$\operatorname{trunc}^\ell(\varphi|_\Sigma) = (\operatorname{trunc}^\ell \varphi)|_\Sigma.$$

if the subdomain hierarchy $\widehat{\Omega}$ *satisfies the weak condition.*

Proof. We consider a function φ as defined in (10). On the one hand, we restrict it to the side and obtain

$$\varphi|_\Sigma = \sum_{\boldsymbol{j}\in\mathcal{J}^\ell} f^\ell_{(1,\boldsymbol{j})}\beta^\ell_{(1,\boldsymbol{j})} = \sum_{\boldsymbol{j}\in\mathcal{J}^\ell} f^\ell_{(1,\boldsymbol{j})}\gamma^\ell_{\boldsymbol{j}},$$

see (6) and (7). Applying the truncation gives

$$\operatorname{trunc}^\ell(\varphi|_\Sigma) = \sum_{\substack{\boldsymbol{j}\in\mathcal{J}^\ell,\\ \gamma^\ell_{\boldsymbol{j}}\notin\mathcal{A}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma}^\ell)}} f^\ell_{(1,\boldsymbol{j})}\gamma^\ell_{\boldsymbol{j}}. \tag{14}$$

On the other hand, restricting the truncated function $\operatorname{trunc}^\ell\varphi$ (see (11)) to the side gives

$$(\operatorname{trunc}^\ell\varphi)|_\Sigma = \sum_{\substack{\boldsymbol{j}\in\mathcal{J}^\ell,\\ \beta^\ell_{(1,\boldsymbol{j})}\notin\mathcal{A}^\ell(\widehat{\mathcal{B}},\widehat{\Omega}^\ell)}} f^\ell_{(1,\boldsymbol{j})}\beta^\ell_{(1,\boldsymbol{j})}, \tag{15}$$

see again (6) and (7). The two functions (14) and (15) are identical as the sums consider the same indices $\boldsymbol{j}$ according to part (i) of Lemma 5. □

4.2. Basis functions

We recursively define the *vectors of truncated B-splines of level* ℓ. We initialize the construction by taking the active functions of level 0,

$$\mathcal{T}^0(\widehat{\mathcal{B}},\widehat{\Omega}^0) = \mathcal{A}^0(\widehat{\mathcal{B}},\widehat{\Omega}^0) \quad\text{and}\quad \mathcal{T}^0(\widehat{\mathcal{C}},\widehat{\Sigma}^0) = \mathcal{A}^0(\widehat{\mathcal{C}},\widehat{\Sigma}^0).$$

In each step, we apply truncation with respect to level ℓ to the truncated functions of level $\ell-1$ and append the vector of active functions of level ℓ,

$$\begin{array}{rcl} \mathcal{T}^\ell(\widehat{\mathcal{B}},\widehat{\Omega}^\ell) &=& (\operatorname{trunc}^\ell\mathcal{T}^{\ell-1}(\widehat{\mathcal{B}},\widehat{\Omega}^{\ell-1}),\mathcal{A}^\ell(\widehat{\mathcal{B}},\widehat{\Omega}^\ell)) \quad\text{and}\\ \mathcal{T}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma}^\ell) &=& (\operatorname{trunc}^\ell\mathcal{T}^{\ell-1}(\widehat{\mathcal{C}},\widehat{\Sigma}^{\ell-1}),\mathcal{A}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma}^\ell)). \end{array} \tag{16}$$

The full system of truncated functions is obtained when reaching the highest level,

$$\mathcal{T}(\widehat{\mathcal{B}},\widehat{\Omega}) = \mathcal{T}^N(\widehat{\mathcal{B}},\widehat{\Omega}^N) \quad\text{and}\quad \mathcal{T}(\widehat{\mathcal{C}},\widehat{\Sigma}) = \mathcal{T}^N(\widehat{\mathcal{C}},\widehat{\Sigma}^N). \tag{17}$$

Comparing this construction with the definition in [31] confirms that the non-zero elements

$$\mathcal{T}^\ell(\widehat{\mathcal{B}},\widehat{\Omega}^\ell)_0 \quad\text{and}\quad \mathcal{T}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma}^\ell)_0 \tag{18}$$

of these vectors are the *truncated hierarchical (TH-) B-splines* on the domain and on its side, which are associated with the partial subdomain hierarchies. In particular, the non-zero elements of (17) are the THB-splines of the full subdomain hierarchies. Among other results, this also implies the linear independence of these functions.

Indeed, any non-zero function $\varphi\in\mathcal{T}^\ell(\widehat{\mathcal{B}},\widehat{\Omega})$ and $\psi\in\mathcal{T}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma})$ is either an element of $\mathcal{A}^\ell(\widehat{\mathcal{B}},\widehat{\Omega}^\ell)$ and $\mathcal{A}^\ell(\widehat{\mathcal{C}},\widehat{\Sigma}^\ell)$, or it has been generated by repeatedly applying truncation

to some $\beta_{\boldsymbol{i}}^{\ell'} \in \mathcal{S}^{\ell'}(\widehat{\mathcal{B}}, \widehat{\Omega})$ and $\gamma_{\boldsymbol{j}}^{\ell'} \in \mathcal{S}^{\ell'}(\widehat{\mathcal{C}}, \widehat{\Sigma})$ with $\ell' < \ell$, respectively. In the latter case it satisfies

$$\varphi|_{\Omega \setminus \Omega^{\ell'+1}} = \beta_{\boldsymbol{i}}^{\ell'}|_{\Omega \setminus \Omega^{\ell'+1}} \neq 0 \quad \text{and} \quad \psi|_{\Sigma \setminus \Sigma^{\ell'+1}} = \gamma_{\boldsymbol{j}}^{\ell'}|_{\Sigma \setminus \Sigma^{\ell'+1}} \neq 0.$$

Consequently, there is a one-to-one correspondence between the non-zero elements of the vector of truncated functions $\mathcal{T}^{\ell}(\widehat{\mathcal{B}}, \widehat{\Omega}^{\ell})_0$ and the hierarchical basis $\mathcal{H}(\widehat{\mathcal{B}}, \widehat{\Omega}^{\ell})$, which is obtained by considering the partial subdomain hierarchy, and the values of corresponding functions take the same value on $\Omega \setminus \Omega^{\ell'+1}$, where ℓ' is the associated level. Similarly, there is a one-to-one correspondence between the non-zero elements of vector of truncated functions $\mathcal{T}^{\ell}(\widehat{\mathcal{C}}, \widehat{\Sigma}^{\ell})_0$ and the hierarchical basis $\mathcal{H}(\widehat{\mathcal{C}}, \widehat{\Sigma}^{\ell})$, which is obtained by considering the partial subdomain hierarchy on the side. Corresponding functions take the same values on $\Sigma \setminus \Sigma^{\ell'+1}$, where ℓ' is the associated level. These facts can be used to prove the linear independence, see [31].

4.3. Boundary compatibility

Now we are ready to present the main result of this section.

Theorem 10. *The THB-splines $\mathcal{T}(\widehat{\mathcal{B}}, \widehat{\Omega})$ and $\mathcal{T}(\widehat{\mathcal{C}}, \widehat{\Sigma})$ are compatible if the subdomain hierarchy satisfies the weak condition.*

Proof. We use Lemma 2 to prove that the generating systems $\mathcal{T}^{\ell}(\widehat{\mathcal{B}}, \widehat{\Omega}^{\ell})$ and $\mathcal{T}^{\ell}(\widehat{\mathcal{C}}, \widehat{\Sigma}^{\ell})$ are compatible for $\ell = 0, \dots, N$.

Lemma 6 confirms this fact for $\ell = 0$. In order to proceed from $\ell - 1$ to ℓ, we need to construct a bijection between

$$(\mathcal{T}^{\ell}(\widehat{\mathcal{B}}, \widehat{\Omega}^{\ell})|_{\Sigma})_0 \quad \text{and} \quad \mathcal{T}^{\ell}(\widehat{\mathcal{C}}, \widehat{\Sigma}^{\ell})_0. \tag{19}$$

that identifies functions with identical values, and we need to show that there are no repeated non-zero entries in $\mathcal{T}^{\ell}(\widehat{\mathcal{C}}, \widehat{\Sigma}^{\ell})$. The bijection is constructed in two steps.

In the first step, we consider the non-zero parts

$$(\mathrm{trunc}^{\ell}\, \mathcal{T}^{\ell-1}(\widehat{\mathcal{B}}, \widehat{\Omega}^{\ell-1})|_{\Sigma})_0 \quad \text{and} \quad (\mathrm{trunc}^{\ell}\, \mathcal{T}^{\ell-1}(\widehat{\mathcal{C}}, \widehat{\Sigma}^{\ell-1}))_0 \tag{20}$$

of the two vectors that occur in the recursive definition (16). We use the bijection between

$$(\mathcal{T}^{\ell-1}(\widehat{\mathcal{B}}, \widehat{\Omega}^{\ell-1})|_{\Sigma})_0 \quad \text{and} \quad \mathcal{T}^{\ell-1}(\widehat{\mathcal{C}}, \widehat{\Sigma}^{\ell-1})_0,$$

and use the fact that performing the truncation trunc^{ℓ} preserves the equality of values on the side, due to Lemma 9. This gives us the required bijection between the vectors in (20), which identifies functions taking identical values on the side.

In the second step we extend this to the required bijection between the two vectors in (19). Owing to the recursive definition (16), this is achieved by combining it with the bijection between the vectors

$$(\mathcal{A}^{\ell}(\widehat{\mathcal{B}}, \widehat{\Omega}^{\ell})|_{\Sigma})_0 \quad \text{and} \quad \mathcal{A}^{\ell}(\widehat{\mathcal{C}}, \widehat{\Sigma}^{\ell})_0$$

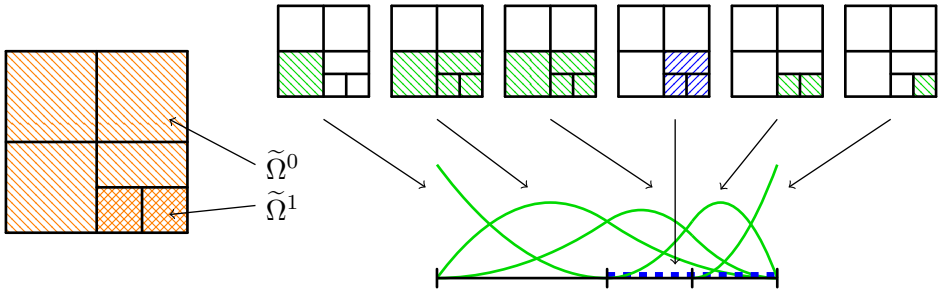


Figure 5: THB-splines on the mesh shown in Figure 2, which satisfies the weak, but not the strong condition. The tensor-product B-spline of level 0, which does not vanish at the boundary, is truncated to zero.

that contain the active functions. The existence of the latter bijection is guaranteed by Lemmas 2 and 8.

Finally we note that there are no repeated entries in $\mathcal{T}^\ell(\widehat{\mathcal{C}}, \widehat{\Sigma}^\ell)_0$ as the non-zero elements of each vector of truncated B-splines are linearly independent. This completes the proof, due to (17). □

Recall that the hierarchical mesh shown in Figure 2 satisfies the weak condition, but not the strong condition. As illustrated in Figure 2, the HB-splines defined on this mesh are not boundary compatible. The THB-splines defined on the same mesh are boundary compatible, which is illustrated in Figure 5. The spline of level 0, which does not vanish at the considered boundary in Figure 2, is truncated to zero during the THB-splines construction (its support is indicated by the blue area).

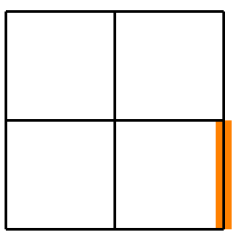
4.4. Geometric modeling examples

Finding a valid domain mesh from a given boundary mesh can be a challenging task in geometric modeling. In the case of THB-/HB-splines, we know from the previous sections that it is sufficient to satisfy the weak/strong condition to guarantee boundary compatibility. The task is thus reduced to adding one layer of cells on the correct levels, as indicated in the following two examples.

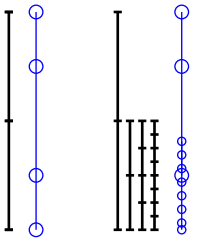
Academic example

Our first example illustrates how a small part of a domain boundary can be modified locally by modifying only the boundary curve at first, and then obtaining the domain mesh from the boundary mesh in a second step.

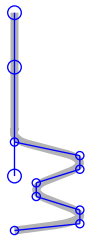
Figure 6(a) shows the unit square, represented by a 2×2-mesh, and an area, marked in orange, where the boundary curve should be modified in order to design a local feature. Instead of refining the full mesh, we consider only the boundary mesh, see Figure 6(b). The original mesh with two knot spans and four control points is shown on the left of Figure 6(b). On the right, the refined hierarchical mesh with four levels is illustrated together with the corresponding control polygon. Figure 6(c) shows a modification of this control polygon and the resulting curve.



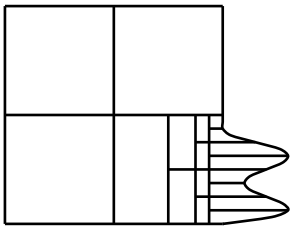
(a) Original domain. Area to be modified marked in orange.



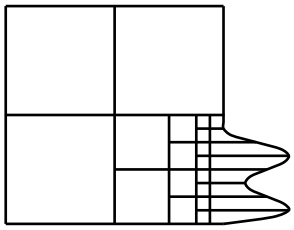
(b) Original (left) and refined (right) boundary subdomain hierarchy with corresponding control polygons.



(c) Modified control polygon and curve.



(d) Modified mesh, satisfying weak condition.



(e) Modified mesh, satisfying strong condition.

Figure 6: Local modification of boundary curve by refinement of boundary subdomain hierarchy.

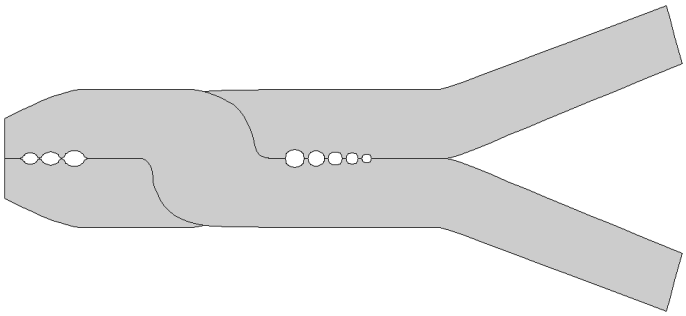
Starting from the hierarchical boundary mesh shown in Figure 6(b), the domain mesh is obtained by taking its closure on all levels. The resulting meshes are shown in Figure 6(d) and 6(e) for the weak and the strong condition, respectively.

This simple example is chosen such that the manipulation of the control points results in some distorted cells near the modified area, but such that the geometry mapping stays regular (without self-overlaps). A more complicated geometry is shown in our next example.

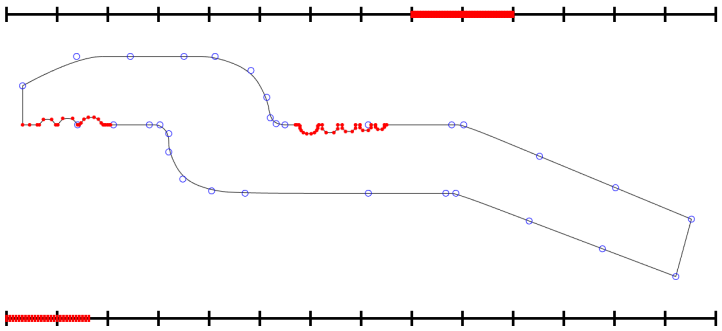
Wire cutter

The wire cutting tool depicted in Figure 7(a) has some small details incorporated into an otherwise simple geometry, which motivates the use of THB-/HB-splines. The initial shape is obtained by a refinement and manipulation of the boundary curves only, which is illustrated in Figure 7(b). The knots which are inserted in the original, black knot vectors, are indicated by red, shorter lines (they may appear like a single, thick line but are actually many single knots). The corresponding refined and modified control points are indicated by solid, red points, while the blue circles mark the original, coarse boundary control points.

Figure 8 shows the closures of the boundary hierarchy to hierarchical domain meshes under the weak and the strong condition, and the extension to a tensor-product mesh. As a difference to the previous, simpler example, only taking the closure of the boundary hierarchy would lead to self-overlaps and thus an irregular mesh. Further refinement near the small details is needed to obtain a sufficient number of control points which are



(a) Wire cutter.



(b) Control points and locally refined knot vectors of boundary curves.

Figure 7: Wire cutter geometry and boundary curves with small details.

adjusted to a regular geometry mapping, see Figure 9.

In this particular example, the adjustments were done manually. The issue of automatic generation of a surface mesh from given boundary curves such that certain quality criteria are satisfied is a field of active research. An in-depth discussion of such methods is beyond the scope of this paper, and the reader is referred to, e.g., [27, 33, 53, 68] and the references therein.

5. Multipatch THB-/HB-structures

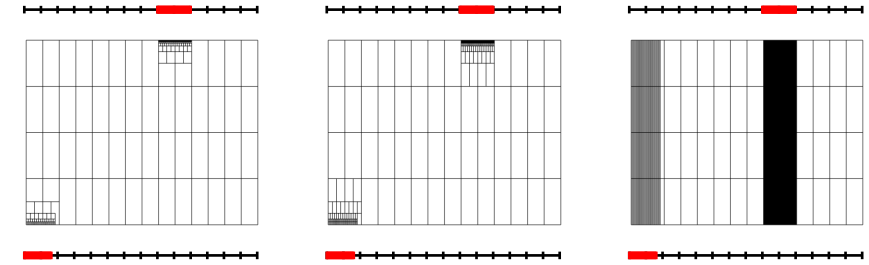
In this section, we define a THB-/HB-spline multi-patch structure and discuss some conditions, which ensure that the patch-wise defined THB-/HB-splines can be glued.

5.1. Geometrically conforming multi-patch structures

We consider a physical domain $\widetilde{\Omega} \subset \mathbb{R}^d$ which is given as a union of patches $\widetilde{\Omega}_p$, each of which is defined by a geometry mapping $G_p : \Omega \to \mathbb{R}^d$, $p \in \mathcal{P}$,

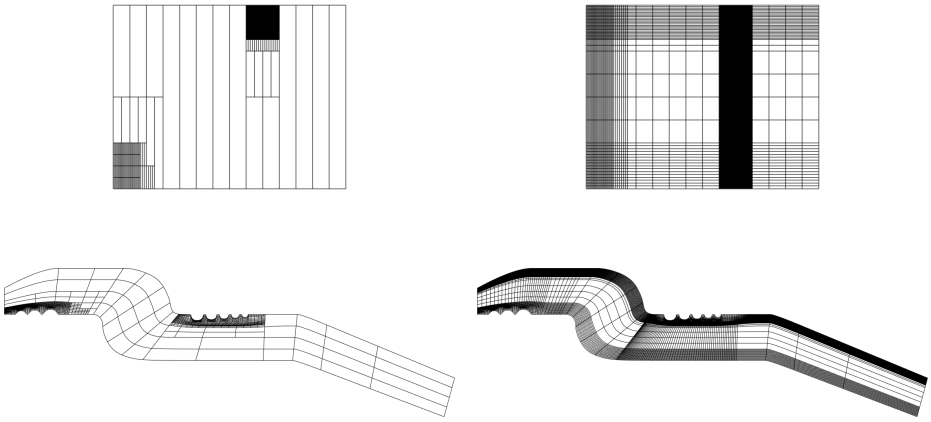
$$\widetilde{\Omega} = \bigcup_{p\in\mathcal{P}} \widetilde{\Omega}_p, \qquad \text{where } \widetilde{\Omega}_p = G_p(\Omega). \tag{21}$$

While we assume that all patches are images of the parameter domain $[0,1]^d$, we use the subscript index to indicate that the parameter domain Ω_p is associated with the



THB-splines (263 DOF) HB-splines (323 DOF) TP-splines (603 DOF)

Figure 8: Closures of the boundary meshes shown in Figure 7 under the weak condition (left), the strong condition (middle), and with a tensor-product mesh (right).



HB-splines (2129 DOF) TP-splines (5184 DOF)

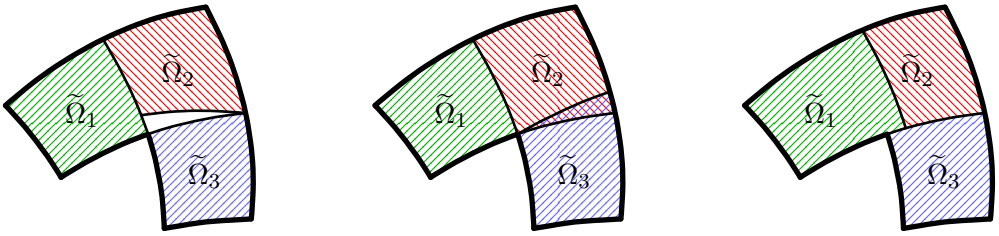
Figure 9: Extended meshes from Figure 8 after adjustment near the holes, shown on parameter domain (top) and physical domain (bottom).

geometry mapping G_p. We assume that all geometry mappings are regular on Ω_p and that the patches are mutually disjoint,

$$\operatorname{int}(\widetilde{\Omega}_p) \cap \operatorname{int}(\widetilde{\Omega}_q) = \emptyset, \text{ for } p \neq q, \tag{22}$$

where $\operatorname{int}(A)$ denotes the interior of A. The boundary of A will be denoted by ∂A.

Recall that the unit cube $\Omega_p = [0,1]^d$ consists of *n-faces* for $n = 0, \ldots, d$. For instance, for $d = 3$ the notion of n-face refers to vertices, edges, faces, or the entire volume for $n = 0, 1, 2, 3$, respectively. Note that Ω_p is a d-face (and the only one) of Ω_p. This helps to unify the notation later on. The boundary $\partial\Omega_p$ of Ω_p consists of boundary n-faces, $n = 0, \ldots, d-1$. An *n-face* of $\widetilde{\Omega}_p$ is the image of an *n-face* of Ω_p under the geometry mapping G_p. Again, $\widetilde{\Omega}_p$ is its only d-face, and $\partial\widetilde{\Omega}_p$ consists of boundary n-faces, $n = 0, \ldots, d-1$.



(a) Gap violating (21). (b) Overlap violating (22). (c) Not all interfaces are boundary n-faces.

Figure 10: Examples of multipatch structures which are not geometrically conforming.

We assume that the intersections

$$\partial\widetilde{\Omega}_p \cap \partial\widetilde{\Omega}_q, \quad p \neq q,$$

are either empty or boundary n-faces of both $\partial\widetilde{\Omega}_p$ and $\partial\widetilde{\Omega}_q$. A multi-patch structure $\widetilde{\Omega}$ which satisfies this assumption and additionally (21) and (22), is called *geometrically conforming* (see Figure 10 for examples of multi-patch structures which are not geometrically conforming). For the remainder of this paper, we will only consider geometrically conforming multi-patch structures.

If an n-face $\widetilde{\Sigma}$ with $n < d$ is shared by more than one patch, we say that these patches are *neighbouring* patches, and we call $\widetilde{\Sigma}$ an *interface*. We collect the index-pairs of patches which share an interface in the set

$$\mathcal{N} = \{(p,q) : \ \partial\widetilde{\Omega}_p \cap \partial\widetilde{\Omega}_q \neq \emptyset, \ p \neq q\}, \tag{23}$$

and we denote the interface of two patches $\widetilde{\Omega}_p$ and $\widetilde{\Omega}_q$ by

$$\widetilde{\Sigma}_{pq} = \partial\widetilde{\Omega}_p \cap \partial\widetilde{\Omega}_q = \widetilde{\Sigma}_{qp}, \quad \text{for } (p,q) \in \mathcal{N}.$$

Note that an interface of the multi-patch structure is not necessarily a $d-1$-face, but may also be, e.g., an edge or a single vertex. Interfaces may be associated with more than one pair of patches, see Figure 11 for an example.

5.2. THB-/HB-spline multi-patch

A *THB-/HB-spline multi-patch* is multi-patch structure composed of THB-/HB-spline patches. As mentioned above, we only consider geometrically conforming multi-patch structures. The parameter domains Ω_p of the patches $\widetilde{\Omega}_p$ have associated mesh hierarchies, subdomain hierarchies, and tensor-product B-spline hierarchies, which are denoted by $\widehat{\mathcal{M}}_p = (\mathcal{M}_p^\ell)_{\ell=0,\dots,N}$, $\widehat{\Omega}_p = (\Omega_p^\ell)_{\ell=0,\dots,N}$, and $\widehat{\mathcal{B}}_p = (\mathcal{B}_p^\ell)_{\ell=0,\dots,N}$, respectively.

We assume that the THB-/HB-spline multi-patch has *matching tensor-product B-spline hierarchies*, in the following sense: For any tensor-product B-spline $\beta_{p,\boldsymbol{i}}^\ell$ of level ℓ on patch p which has a non-vanishing push-forward onto $\widetilde{\Sigma}_{pq}$, there exists exactly one

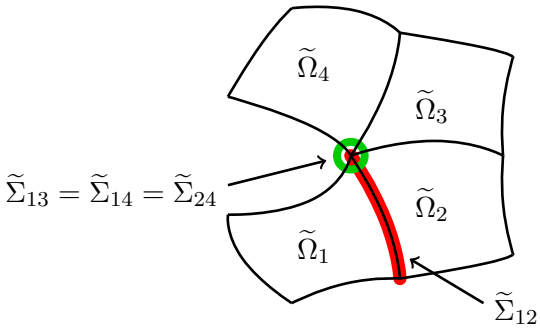


Figure 11: Examples of interfaces on a four-patch structure in $\mathbb{R}^2$.

tensor-product B-spline $\beta^\ell_{q,\boldsymbol{j}}$ of level ℓ on patch q, such that its push-forward equals $\beta^\ell_{p,\boldsymbol{i}}$ on $\widetilde{\Sigma}_{pq}$, i.e.,

$$\forall \boldsymbol{i} \in \mathcal{I}^\ell_p,\ \left(\beta^\ell_{p,\boldsymbol{i}} \circ G_p^{-1}\right)\big|_{\widetilde{\Sigma}_{pq}} \neq 0,\ \exists!\, \boldsymbol{j} \in \mathcal{I}^\ell_q :\ \left(\beta^\ell_{p,\boldsymbol{i}} \circ G_p^{-1}\right)\big|_{\widetilde{\Sigma}_{pq}} = \left(\beta^\ell_{q,\boldsymbol{j}} \circ G_q^{-1}\right)\big|_{\widetilde{\Sigma}_{pq}}, \tag{24}$$

where $\mathcal{I}^\ell_p$ and $\mathcal{I}^\ell_q$ denote the index sets of all tensor-product B-splines of level ℓ on patch p and q, respectively. If two basis functions $\beta_{p,\boldsymbol{i}}$ and $\beta_{q,\boldsymbol{j}}$ satisfy the equality in (24), we call them *matching basis functions* on $\widetilde{\Sigma}_{pq}$, see Figure 12 for an illustration. Recall that p indicates the patch index, not the polynomial degree of the tensor-product B-spline.

Note that matching tensor-product B-spline hierarchies also imply *matching mesh hierarchies* in the sense that, for any element $e_p \in \mathcal{L}^\ell_p(\Omega_p)$ of a fixed level ℓ, which intersects $G_p^{-1}(\widetilde{\Sigma}_{pq})$, there exists exactly one element $e_q \in \mathcal{L}^\ell_q(\Omega_q)$ of the same level ℓ such that

$$G_p(e_p) \cap \widetilde{\Sigma}_{pq} = G_q(e_q) \cap \widetilde{\Sigma}_{pq}. \tag{25}$$

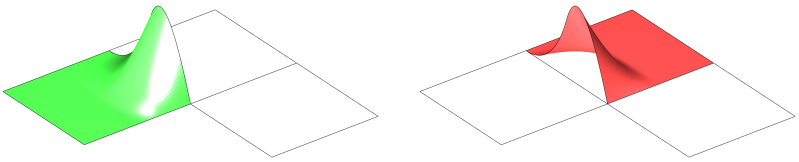
5.3. Global THB-/HB-spline basis

Two (or more) basis functions with are matching at an interface can be associated with one global degree of freedom. This can be viewed as constructing a global basis function which is patchwise defined by the respective local basis functions and which is C^0-continuous at interfaces, see Figure 12 for an illustration. The resulting basis is thus globally C^0-conforming. We refer to this method as *gluing* the bases. If local bases of a multi-patch structure can be glued at all interfaces, we say that the multi-patch can be glued.

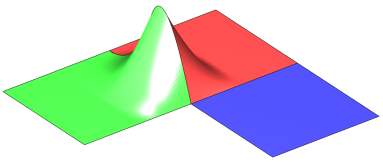
In the following theorem, the results from Section 3 are used to provide a condition under which THB-/HB-spline multi-patch discretization can be glued.

Theorem 11. *Let $\widetilde{\Omega}$ be a geometrically conforming THB-spline/HB-spline multi-patch with matching tensor-product B-spline hierarchies, and assume that all patches satisfy the weak/strong condition. The patch-wise defined THB-splines/HB-splines can be glued to form a globally C^0-conforming basis on $\widetilde{\Omega}$ if, for all $(p,q) \in \mathcal{N}$, the subdomain hierarchies are compatible in the sense that*

$$G_p\big(\Omega^\ell_p\big) \cap \widetilde{\Sigma}_{pq} = G_q\big(\Omega^\ell_q\big) \cap \widetilde{\Sigma}_{pq}, \quad \forall \ell = 0, \dots, N. \tag{26}$$



(a) Two matching basis functions.



(b) Glued, globally C^0-continuous basis functions.

Figure 12: Gluing of matching basis functions.

Proof. Consider the interface hierarchies which are obtained by restricting the mesh and subdomain hierarchies of any two neighbouring patches to their common interface. Due to the matching tensor-product B-spline hierarchies, mesh hierarchies, and subdomain hierarchies, the THB-/HB-spline bases on these interface hierarchies must be matching. Since the weak/strong condition is satisfied, we know from Theorem 7, respectively Theorem 10 that the bases on the interface are compatible with the "full" bases on the respective patches due to the compatibility property, cf. Definition 1. Hence, the bases are matching and can be glued. □

6. Adaptive refinement on THB-/HB-multi-patches

In this section, we recall the basic concept of adaptive refinement and we present an algorithm which ensures that the conditions given in Section 5.3 remain satisfied.

6.1. Adaptive refinement loop

An adaptive strategy for numerically solving a partial differential equation (PDE) typically consists of the following steps, to which we refer as the *adaptive loop*:

1. Compute a numerical solution of the PDE.
2. Estimate the error distribution.
3. Mark areas with large error for refinement.
4. Refine marked cells.

Continue with step 1. until stopping criterion is satisfied.

Depending on the actual problem, a large number of methods is available for steps 1 to 3, the discussion of which is beyond the scope of this paper. For details, the reader is referred to, e.g., [1, 18, 26, 30, 34], or the monographs [2, 56] and the references therein.

In the context of THB-/HB-splines, step 4, i.e., local refinement, corresponds to increasing the level of marked cells. Geometric conformity and matching tensor-product

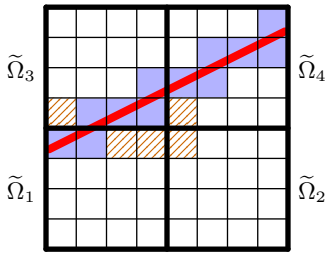


Figure 13: Refinement along the red line leading to marking of blue colored cells. Areas where subdomains are no longer compatible indicated by orange areas.

B-spline hierarchies are preserved throughout the adaptive loop. By choosing of a proper marking and refinement method, the weak/strong condition can also be ensured easily. For example, if the marked areas are composed of cells of a certain level ℓ, and these cells are then shifted to level $\ell + 1$, the strong condition is automatically satisfied.

However, if an area next to an interface is refined, this may lead to a situation where subdomain hierarchies are no longer compatible. An example is shown in Figure 13, where a four-patch structure (patch-boundaries are indicated by thick lines) is refined along the red line, which crosses through patches $\widetilde{\Omega}_1$, $\widetilde{\Omega}_3$, and $\widetilde{\Omega}_4$. The marked elements, indicated blue, touch interfaces and, after refinement, result in a setting where the subdomains are no longer compatible. The orange areas mark additional cells which would have to be refined in order to restore compatibility.

6.2. Algorithm for ensuring compatible subdomain hierarchies

In this section, we propose an algorithm which will modify non-compatible subdomain hierarchies such that the compatibility condition (26) is satisfied again. Before proceeding, we define the *closure* $\operatorname{clos}^{\ell} A$ *of a subdomain* $A \subseteq \Omega$ *with respect to level* ℓ *under the weak/strong condition* as the minimal set of cells of level ℓ covering A and satisfying the weak/strong condition. The closure of an empty set is defined as the empty set. Figures 14(a) and 14(b) show an example of a subdomain (colored green) and of its closures under the weak, respectively, strong condition (orange areas). Note that the closure of a subdomain A which satisfies the weak/strong condition is the subdomain A itself.

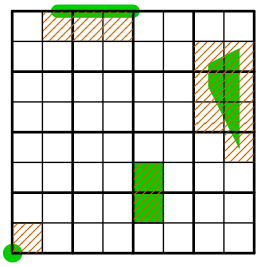
We will use the closure to extend boundary hierarchies into the domain. Note that the closure under the weak/strong condition results in a subdomain which satisfies the weak/strong condition, hence the THB-/HB-splines defined on the respective hierarchical mesh are boundary compatible.

We denote the set of indices of all patches which have an interface with $\widetilde{\Omega}_p$ by

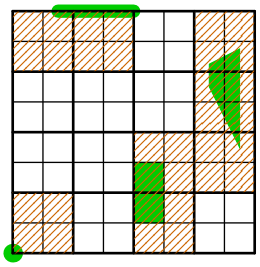
$$\mathcal{N}_p = \{q : \ (p, q) \in \mathcal{N}\}.$$

The method we propose is specified in Algorithm 1.

The algorithm obviously terminates since the number of elements for each level is finite and the number of levels remains unchanged.



(a) $\mathrm{clos}^\ell(A)$ under weak condition.



(b) $\mathrm{clos}^\ell(A)$ under strong condition.

Figure 14: Subdomain A (green area) and its closure $\mathrm{clos}^\ell A$ (orange area), with an 8×8-mesh on level ℓ and a 4×4-mesh on level $\ell - 1$.

The procedure is illustrated in Figure 15 on a three-patch structure with non-matching subdomains. The initial configuration is depicted in Figure 15(a), where the subdomains of a fixed level ℓ are indicated by the colored areas. Figure 15(b) illustrates the modification of the subdomain Ω_1^ℓ of patch $\widetilde{\Omega}_1$, i.e., for $p = 1$. The information from the neighbouring patches Ω_q, $q \in \mathcal{N}_1 = \{2, 3\}$ is transferred to the boundary of Ω_p via the entity

$$T_{pq}^\ell = G_p^{-1}\big(G_q(\Omega_q^\ell) \cap \widetilde{\Sigma}_{pq}\big), \tag{27}$$

which is indicated at the top of Figure 15(b). The new subdomain $\widetilde{Q}_p^\ell$ is defined by the closure of the union of the original subdomain Ω_p^ℓ and all neighbour information T_{pq}^ℓ, $q \in \mathcal{N}_p$, i.e., it is defined by

$$\mathrm{clos}^\ell\,\big(\Omega_p^\ell \cup \bigcup_{q \in \mathcal{N}_p} T_{pq}^\ell\big),$$

which is shown at the bottom of Figure 15(b). The resulting modified subdomains which are obtained from applying these steps to all threes patches are shown in Figure 15(c). The following theorem ensures that the subdomain hierarchies are matching, when Algorithm 1 terminates.

Theorem 12. *The subdomain hierarchies generated by Algorithm 1*

(i) are nested, if the initial subdomain hierarchies are nested, and

(ii) satisfy the compatibility condition (26) *and hence the THB-/HB-spline multi-patch spline bases can be glued to define a globally C^0-conforming discretization.*

Proof. (i) Since $\Omega_p^\ell \subseteq \Omega_p^{\ell-1}$, and since all geometry mappings are assumed to be regular, the inclusion $T_{pq}^\ell \subseteq T_{pq}^{\ell-1}$ holds. Furthermore, for any $A, B \subseteq Q$,

$$\begin{aligned} A \subseteq B \quad &\Rightarrow \quad \mathrm{clos}^\ell A \subseteq \mathrm{clos}^\ell B, \\ \mathrm{clos}^\ell A \quad &\subseteq \quad \mathrm{clos}^{\ell-1} A, \end{aligned}$$

therefore

$$\mathrm{clos}^\ell\,\big(\Omega_p^\ell \cup \bigcup_{q \in \mathcal{N}_p} T_{pq}^\ell\big) \subseteq \mathrm{clos}^{\ell-1}\,\big(\Omega_p^{\ell-1} \cup \bigcup_{q \in \mathcal{N}_p} T_{pq}^{\ell-1}\big),$$

Algorithm 1

input: Subdomain hierarchies $\widehat{\Omega}_p$, $p \in \mathcal{P}$, of a geometrically conforming THB-/HB-multi-patch with matching bases.

output: Compatible subdomain hierarchies.

---

for all $\ell = 0$ **to** N
  flagChanged $\leftarrow$ true
  while flagChanged
    flagChanged $\leftarrow$ false
    for all $p \in \mathcal{P}$
      $\Omega_p^{\ell,+} \leftarrow \operatorname{clos}^\ell \left(\Omega_p^\ell \cup \bigcup_{q \in \mathcal{N}_p} G_p^{-1}\big(G_q(\Omega_q^\ell) \cap \widetilde{\Sigma}_{pq}\big) \right)$
      if $\Omega_p^{\ell,+} \neq \Omega_p^\ell$ **then**
        flagChanged $\leftarrow$ true
      end if
      $\Omega_p^\ell \leftarrow \Omega_p^{\ell,+}$
    end for
  end while
end for

---

i.e., the modified subdomains are nested.

(ii) When Algorithm 1 has terminated, all subdomains satisfy

$$\Omega_p^\ell = \operatorname{clos}^\ell \big(\Omega_p^\ell \cup \bigcup_{q \in \mathcal{N}_p} T_{pq}^\ell\big) = \operatorname{clos}^\ell \Omega_p^\ell \cup \bigcup_{q \in \mathcal{N}_p} \operatorname{clos}^\ell T_{pq}^\ell, \tag{28}$$

where T_{pq}^ℓ is as defined in (27). Consider two neighbouring patches, Ω_p and Ω_q, $(p,q) \in \mathcal{N}$, and their interface $\widetilde{\Sigma}_{pq}$. Let

$$x \in G_p(Q_p^\ell) \cap \widetilde{\Sigma}_{pq}.$$

Then, by definition of $T_{qp}^\ell = G_q^{-1}(G_p(Q_p^\ell) \cap \widetilde{\Sigma}_{pq})$, it follows that

$$G_q^{-1}(x) \in T_{qp}^\ell \ \subseteq \ \operatorname{clos}^\ell T_{qp}^\ell \ \subseteq \ \Omega_q^\ell,$$

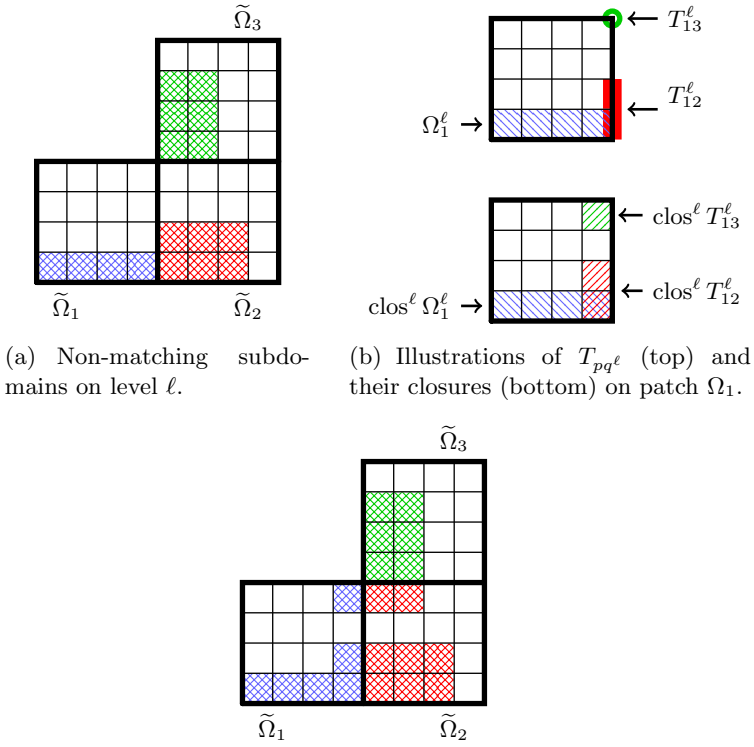
i.e., $x \in G_q(\Omega_q^\ell)$ and thus

$$x \in G_q(\Omega_q^\ell) \cap \widetilde{\Sigma}_{pq}.$$

Since x was arbitrary in $G_p(Q_p^\ell) \cap \widetilde{\Sigma}_{pq}$, we conclude that

$$G_p(Q_p^\ell) \cap \widetilde{\Sigma}_{pq} = G_q(Q_q^\ell) \cap \widetilde{\Sigma}_{pq},$$

and therefore, it follows from Theorem 11 that the subdomains can be glued. □



(a) Non-matching subdomains on level ℓ.

(b) Illustrations of T_{pq^ℓ} (top) and their closures (bottom) on patch Ω_1.

(c) Matching subdomains after application of Alg. 1.

Figure 15: Illustration of Algorithm 1 on a three-patch structure with non-matching subdomains on a fixed level ℓ.

The number of loops executed in Algorithm 1 depends on the configuration of the patches and meshes. In particular, if the subdomains are already matching (and satisfy the weak/strong condition), then no subdomain is modified and the algorithm will terminate after one loop.

Figure 16 shows a worst-case example with $|\mathcal{P}|$ patches aligned in a row. The patches $\widetilde{\Omega}_1$ to $\widetilde{\Omega}_{|\mathcal{P}|-1}$ have meshes where a single cell extends form the eastern to the western patch boundary. The blue area on patch $\widetilde{\Omega}_{|\mathcal{P}|}$ triggers a modification on its western neighbour, which, in the next loop, will cause a modification on its western neighbour, and so on. In this specially chosen setting, Algorithm 1 will require $|\mathcal{P}|$ many loops before it terminates.

The issue in this example is that the closure of a boundary mesh on the eastern interface $\widetilde{\Sigma}_{p,p+1}$ also affects the western interface $\widetilde{\Sigma}_{p-1,p}$, without the possibility of a direct communication between the patches $\widetilde{\Omega}_{p-1}$ and $\widetilde{\Omega}_{p+1}$. This can be avoided, for example, if the meshes on the patches are fine enough to limit the effect of the closure operation, see Figure 17 for an example.

In general, the algorithm will terminate after one loop, if the communication between

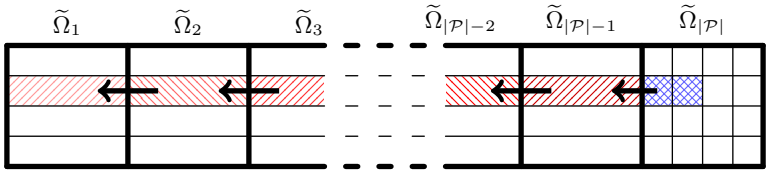


Figure 16: Worst-case setting where Algorithm 1 requires $|\mathcal{P}|$ many steps until it terminates.

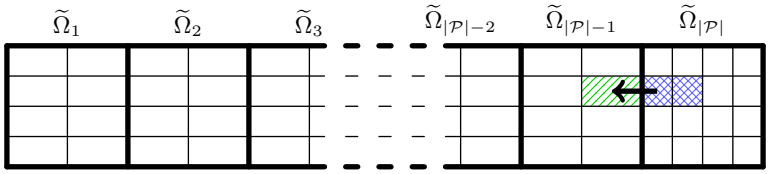


Figure 17: Setting similar to Figure 16, but with a mesh which is fine enough to limit the effect of the closure operation.

patches is sufficient in the following sense: If a subdomain on patch $\widetilde{\Omega}_r$ triggers a refinement in a neighbouring patch $\widetilde{\Omega}_q$ (in order to repair interface $\widetilde{\Sigma}_{qr}$), which then leads to a refinement on a third patch $\widetilde{\Omega}_p$, then this refinement on $\widetilde{\Omega}_p$ must also be induced directly from patch $\widetilde{\Omega}_r$ via the interface $\widetilde{\Sigma}_{pr}$. Formally, this is expressed by

$$G_p\big(\operatorname{clos}^\ell T_{pr}\big) \cap \widetilde{\Sigma}_{pq} = G_q\big(\operatorname{clos}^\ell T_{qr}\big) \cap \widetilde{\Sigma}_{pq} \quad \forall p, q, r \in \mathcal{P}. \tag{29}$$

Note that this condition depends on the configuration of patches, their meshes, and on the closure operation. The setting presented in Figure 17 satisfies (29), if the closure is taken with respect to the weak condition, but not if it is taken with respect to the strong condition.

Figure 18 shows an example where the blue subdomain on patch $\widetilde{\Omega}_3$ triggers a refinement on patch $\widetilde{\Omega}_2$ (indicated by the black arrow) in order to repair interface $\widetilde{\Sigma}_{23}$. This, in turn, leads to a refinement on patch $\widetilde{\Omega}_1$ (indicated by the red dashed arrow). The refinement on $\widetilde{\Omega}_1$, however, is also induced directly from patch $\widetilde{\Omega}_3$ via the interface vertex $\widetilde{\Sigma}_{13}$ (indicated by the greed dotted arrow). In this setting, the subdomains will match after a single loop of Algorithm 1.

6.3. Numerical examples

The following two examples illustrate the application of Algorithm 1 in numerical simulation on multi-patch structures. Note that in both examples, refinement was chosen such that the strong condition is satisfied.

Laplace equation on circular domain

In this example, we consider a circular domain which is represented *exactly* by a five-patch NURBS-structure as indicated in Figure 19(a). We solve the Laplace equation

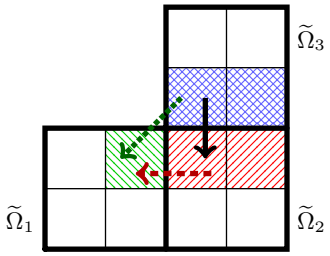


Figure 18: Setting where the communication between patches is sufficient to terminate Algorithm 1 after one loop.

with homogenous Dirichlet boundary conditions,

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \operatorname{int}(\widetilde{\Omega}), \\ u &= 0 \qquad \text{on } \partial\widetilde{\Omega}, \end{aligned}$$

where the right hand side is chosen such that the eyes and the mouth of a smiling face are produced at the positions indicated by the dashed lines in Figure 19(a). The computed solution after four steps of adaptive refinement, which is shown in Figure 19(b), nicely captures these features. The corresponding meshes are presented in Figures 19(c) and 19(d) in total and in a zoomed-in view, respectively.

Comparing the meshes before (on the left) and after (on the right) application of Alorithm 1, shows that interfaces with non-compatible subdomain hierarchies are successfully repaired. Even though this example was chosen such that refinement occurs at or near patch interfaces, only little refinement is added by Algorithm 1.

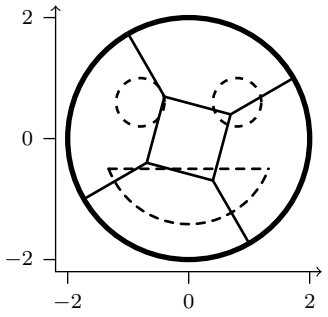
Linear elasticity on furniture foot

In our final example, we consider a 3D linear elasticity problem on the domain shown in Figure 20(a), respresenting a furniture foot. The object is $96mm$ high and composed of five patches, see Figure 20(b). The material parameters are set to $E = 69kN/mm^2$ (Young's modulus), $\nu = 0.334$ (Poisson's ratio), and $\rho = 2.7mg/mm^3$ (density). The foot is fixed at the bottom, while the top surfaces are subject to a constant downward-pointing force of $0.04N/mm^2$, which results in a total load of est. $165N$.

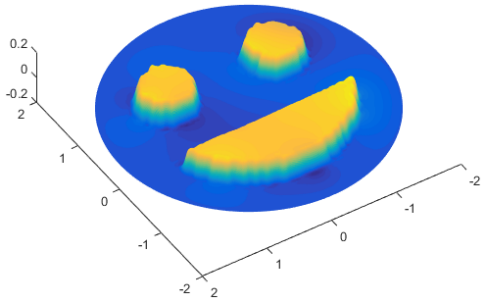
As in the previous example, the mesh is refined adaptively using a residual-based error estimator. In each step, we refine all cells e, for which the local error estimate η_e satisfies

$$\eta_e \geq 0.25 \cdot \max_{e'} \eta_{e'}.$$

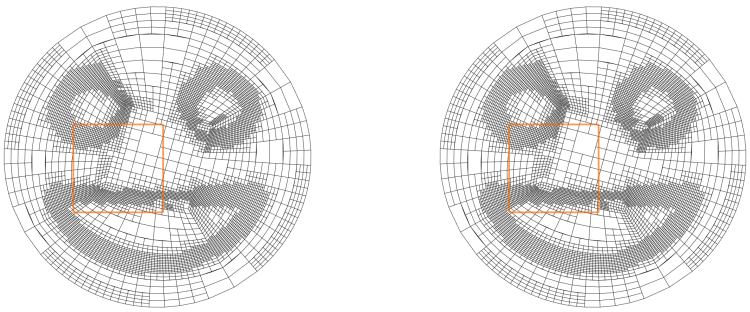
This strategy is referred to as *maximum marking* in [34] and as *absolute threshold marking* in [30]. The mesh after three refinement steps is shown in Figure 21(a), the corresponding distribution of the von Mises stress in Figure 21(b). Figure 21(d) shows a zoomed view of the stress peak which appears at the interface of two patches, and Figure 21(c) the adaptively refined mesh in this area. As in the previous example, the application of Algorithm 1 ensures matching subdomains. The energy norm of the computed solutions is plotted in Figure 22 and shows that the adaptively refined solution



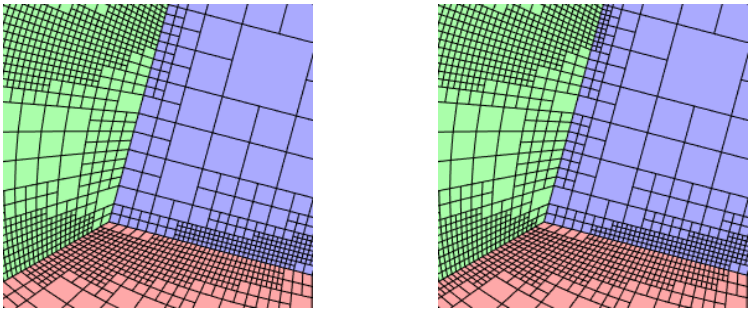
(a) Multi-patch structure and problem setting



(b) Numerical solution after 4 refinement steps.

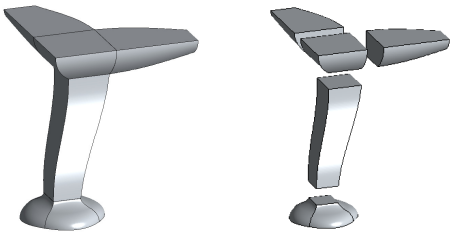


(c) Mesh after 4 refinement steps, before (left) and after (right) application of Algorithm 1. Area marked by orange box is enlarged in Figure 19(d).



(d) Zoom in on meshes after 4 refinement steps, before (left) and after (right) application of Algorithm 1 (different colors indicate different patches).

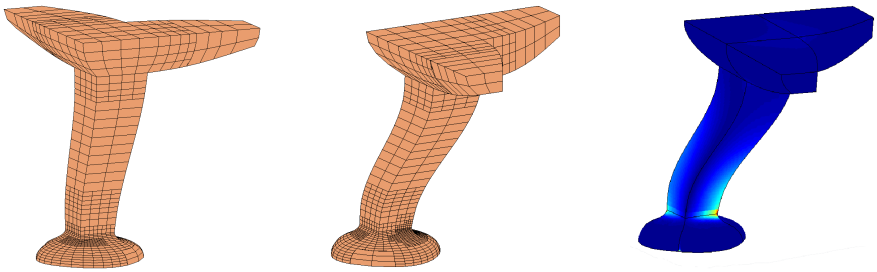
Figure 19: Problem setting and adaptively refined meshes in example “Laplace equation on circular domain”.



(a) Geometry. (b) 5 patches.

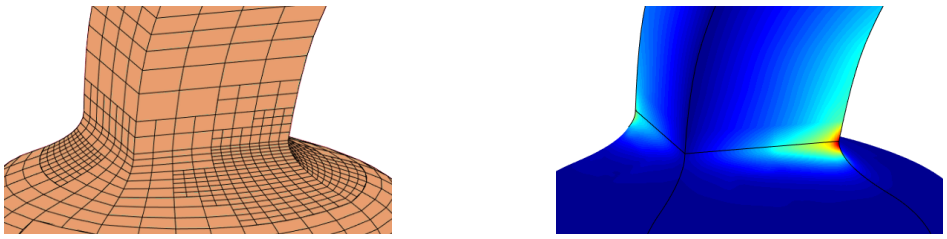
Figure 20: Physical domain in furniture foot example, composed of 5 patches.

converges faster (on a degree-of-freedom basis) than the solution computed with uniform, global refinement.



(a) Mesh after three refinement steps.

(b) Von Mises stress distribution.



(c) Zoomed view of mesh near stress peak.

(d) Zoomed view of stress peak.

Figure 21: Adaptively refined mesh and stress distribution in furniture foot example.

7. Conclusion

Given two systems of functions $\mathcal{X}$ and $\mathcal{Y}$ defined on the unit box (in any dimension) and on one of its sides, respectively, we introduced the notion of compatibility of these systems. Essentially, it is required that the restriction of the system $\mathcal{X}$ to the side is in one-to-one correspondence to the system $\mathcal{Y}$. After introducing this property in an abstract framework, we analyzed it in the case of several constructions of spline spaces, in particular focusing on (truncated) hierarchical B-splines. We identified conditions that guarantee the compatibility property and discussed its application to geometric

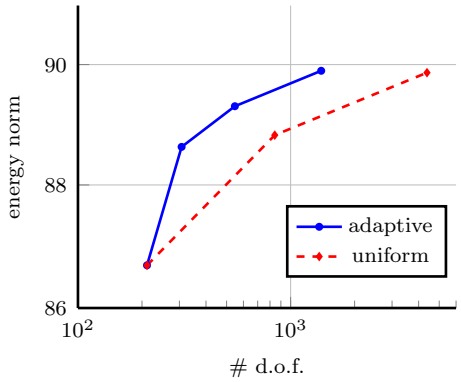


Figure 22: Energy norm convergence in furniture foot example, comparison of adaptive and uniform refinement.

modeling. These conditions are dimension-independent and can be checked easily. We also showed how to use them in an adaptive refinement scheme for multi-patch spline spaces. Future work might be devoted to the analysis of conditions that guarantee this property of locally refined (LR) B-splines also.

Acknowledgements

Funding: This work was supported by the Austrian Science Fund [NFN S117 "Geometry + Simulation"]; and the European Union through the Horizon 2020 programme [GA no. 678727 "MOTOR"].

[1] M. Ainsworth and J.T. Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142(1-2):1–88, 1997.

[2] M. Ainsworth and J.T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. Pure and Applied Math. Wiley, 2000.

[3] A. Apostolatos, R. Schmidt, R. Wüchner, and K.-U. Bletzinger. A Nitsche-type formulation and comparison of the most common domain decomposition methods in isogeometric analysis. *International Journal for Numerical Methods in Engineering*, 97(7):473–504, 2014.

[4] Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric Analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(7):1031–1090, 2006.

[5] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):229–263, 2010.

[6] Y. Bazilevs, V.M. Calo, Y. Zhang, and T.J.R. Hughes. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38:310–322, 2006.

[7] Y. Bazilevs, J.R. Gohean, T.J.R. Hughes, R.D. Moser, and Y. Zhang. Patient-specific isogeometric fluidstructure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Computer Methods in Applied Mechanics and Engineering*, 198(45-46):3534–3550, 2009.

[8] L. Beirão da Veiga, A. Buffa, D. Cho, and Sangalli G. Analysis-Suitable T-splines are Dual-Compatible. *Computer Methods in Applied Mechanics and Engineering*, 249-252:42–51, 2012.

[9] L. Beirão da Veiga, A. Buffa, J. Rivas, and G. Sangalli. Some estimates for h-p-k-refinement in isogeometric analysis. *Numerische Mathematik*, 118:271–305, 2011.

[10] L. Beirão da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Analysis-suitable T-splines of arbitrary degree: Definition, linear independence and approximation properites. *Mathematical Models and Methods in Applied Sciences*, 23(11):1979–2003, 2013.

[11] L. Beirão da Veiga, A. Buffa, G. Sangalli, and R. Vázquez. Mathematical analysis of variational isogeometric methods. *Acta Numerica*, 23:157–287, 5 2014.

[12] L. Beirão da Veiga, D. Cho, L.F. Pavarino, and S. Scacchi. BDDC Preconditioners for Isogeometric Analysis. *Mathematical Models and Methods in Applied Sciences*, 23(06):1099–1142, 2013.

[13] L. Beirão da Veiga, D. Cho, L.F. Pavarino, and S. Scacchi. Isogeometric Schwarz preconditioners for linear elasticity systems. *Computer Methods in Applied Mechanics and Engineering*, 253(0):439–454, 2013.

[14] A. Bressan. Some Properties of LR-splines. *Comput. Aided Geom. Des.*, 30(8):778–794, 2013.

[15] E. Brivadis, A. Buffa, B. Wohlmuth, and L. Wunderlich. Isogeometric mortar methods. *Computer Methods in Applied Mechanics and Engineering*, 284:292–319, 2015. Isogeometric Analysis Special Issue.

[16] F. Buchegger and B. Jüttler. Planar multi-patch domain parameterization via patch adjacency graphs. *Computer-Aided Design*, pages –, 2016.

[17] F. Buchegger, B. Jüttler, and A. Mantzaflaris. Adaptively Refined Multi-patch B-splines with Enhanced Smoothness. *Appl. Math. Comput.*, 272(P1):159–172, January 2016.

[18] C. Carstensen. Some remarks on the history and future of averaging techniques in a posteriori finite element error analysis. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 84(1):3–21, 2004.

[19] L. Coox, F. Greco, O. Atak, D. Vandepitte, and W. Desmet. A robust patch coupling method for NURBS-based isogeometric analysis of non-conforming multipatch surfaces. *Computer Methods in Applied Mechanics and Engineering*, pages –, 2016. In press.

[20] J. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, Chichester, 2009.

[21] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196:4160–4183, 2007.

[22] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5296, 2006.

[23] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, and Y. Feng. Polynomial splines over hierarchical T-meshes. *Graphical Models*, 70:76–86, 2008.

[24] T. Dokken, T. Lyche, and K.F. Pettersen. Polynomial splines over locally refined box-partitions. *Computer Aided Geometric Design*, 30(3):331–356, 2013.

[25] M.R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(5-8):264–275, 2010.

[26] W. Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, jun 1996.

[27] A. Falini, J. Špeh, and B. Jüttler. Planar domain parameterization with THB-splines. *Computer Aided Geometric Design*, 35–36, 2015.

[28] D.R. Forsey and R.H. Bartels. Hierarchical B-spline Refinement. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, pages 205–212, New York, NY, USA, 1988. ACM.

[29] C. Giannelli and B. Jüttler. Bases and dimensions of bivariate hierarchical tensor-product splines. *Journal of Computational and Applied Mathematics*, 239:162–178, 2013.

[30] C. Giannelli, B. Jüttler, S.K. Kleiss, A. Mantzaflaris, B. Simeon, and J. Špeh. THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 299:337–365, 2016.

[31] C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: The truncated basis for hierarchical splines. *Computer Aided Geometric Design*, 29(7):485–498, 2012.

[32] C. Giannelli, B. Jüttler, and H. Speleers. Strongly stable bases for adaptively refined multilevel spline spaces. *Advances in Computational Mathematics*, 40(2):459–490, 2014.

[33] J. Gravesen, A. Evgrafov, D.M. Nguyen, and P. Nørtoft. Planar Parametrization in Isogeometric Analysis. In *Mathematical Methods for Curves and Surfaces*, pages 189–212. Springer, 2014.

[34] P. Hennig, M. Kästner, P. Morgenstern, and D. Peterseim. Adaptive mesh refinement strategies in isogeometric analysis - A computational comparison. *Computer Methods in Applied Mechanics and Engineering*, pages –, 2016. In press.

[35] P. Hennig, S. Müller, and M. Kästner. Bézier extraction and adaptive refinement of truncated hierarchical NURBS. *Computer Methods in Applied Mechanics and Engineering*, 305:316–339, 2016.

[36] C. Hesch and P. Betsch. Isogeometric analysis and domain decomposition methods. *Computer Methods in Applied Mechanics and Engineering*, 213-216:104–112, 2012.

[37] C. Hofer and U. Langer. Dual-Primal Isogeometric Tearing and Interconnecting Solvers for large-scale systems of multipatch continuous Galerkin IgA equations. Technical Report NFN Technical Report No. 39, Geometry+Simulation, 2015.

[38] C. Hofer and U. Langer. Dual-primal isogeometric tearing and interconnecting solvers for multipatch dG-IgA equations. *Computer Methods in Applied Mechanics and Engineering*, pages –, 2016. In press.

[39] C. Hofer, U. Langer, and I. Toulopoulos. Discontinuous Galerkin Isogeometric Analysis of Elliptic Problems on Non-matching Interfaces. Technical Report NFN Technical Report No. 38, Geometry+Simulation, 2015.

[40] C. Hofer and I. Toulopoulos. Discontinuous Galerkin Isogeometric Analysis of elliptic problems on segmentations with non-matching interfaces. *Computers & Mathematics with Applications*, 72(7):1811–1827, 2016.

[41] T.J.R. Hughes, J. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195, 2005.

[42] K.A. Johannessen, T. Kvamsdal, and T. Dokken. Isogeometric analysis using LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 269:471–514, 2014.

[43] K.A. Johannessen, F. Remonato, and T. Kvamsdal. On the similarities and differences between Classical Hierarchical, Truncated Hierarchical and LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 291:64–101, 2015.

[44] G. Kiss, C. Giannelli, U. Zore, B. Jüttler, D. Großmann, and J. Barner. Adaptive CAD model (re-)construction with THB-splines. *Graphical Models*, 76(5):273–288, 2014. Geometric Modeling and Processing 2014.

[45] S.K. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. IETI – Isogeometric tearing and interconnecting. *Computer Methods in Applied Mechanics and Engineering*, 247-248(0):201–215, 2012.

[46] R. Kraft. Adaptive and linearly independent multilevel B-splines. In A. Le Méhauté, C. Rabut, and L.L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, pages 209–218. Vanderbilt University Press, Nashville, 1997.

[47] U. Langer, A. Mantzaflaris, S.E. Moore, and I. Toulopoulos. *Multipatch Discontinuous Galerkin Isogeometric Analysis*, pages 1–32. Springer International Publishing, Chabressm, 2015.

[48] X. Li, F.L. Chen, H.M. Kang, and J.S. Deng. A survey on the local refinable splines. *Science China Mathematics*, 59(4):617–644, 2016.

[49] X. Li, J. Deng, and F. Chen. Surface modeling with polynomial splines over hierarchical T-meshes. *The Visual Computer*, 23(12):1027–1033, 2007.

[50] X. Li, J. Zheng, T.W. Sederberg, T.J.R. Hughes, and M.A. Scott. On linear independence of T-spline blending functions. *Computer Aided Geometric Design*, 29(1):63–76, 2012.

[51] P. Morgenstern and D. Peterseim. Analysis-suitable adaptive T-mesh refinement with linear complexity. *Computer Aided Geometric Design*, 34:50–66, 2015.

[52] D.-M. Nguyen, M. Pauley, and B. Jüttler. Isogeometric segmentation: Construction of auxiliary curves. *Computer-Aided Design*, 70:89–99, 2016. {SPM} 2015.

[53] T. Nguyen and B. Jüttler. Parameterization of contractible domains using sequences of harmonic maps. In *Curves and Surfaces*, pages 501–514. Springer, 2012.

[54] N. Nguyen-Thanh, H. Nguyen-Xuan, S.P.A. Bordas, and T. Rabczuk. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering*, 200(21-22):1892–1908, 2011.

[55] M. Pauley, D.-M. Nguyen, D. Mayer, J. Špeh, O. Weeger, and B. Jüttler. *The Isogeometric Segmentation Pipeline*, pages 51–72. Springer International Publishing, Cham, 2015.

[56] S.I. Repin. *A Posteriori Estimates for Partial Differential Equations*. Walter de Gruyter, Berlin, Germany, 2008.

[57] M.A. Scott, X. Li, T.W. Sederberg, and T.J.R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213-216:206–222, 2012.

[58] M.A. Scott, D.C. Thomas, and E.J. Evans. Isogeometric spline forests. *Computer Methods in Applied Mechanics and Engineering*, 269:222–264, 2014.

[59] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng, and T. Lyche. T-spline simplification and local refinement. *ACM Transactions on Graphics*, 23(3):276–283, aug 2004.

[60] T.W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCs. *ACM Transactions on Graphics*, 22(3):477–484, 2003.

[61] V. Skytt, O. Barrowclough, and T. Dokken. Locally refined spline surfaces for representation of terrain data. *Computers & Graphics*, 49:58–68, 2015.

[62] İ. Temizer and C. Hesch. Hierarchical NURBS in frictionless contact. *Computer Methods in Applied Mechanics and Engineering*, 299:161–186, 2016.

[63] İ. Temizer, P. Wriggers, and T.J.R. Hughes. Contact treatment in isogeometric analysis with NURBS. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):1100–1112, 2011.

[64] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 200(49-52):3554–3567, 2011.

[65] P. Wang, J. Xu, J. Deng, and F. Chen. Adaptive isogeometric analysis using rational PHT-splines. *Computer-Aided Design*, 43(11):1438–1448, 2011.

[66] X. Wei, Y.J. Zhang, T.J.R. Hughes, and M.A. Scott. Extended Truncated Hierarchical Catmull-Clark Subdivision. *Computer Methods in Applied Mechanics and Engineering*, 299:316–336, 2016.

[67] M. Wu, J. Xu, R. Wang, and Z. Yang. Hierarchical bases of spline spaces with highest order smoothness over hierarchical T-subdivisions. *Computer Aided Geometric Design*, 29(7):499–509, 2012.

[68] G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Parameterization of computational domain in isogeometric analysis: Methods and comparison. *Computer Methods in Applied Mechanics and Engineering*, 200(23-24):2021–2031, 2011.

[69] Y. Zhang, Y. Bazilevs, S. Goswami, C.L. Bajaj, and T.J.R. Hughes. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196:2943–2959, 2007.