Patchwork B-Spline Refinement

Nora Engleitner[a], Bert Jüttler[b]

[a] *MTU Aero Engines AG, Munich, Germany*
[b] *Institute of Applied Geometry, Johannes Kepler University, Linz, Austria*

Abstract

Hierarchical splines allow to use representations with varying level of detail in different parts of a geometric model. However, the progression from coarse to fine scale is based on a single sequence of *nested* spline spaces. More precisely, each space defining a representation of some level must simultaneously be a subspace of all the higher level spaces and contain all the lower level ones. This requirement imposes severe restrictions on the available refinement strategies. We introduce the new framework of Patchwork B-splines (PB-splines), which alleviates these constraints and therefore increases the flexibility of the representations that are available in different parts of a geometric model. We derive the mathematical foundations of multivariate PB-splines, in particular focusing on the construction of a basis that forms a convex partition of unity. This generalizes the concept of truncated hierarchical (TH) B-splines to the novel framework. Moreover, we discuss the application of PB-splines to surface reconstruction with adaptive refinement. It is observed that the increased flexibility of the local representations provides significant advantages.

Keywords: adaptive spline refinement, hierarchical splines, truncation, surface approximation, multivariate splines

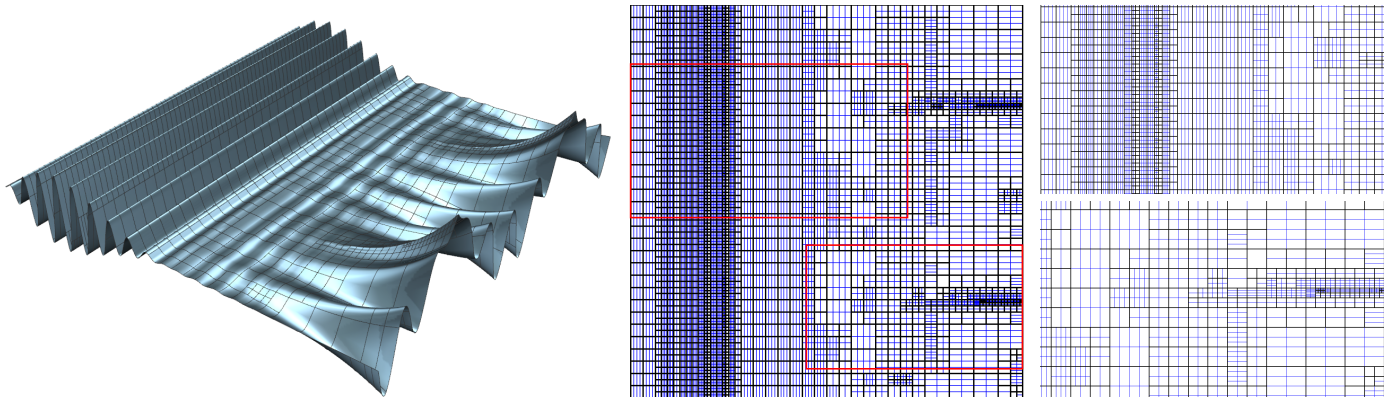

Figure 1: Surface approximation using PB-splines (left). Vertical refinement dominates in the domain's western part (center and top right), while mostly horizontal refinement is used in the eastern one (center and bottom right).

1. Introduction

Due to its computational advantages, the use of tensor–product spaces has become one of the most popular approaches to generate multivariate splines. The resulting representations have become a de-facto standard for the description of free-form surfaces in Computer-Aided Design [6]. More recently, they have also been adopted for discretizing partial differential equations in the framework of isogeometric analysis [14], which aims at bridging the gap between geometry design and numerical simulation by employing the same class of functions for describing the geometry of an object and for representing the fields describing physical phenomena.

As a limitation, the use of tensor-product splines essentially rules out the possibility of performing local refinement, since the insertion of a knot creates a knot line (or, more generally, a knot hyperplane) that extends throughout the entire domain. Generalizations of tensor-product splines that provide the possibility of local refinement are therefore of vital interest, both for design and analysis.

Their investigation started with the work of Forsey and Bartels [9] on hierarchical B-spline refinement, which was later revised and extended by Kraft [19]. T-splines, which were introduced by Sederberg et al. [25], address the local refinement challenge by generating surfaces from control meshes with T-junctions. Additional constraints are required to guarantee linear independence [22], which is an essential prerequisite for using T-splines in analysis. The algebraically complete spaces of polynomial spline functions on hierarchical T-meshes were studied in numerous publications, and found to be useful for design and analysis [21, 17]. Finally, Dokken et al. addressed the adaptive refinement challenge by introducing the framework of locally refined splines [3].

The hierarchical approach to adaptive spline refinement has lately been revisited, resulting in the introduction of truncated hierarchical B-splines [11]. Besides form-

ing a convex partition of unity, they provide good stability properties, algebraic completeness under mild assumptions on the refinement procedure, and optimal approximation power [12, 23, 27]. Additionally, several generalizations have been proposed, including hierarchical versions of Powell-Sabin splines [26], triangular splines [30], box splines [31, 15], B-splines on triangulations [16], T-splines [5], and of subdivision spline functions [29, 32]. Various applications of hierarchical tensor-product splines have been presented: These include surface fitting [13, 18] and isogeometric analysis [1, 2, 20, 24], while applications and implementation aspects of truncated hierarchical B-splines were studied in [10].

For hierarchical B-splines, the progression from coarse to fine scale relies on a single sequence of nested tensor-product spline spaces, where each element is a refinement of its predecessor by knot insertion and/or degree elevation. This requirement restricts the choice of refinement strategies. For instance, in the example shown in Fig. 1, it is advantageous to combine vertical refinement in the western part of the domain with horizontal refinement in the eastern part, but this cannot be achieved by the existing construction. See also Fig. 8 on page 10 and Fig. 9 on page 11 for a comparison with hierarchical B-splines.

In order to increase the flexibility, we introduce the more general framework of Patchwork B-splines. In an earlier conference publication [4], we investigated bivariate splines of uniform degree and maximum smoothness. The present paper focuses on the general multivariate patchwork spline space, which combines splines of arbitrary degree and smoothness. We present the construction of a basis for this space, which relies on a suitable selection mechanism for tensor-product B-splines. Furthermore, in order to obtain a non-negative partition of unity, we establish a new truncation mechanism. It is different from the one described in [4], which did not allow us to prove the non-negativity of the resulting truncated B-splines. We also complement the presented theory by algorithms and experimental results for the automatic generation of feasible patchwork hierarchies and for adaptive surface approximation.

The remainder of the paper is organized as follows: First, we set up our framework and define the patchwork spline space and its basis in the next three sections. We then proceed to the characterization result (Theorem 3 in Section 6), which is derived under suitable assumptions and with the help of local spline spaces satisfying homogeneous boundary conditions, which are analyzed before. The truncation mechanism, which is established in Section 8, requires us to introduce the concept of tail hierarchies in the preceding section. The last part of the paper focuses on algorithms and the application of Patchwork B-splines to adaptive surface approximation.

2. Preliminaries: Tensor-product splines

Throughout this paper we consider a finite sequence of spline spaces $V^\ell = \operatorname{span} B^\ell$, $\ell = 1, \dots, N$, which are defined on the d–dimensional unit cube $[0,1]^d$. Each space is spanned by tensor-product B-splines $B^\ell = (\beta_j^\ell)_{j \in \mathcal{J}^\ell}$, the elements of which are enumerated by an associated index set $\mathcal{J}^\ell$. The upper index ℓ will be called the *level*.

The spline bases B^ℓ consist of tensor-product B-splines that are defined by d open knot vectors with boundary knots 0 and 1. For each spline space, we choose a polynomial degree $\mathbf{p}^\ell = (p_1^\ell, \dots, p_d^\ell)$ and use only inner knots with multiplicity not exceeding p_i^ℓ in the i-th coordinate. All boundary knots possess multiplicity $p_i^\ell + 1$. The supports of the basis functions are axis-aligned boxes in $[0,1]^d$.

The indices of the spaces V^ℓ define a natural total ordering of the sequence. The subspace relation between the spline spaces restricts it to a *partial ordering*, as follows: The level k is said to *precede* the level ℓ, $k \prec \ell$, if k is less than ℓ and V^k is a subspace of V^ℓ.

The order of smoothness of the functions in the spline spaces V^ℓ depends on the level ℓ and the location $\mathbf{x} \in (0,1)^d$. The order of smoothness at $\mathbf{x} = (x_1, \dots, x_d)$ is equal to

$$s_i^\ell(\mathbf{x}) = p_i^\ell - m_i^\ell(x_i),$$

where $m_i^\ell(x_i)$ denotes the multiplicity of x_i in the knot vector of direction i that has been used to define the tensor-product spline space V^ℓ. In particular, we have $m_i^\ell(x_i) = 0$ if x_i is not present in that knot vector. In fact, the function is even C^∞ if $m_i^\ell(x_i) = 0$ and thus $s_i^\ell(x_i) = p_i^\ell$, but we shall not consider derivatives of an order exceeding the maximum polynomial degree.

More precisely, given a point $\mathbf{x} \in (0,1)^d$, a function $f \in V^\ell$ possesses continuous partial derivatives

$$\partial_{\mathbf{i}} f, \quad \mathbf{0} \le \mathbf{i} \le \mathbf{s}^\ell(\mathbf{x}), \tag{1}$$

in a certain neighborhood of $\mathbf{x}$. Here we use the partial derivative operators

$$(\partial_{\mathbf{i}} f)(\boldsymbol{\xi}) = \frac{\partial^{i_1}}{\partial \xi_1^{i_1}} \cdots \frac{\partial^{i_d}}{\partial \xi_d^{i_d}} f(\xi_1, \dots, \xi_d), \tag{2}$$

where $\mathbf{i} = (i_1, \dots, i_d)$. The following first example will illustrate the discussion of notions and results.

Example 1. We construct the multivariate spline spaces by considering tensor-products of univariate ones. Our construction is based on the univariate spline spaces $\mathcal{S}^p_{\mathbf{m}}$ of degree p with $(p+1)$-fold boundary knots $0, 1$ and dyadically refined inner knots with knot multiplicities $\mathbf{m} = [m^{(r)}]_{r=1,\dots,R}$. In detail, the knots of $\mathcal{S}^p_{\mathbf{m}}$ take the values $\kappa^i = \frac{i}{2^R}$, $i = 1, \dots, 2^R - 1$, and their multiplicities are equal to $m^{(r)}$ if they are created in the r-th step of the dyadic refinement, i.e., for $r = \min\{\ell : 2^\ell \kappa^i \in \mathbb{Z}\}$. For instance, $\mathcal{S}^2_{[1,2]}$ has the knots

$$(0, 0, 0, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{2}, \tfrac{3}{4}, \tfrac{3}{4}, 1, 1, 1).$$

In particular, for dimension $d = 2$, we will use the spaces

$$D^{(p_1,p_2)}_{(\mathbf{m}_1,\mathbf{m}_2)} = \mathcal{S}^{p_1}_{\mathbf{m}_1} \otimes \mathcal{S}^{p_2}_{\mathbf{m}_2}.$$

More precisely, we select the sequence

$$V^1 = D^{(1,2)}_{([1,1,1],[1,1,1])} \qquad V^2 = D^{(1,2)}_{([1,1,1,1],[1,1,1,2])}$$
$$V^3 = D^{(3,2)}_{([1,1,1,2,1],[1,1,1,2])} \qquad V^4 = D^{(3,2)}_{([3,3,3,3],[1,1,1,2])}$$
$$V^5 = D^{(3,2)}_{([3,3,3,3,2],[1,1,1,2])} \qquad V^6 = D^{(3,3)}_{([3,3,3,3,3,2],[2,2,2,3,1])}$$
$$V^7 = D^{(3,3)}_{([3,3,3,3,2,2],[2,2,2,3,1])}$$

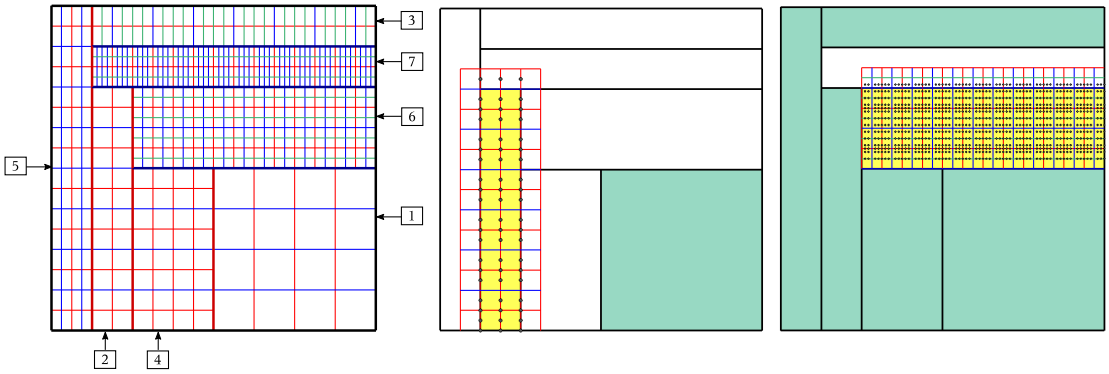

Figure 2: Left: Subdivision of the domain into patches with associated spline spaces. The line color visualizes the smoothness: red, blue and green lines correspond to C^0, C^1, and C^2-smoothness, respectively. Center and right for levels 2 and 6, respectively: Selected PB-splines (visualized by the black points, which are the Greville points), shadows (represented by the knot line grid of the selected PB-splines) and patches of lower levels (green).

which defines the partial ordering

$$\begin{array}{c} 1 \prec 2 \prec 4 \\ 3 \end{array} \Big\rangle 5 \prec 6 \prec 7.$$

◊

3. The patchwork spline space

Besides the sequence of spline spaces, we choose an associated sequence $(\pi^\ell)_{\ell=1,\dots,N}$ of *patches*, which are mutually disjoint open subsets of $(0,1)^d$. The closure of their union is the entire unit cube, defining the domain

$$\Omega = \operatorname{int}\left(\bigcup_{\ell=1}^{N} \overline{\pi^\ell} \right) = (0,1)^d$$

of the patchwork spline space. In principle, it is also possible to consider patches covering only a subset of the cube. In fact, this is the case for the R-tail domains introduced in Section 7, and all the results apply to these domains also. For the sake of simplicity we will now consider the entire cube.

Certain parts of the boundary $\partial\pi^\ell$ of each patch may be shared with the boundaries of other patches. We define the *constraining boundary* of level ℓ,

$$\Gamma^\ell = \bigcup_{k=1}^{\ell-1} \overline{\pi^k} \cap \overline{\pi^\ell},$$

as those parts of the boundary $\partial\pi^\ell$ that are shared with patches of *lower* levels. Note that the constraining boundary may be empty. In particular, this is the case for $\ell = 1$.

Example 1 (Continued)**.** Fig. 2 (left) shows a subdivision of the domain into seven patches that we will use to illustrate the construction of bivariate Patchwork B-splines. The knot lines of the corresponding spline spaces define a mesh on each patch. ◊

Definition. The *patchwork spline space* $P(\Delta)$ on an open subset $\Delta \subseteq \Omega$ consists of functions $f \in \mathcal{C}(\Delta)$ that possess the following two properties:

(i) The restrictions of f to the *restricted* patches $\pi^\ell \cap \Delta$ belong to the associated spline spaces V^ℓ,

$$f|_{\pi^\ell \cap \Delta} \in V^\ell|_{\pi^\ell \cap \Delta}, \quad \ell = 1, \dots, N.$$

(ii) The function $f|_{\overline{(\pi^\ell \cup \pi^k) \cap \Delta}}$ has smoothness $\mathbf{s}^{\max(\ell,k)}$ at any point $\mathbf{x}$ on the intersection of the boundaries of the two restricted patches $\pi^\ell \cap \Delta$ and $\pi^k \cap \Delta$.

In particular we obtain the *full patchwork spline space* $P(\Omega)$ when considering $\Delta = \Omega$. ◊

Clearly, the patchwork spline space contains at least all tensor-product polynomials of degree

$$\min_{\ell=1,\dots,N} \mathbf{p}^\ell = (\min_{\ell=1,\dots,N} p_1^\ell, \dots, \min_{\ell=1,\dots,N} p_d^\ell),$$

restricted to the domain Δ.

Example 2. We consider function from the patchwork spline space, $f \in P(\Omega)$, on the four patches shown in Fig. 3 with the associated biquadratic spline spaces

$$V^1 = V^2 = D^{(2,2)}_{([1,1],[1,1])} \text{ and } V^3 = V^4 = D^{(2,2)}_{([2,2],[2,2])}.$$

Note that the pictures in Fig. 3 show only the knot lines in the interior of the patches, while the blue lines represent their boundaries (which are also knot lines of multiplicities 1 and 2). The second property of the patchwork spline space $P(\Omega)$ implies that $f|_{\overline{(\pi^1 \cup \pi^2) \cap \Omega}}$ is $C^{(1,1)}$ at $\mathbf{x} = (\frac{1}{2}, \frac{1}{2})$, while $f|_{\overline{(\pi^3 \cup \pi^4) \cap \Omega}}$ is only $C^{(0,0)}$ at this point. Considering the restriction to patches 1 and 3, we notice that f is $C^{(0,2)}$ in $\mathbf{x} = (\frac{1}{2}, \frac{5}{8})$ and $C^{(0,0)}$ in $\mathbf{x} = (\frac{1}{2}, \frac{3}{4})$. ◊

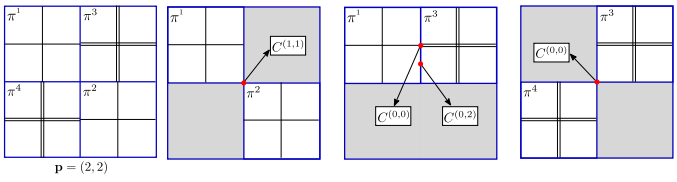

Figure 3: Example 2 – Smoothness of $f \in P(\Omega)$ at various points.

4. Patchwork B-splines

We generalize Kraft's selection mechanism [19] to obtain a basis for the patchwork spline space. Each basis B^ℓ contributes those tensor-product B-splines β_j^ℓ that do

not vanish on the patch π^ℓ but which do vanish on the constraining boundary Γ^ℓ, i.e.,

$$\mathcal{K}^\ell = \{j \in \mathcal{J}^\ell : \beta_j^\ell|_{\pi^\ell} \neq 0 \quad \text{and} \quad \beta_j^\ell|_{\Gamma^\ell} = 0\}. \tag{3}$$

The selected B-splines of all levels form the *Patchwork B-splines* (PB-splines)

$$K = \{\beta_j^\ell : \ell = 1\ldots, N;\ j \in \mathcal{K}^\ell\}. \tag{4}$$

In addition, the union of all supports of selected level ℓ B-splines is said to be the *shadow* of the patch π^ℓ,

$$\hat{\pi}^\ell = \bigcup_{j\in\mathcal{K}^\ell} \mathrm{supp}\beta_j^\ell.$$

Example 1 (Continued)**.** Figure 2 (center and right) visualizes the shadows and selected basis functions for the patches 2 and 6 of our example. ◊

Throughout the paper we will introduce additional assumptions (summarized in Fig. 4) that guarantee certain properties of the PB-splines. The first assumption is needed for proving linear independence of PB-splines, and for characterizing the space spanned by them:

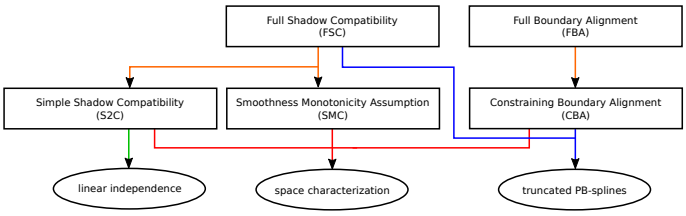

Figure 4: Relations between assumptions and PB-spline properties.

Assumption. The patches and spline spaces possess the property of *Simple*[1] *Shadow Compatibility:* If the shadow $\hat{\pi}^\ell$ of a patch π^ℓ intersects a patch π^k with level k, then the first level is either equal to the second one, or the first level precedes the second one,

$$\hat{\pi}^\ell \cap \pi^k \neq \emptyset \quad \Rightarrow \quad \ell = k \quad \text{or} \quad \ell \prec k. \tag{S2C}$$

Thus, apart from its native patch, each shadow $\hat{\pi}^\ell$ intersects only patches of higher levels and with finer spline spaces. We immediately obtain our first result:

Theorem 1. *The Patchwork B-splines are linearly independent on Ω if S2C holds.*

The proof of linear independence is based on an idea originally formulated in [19]. We keep the paper self-contained by presenting its adaptation to PB-splines.

Proof. Consider a linear combination of the null function

$$0 = \sum_{j\in\mathcal{K}^1} c_j^1\beta_j^1(\mathbf{x}) + \sum_{j\in\mathcal{K}^2} c_j^2\beta_j^2(\mathbf{x}) + \ldots + \sum_{j\in\mathcal{K}^N} c_j^N\beta_j^N(\mathbf{x}).$$

Due to S2C only the level 1 B-splines appearing in the first sum take nonzero values for $\mathbf{x} \in \pi^1$. Due to their linear independence on π^1 we conclude that the associated coefficients c_j^1 all vanish, hence the first sum is not present. We repeat this argument until all levels are exhausted. □

[1] A more restrictive version will be introduced in Section 7 when discussing tail hierarchies.

In addition to the linear independence of the PB-splines on Ω, S2C also implies that *the space spanned by them is contained in the full patchwork spline space* $P(\Omega)$, i.e.

$$\mathrm{span}\, K \subseteq P(\Omega).$$

This fact follows from the definition of the latter space.

5. Homogeneous boundary conditions

Two additional assumptions will be needed to derive a completeness result, which shows that the span of the PB-splines is indeed the full patchwork spline space. The first one, which introduces a condition on the location of the constraining boundaries, will be analyzed in this section.

Assumption. The constraining boundaries Γ^ℓ of the patches π^ℓ are aligned with the knot hyperplanes of the corresponding spline spaces V^ℓ for all levels, i.e.,

$$\Gamma^\ell \subset \Xi^\ell, \tag{CBA}$$

where $\Xi^\ell \subset \Omega$ is the union of the knot hyperplanes of the level ℓ spline space. This assumption will be called the condition of *Constraining Boundary Alignment* (CBA).

As a consequence, at least one coordinate x_i of each point $\mathbf{x}$ on the constraining boundary Γ^ℓ is present in the associated knot vector that defines B^ℓ, i.e., it is one of the knots in the i-th coordinate direction. Assumption CBA generalizes the "weak condition" on the patch boundaries, which has been introduced for hierarchical splines [28].

Next we introduce certain boundary conditions for each patch. For any point $\mathbf{x} \in \partial\pi^\ell$, we consider the partial derivatives with respect to all variables up to the order that is determined by the smoothness of V^ℓ at $\mathbf{x}$,

$$(\vartheta_{\mathbf{x}}^\ell f) = \left[(\partial_{\mathbf{i}} f)(\mathbf{x})\right]_{0\le\mathbf{i}\le\mathbf{s}^\ell(\mathbf{x})}.$$

These derivatives are collected in a tensor of order d and dimension $\mathbf{1}+\mathbf{s}^\ell(\mathbf{x})$, where $\mathbf{1} = (1,\ldots,1) \in \mathbb{R}^d$. A function f is said to satisfy the *homogeneous boundary conditions* with respect to the patch π^ℓ if this tensor vanishes at all points of the constraining boundary, i.e., if

$$(\vartheta_{\mathbf{x}}^\ell f) = \mathbf{0} \text{ for all } \mathbf{x} \in \Gamma^\ell.$$

All spline functions $f \in V^\ell$ that satisfy this condition form the *constrained spline space* V_0^ℓ on the patch π^ℓ.

Lemma 2. *The selected B-splines $\{\beta_j^\ell|_{\pi^\ell} : j \in \mathcal{K}^\ell\}$ of level ℓ form a basis of the constrained spline space $V_0^\ell|_{\pi^\ell}$ on the patch π^ℓ if assumption CBA is satisfied.*

The proof is postponed to the appendix.

6. PB-splines span the patchwork spline space

The second assumption, which is needed for the completeness result, guarantees the compatibility of the smoothness for neighboring patches. For any point $\mathbf{x} \in \Gamma^\ell$ on the constraining boundary, we say that the i-th coordinate direction is *transversal* with respect to Γ^ℓ if the intersection of a sufficiently short line segment, with midpoint $\mathbf{x}$ in this direction, and the constraining boundary

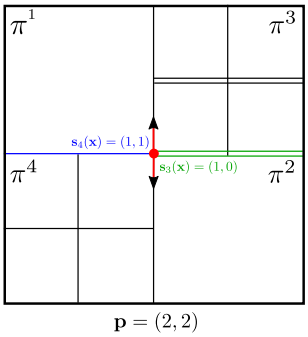

Figure 5: Example 3 – Violation of SMC.

is a single point. More precisely, this is the case if there exists $\varepsilon > 0$ such that

$$(\mathbf{x} + (-\varepsilon, +\varepsilon)\vec{\mathbf{e}}_i) \cap \Gamma^\ell$$

is a single point, where the unit vector $\vec{\mathbf{e}}_i$ spans the i-th coordinate axis. Intuitively speaking, the non-transversal directions are locally aligned with the constraining boundary, while the transversal ones are not.

Assumption. All patches and associated spline spaces fulfill the *Smoothness Monotonicity Condition* (SMC): The smoothness at any point $\mathbf{x}$ on the constraining boundary of a patch π^ℓ in all transversal directions i does not increase with the level. More precisely,

$$s_i^\ell(\mathbf{x}) \le s_i^k(\mathbf{x}) \qquad \text{for all} \quad \mathbf{x} \in \Gamma^\ell \cap \pi^k, \tag{SMC}$$

if $k < \ell$ and the i-th coordinate direction is transversal with respect to Γ^ℓ at $\mathbf{x}$.

In other words, for any two neighboring patches we assume that the smoothness of the higher level patch in all transversal directions does not exceed that of the other patch. SMC is always satisfied if we use uniform (i.e., level-independent) degrees $\mathbf{p}^\ell = (p_1, \dots, p_d)$ and knot multiplicities for all spline spaces. It is also implied by the assumption of Full Shadow Compatibility (FSC), which will be introduced in Section 7.

Example 3. We consider four patches $\pi^1, \dots, \pi^4$ obtained by dyadic subdivision of the domain $[0,1]^2$ and their associated biquadratic spline spaces

$$V^1 = V^2 = D^{(2,2)}_{([1],[1])},\ V^3 = D^{(2,2)}_{([1,1],[2,2])},\ V^4 = D^{(2,2)}_{([1,1],[1,1])},$$

see Fig. 5. SMC is violated at the point $\mathbf{x} = (\frac{1}{2}, \frac{1}{2})$ of the constraining boundary Γ^4 since $s_2^4(\mathbf{x}) = 1 > s_2^3(\mathbf{x}) = 0$. $\lozenge$

Now we characterize the span of the PB-splines:

Theorem 3. *The Patchwork B-splines span the full patchwork spline space $P(\Omega)$ if the assumptions S2C, CBA and SMC are all satisfied.*

Proof. We consider the union of the first L patches

$$\Delta_L = \operatorname{int}\Big(\bigcup_{\ell=1}^{L} \overline{\pi^\ell}\Big) \subseteq \Omega, \quad L = 1, \dots, N, \tag{5}$$

which form the *L-head domain*. We use induction with respect to L in order to prove

$$\operatorname{span} \bigcup_{\ell=1}^{N} \{\beta_j^\ell : j \in \mathcal{K}^\ell\} = P(\Delta_L), \quad \text{for } L = 1, \dots, N. \tag{6}$$

This implies the desired characterization since $\Delta_N = \Omega$.

Eq. (6) is satisfied for $L = 1$ as $\Delta_1 = \pi^1$ and $\Gamma^1 = \emptyset$, hence $\operatorname{span} B^1|_{\pi^1} = \operatorname{span} K^1|_{\pi^1}$. Thus, any $f \in P(\Delta_1)$ admits the representation

$$f(\mathbf{x}) = \sum_{j \in \mathcal{K}^\ell} c_j^1 \beta_j^1(\mathbf{x}), \quad \mathbf{x} \in \Delta_1.$$

Now we assume that (6) holds for L and show that the statement is satisfied for $L+1$ as well. Consider a function $f \in P(\Delta_{L+1})$. The definition of the patchwork spline space implies $P(\Delta_{L+1})|_{\Delta_L} \subseteq P(\Delta_L)$, hence $f|_{\Delta_L} \in P(\Delta_L)$. Thus, we can use the induction hypothesis to obtain a local representation,

$$f(\mathbf{x}) = \sum_{\ell=1}^{L} \sum_{j \in \mathcal{K}^\ell} c_j^\ell \beta_j^\ell(\mathbf{x}), \quad \mathbf{x} \in \Delta_L, \tag{7}$$

with respect to the L-head domain. We use the coefficients c_j^ℓ in (7) to define the auxiliary function

$$g(\mathbf{x}) = f(\mathbf{x}) - \sum_{\ell=1}^{L} \sum_{j \in \mathcal{K}^\ell} c_j^\ell \beta_j^\ell(\mathbf{x}), \quad \mathbf{x} \in \Delta_{L+1} \tag{8}$$

on the level $L+1$-head domain. One may confirm that $\vartheta_{\mathbf{x}}^{L+1} g$ exists for all $\mathbf{x} \in \Gamma^{L+1}$, i.e., g has continuous derivatives up to order $\mathbf{s}^{L+1}(\mathbf{x})$. On the one hand, this is satisfied for f by definition of the patchwork spline space $P(\Delta^{L+1})$. On the other hand, for the Patchwork B-splines β_j^ℓ in (8), this follows from Assumption SMC for the derivatives in any transversal coordinate direction with respect to Γ^{L+1} and it is obtained for the non-transversal coordinate directions by differentiating along the constraining boundary.

By definition we have $g|_{\Delta^L} = 0$. Consequently, the auxiliary function g satisfies homogeneous boundary conditions $(\vartheta^{L+1} g)(\mathbf{x}) = \mathbf{0}$ for all $\mathbf{x} \in \Gamma^{L+1}$. Moreover, the definition of $P(\Delta_{L+1})$ and assumption S2C imply $g|_{\pi^{L+1}} \in V^{L+1}|_{\pi^{L+1}}$. Lemma 2 thus ensures that there exists a representation

$$g(\mathbf{x}) = \sum_{j \in \mathcal{K}^{L+1}} c_j^{L+1} \beta_j^{L+1}(\mathbf{x}), \quad \mathbf{x} \in \pi^{L+1}. \tag{9}$$

With S2C and $g|_{\Delta_L} = 0$ it follows that this is valid for all $\mathbf{x} \in \Delta_{L+1}$. Thus, we may substitute (9) into (8) and solve for f. This results in a linear combination of PB-splines of the first $L+1$ levels,

$$f(\mathbf{x}) = \sum_{\ell=1}^{L} \sum_{j \in \mathcal{K}^\ell} c_j^\ell \beta_j^\ell(\mathbf{x}) + \sum_{j \in \mathcal{K}^{L+1}} c_j^{L+1} \beta_j^{L+1}(\mathbf{x}), \quad \mathbf{x} \in \Delta_{L+1},$$

which concludes the proof by induction. $\square$

According to this result, the space spanned by the PB-splines is actually independent of the ordering of the patches, since it fully determined by the patches and the spline spaces associated with them. However, only orderings that satisfy the assumptions of the Theorem can be considered. Additionally, this result also allows to identify patchwork hierarchies that lead to *nested* patchwork spline spaces: Keeping or enlarging all spline spaces while keeping the patches leads to a patchwork spline space that contains the original one.

7. Tail Hierarchies

Before introducing a truncation mechanism, we have to establish a bottom-up definition of the PB-splines. For this purpose we consider the *R-tail domains*

$$\Delta^R = \operatorname{int}\Big(\bigcup_{\ell=R}^{N} \overline{\pi^\ell}\Big) \subseteq \Omega, \quad R = 1, \dots, N.$$

Clearly we have that $\Delta^N = \pi^N$ and $\Delta^1 = \Omega$. These domains are associated with the *R-tail* $(V^\ell)_{\ell=R,\dots,N}$ of the spline space hierarchy and with the corresponding patches $(\pi^\ell)_{\ell=R,\dots,N}$. We define PB-splines on these domains as follows: Each patch $\pi^\ell \in \Delta^R$ is equipped with an *R-tail constraining boundary*

$$\Gamma^{\ell,R} = \overline{\pi^\ell} \cap \bigcup_{k=R}^{\ell-1} \overline{\pi^k}, \quad \ell = R, \dots, N. \tag{10}$$

Note that $\Gamma^{\ell,R} \subseteq \Gamma^\ell$, $\Gamma^{\ell,\ell} = \emptyset$ and $\Gamma^{\ell,1} = \Gamma^\ell$. The *R-tail PB-splines*

$$K^R = \bigcup_{\ell=R}^{N} \{\beta_j^\ell : j \in \mathcal{K}^{\ell,R}\}$$

are obtained by suitably adapting the selection mechanism (3) to the R-tail hierarchy,

$$\mathcal{K}^{\ell,R} = \{j \in \mathcal{J}^\ell : \beta_j^\ell|_{\pi^\ell} \neq 0 \quad \text{and} \quad \beta_j^\ell|_{\Gamma^{\ell,R}} = 0\},$$

for $\ell = R, \dots, N$. Note that $\mathcal{K}^{\ell,R} \supseteq \mathcal{K}^\ell$ and $\mathcal{K}^{\ell,1} = \mathcal{K}^\ell$. These selected B-splines define the *R-tail shadow* of a patch $\pi^\ell \subseteq \Delta^R$,

$$\hat{\pi}^{\ell,R} = \bigcup_{j \in \mathcal{K}^{\ell,R}} \operatorname{supp}\beta_j^\ell.$$

In order to apply the results of the previous sections to the R-tail hierarchy, we introduce another assumption, which is similar to S2C:

Assumption. The R-tails possess *Full Shadow Compatibility* (FSC): If the R-tail shadow $\hat{\pi}^{\ell,R}$ of a patch $\pi^\ell \subseteq \Delta^R$ intersects a patch $\pi^k \subseteq \Delta^R$ with level k, then the first level is either equal to the second one, or the first level precedes the second one,

$$\hat{\pi}^{\ell,R} \cap \pi^k \neq \emptyset \quad \Rightarrow \quad \ell = k \quad \text{or} \quad \ell \prec k, \tag{FSC}$$

for $\ell, k = R, \dots, N$ and $R = 1, \dots, N$.

Note that *FSC implies S2C and SMC*. Indeed, S2C follows from the relation $\hat{\pi}^{\ell,1} = \hat{\pi}^\ell \subseteq \hat{\pi}^{\ell,R}$. Furthermore, we know that the R-tail shadow $\hat{\pi}^{\ell,\ell}$ of a patch π^ℓ intersects all neighboring patches of higher levels. Consequently, all spline spaces of neighboring patches are nested if FSC is satisfied. This implies SMC.

Assumption FSC enables us to extend the results of the previous sections to the R-tail PB-splines.

Corollary 4. *The R-tail PB-splines K^R are linearly independent on Δ^R if FSC is satisfied. They span the partially nested hierarchical spline space $P(\Delta^R)$ if additionally the assumptions CBA and SMC are fulfilled.*

Proof. S2C is satisfied on each subset Δ^R if FSC holds. Consequently we can prove the two parts of the corollary by applying Theorem 1 and Theorem 3 (with a suitable renumbering) to the R-tail hierarchy. Indeed, the assumptions CBA and SMC are automatically satisfied for these hierarchies as well. □

For future reference we note the following result.

Lemma 5. *We assume that FSC is satisfied. The R-tail PB-splines admit the backward recursive definition*

$$K^R = \{\beta_j^R : j \in \mathcal{K}^{R,R}\} \cup \operatorname{Sel}^R(K^{R+1}), \tag{11}$$

with respect to decreasing levels $R = N, \dots, 1$, that is based on the selection operators, with the initial value $K^{N+1} = \emptyset$,

$$\operatorname{Sel}^R(S) = \{\varphi \in S : \operatorname{supp}\varphi \cap \pi^R = \emptyset\}. \tag{12}$$

Proof. We prove the identity

$$\{\beta_j^\ell : j \in \mathcal{K}^{\ell,R}\} = \operatorname{Sel}^R(\cdots \operatorname{Sel}^{\ell-1}(\{\beta_j^\ell : j \in \mathcal{K}^{\ell,\ell}\}) \cdots) \tag{13}$$

which implies the desired result. We note that

$$\begin{aligned} &\varphi \in \operatorname{Sel}^R(\cdots \operatorname{Sel}^{\ell-1}(S) \cdots) \\ &\Leftrightarrow (\varphi \in S \text{ and } \varphi|_{\pi^k} = 0 \; \forall k = R, \dots, \ell-1). \end{aligned} \tag{14}$$

On the one hand, consider a B-spline β_j^ℓ with $j \in \mathcal{K}^{\ell,R}$. Clearly, its index satisfies $j \in \mathcal{K}^{\ell,\ell}$. Also, due to FSC,

$$\operatorname{supp}\beta_j^\ell \cap \pi^k = \emptyset, \quad k = R, \dots, \ell-1,$$

hence all selection operators $\operatorname{Sel}^R \dots, \operatorname{Sel}^{\ell-1}$ pick this function. On the other hand, take a B-spline β_j^ℓ that belongs to the set on the right-hand side in (13). Clearly, $\beta_j^\ell|_{\pi^k} = 0$ implies $\beta_j^\ell|_{\overline{\pi}^k \cap \overline{\pi}^\ell} = 0$ for $k = R, \dots, \ell-1$. Rewriting the definition (10) of the R-tail constraining boundary

$$\Gamma^{\ell,R} = \bigcup_{k=R}^{\ell-1} \overline{\pi}^\ell \cap \overline{\pi}^k \tag{15}$$

confirms that the B-spline β_j^ℓ belongs to the set on the left-hand side in (13). □

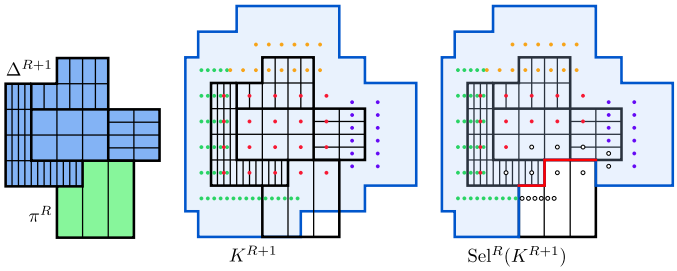

Figure 6: Example 4 – Tail hierarchy (left), associated PB-splines (center) and selection operator (right).

Example 4. We consider an $(R+1)$-tail domain consisting of four patches $\Delta^{R+1} = \bigcup_{\ell=N-3}^{N} \pi^\ell$ and an additional patch $\pi^R = \pi^{N-4}$, see Fig. 6. The left picture shows the $(R+1)$-tail domain (blue) and the additional patch (green). In the center, we visualize the combined supports of the corresponding $(R+1)$-tail PB-splines K^{R+1} (light blue) and the Greville points of the basis functions, where the four different colors correspond to the different levels. Finally, we depict the supports of the *selected* basis functions $\mathrm{Sel}^R(K^{R+1})$ as the light blue region in the right picture. The hollow dots identify the Greville points of the *non-selected* basis-functions. ◊

8. Truncation

We modify the recursive definition (11) of PB-splines and introduce the *truncated PB-splines* (TPB-splines). These functions are defined recursively, starting at the highest level N.

The TPB-splines of level N are simply the PB-splines,

$$\beta_i^N = \tau_i^N \quad i \in \mathcal{K}^{N,N},$$

and they form the set $T^N = \{\tau_i^N : i \in \mathcal{K}^{N,N}\}$.

We now proceed to lower levels. When arriving at some level R, we use the selection operator Sel^R, defined in (12), to split the set of TPB-splines T^{R+1} into the two disjoint subsets

$$\mathrm{Sel}^R(T^{R+1}) \quad \text{and} \quad \mathrm{nSel}^R(T^{R+1}) = T^{R+1} \setminus \mathrm{Sel}^R(T^{R+1}).$$

The first set contributes directly to the level R TPB-splines

$$T^R = \mathrm{Sel}^R(T^{R+1}) \cup \{\tau_j^R : j \in \mathcal{K}^{R,R}\}. \tag{16}$$

It is complemented by adding the set of truncated B-splines τ_j^R, which are derived from the PB-splines β_j^R with indices $j \in \mathcal{K}^{R,R}$, as follows:

The restriction of any B-spline β_j^R, $j \in \mathcal{K}^{R,R}$ to the $(R+1)$-tail domain Δ^{R+1} can be represented as a linear combination of level $R+1$ truncated PB-splines,

$$\beta_j^R(\mathbf{x}) = \sum_{\tau_i^\ell \in T^{R+1}} b_{i,j}^{\ell,R} \tau_i^\ell(\mathbf{x}), \quad \mathbf{x} \in \Delta^{R+1}, \tag{17}$$

with uniquely defined coefficients $b_{i,j}^{\ell,R}$. These coefficients are used to define the truncated B-splines of level R,

$$\tau_j^R(\mathbf{x}) = \beta_j^R(\mathbf{x}) - \sum_{\tau_i^\ell \in \mathrm{Sel}^R(T^{R+1})} b_{i,j}^{\ell,R} \tau_i^\ell(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d. \tag{18}$$

Note that these functions also possess the piecewise representation

$$\tau_j^R(\mathbf{x}) = \begin{cases} \beta_j^R(\mathbf{x}) & \mathbf{x} \in \pi^R \\ \displaystyle\sum_{\tau_i^\ell \in \mathrm{nSel}^R(T^{R+1})} b_{i,j}^{\ell,R} \tau_i^\ell(\mathbf{x}) & \mathbf{x} \in \Delta^{R+1}, \end{cases} \tag{19}$$

for $\mathbf{x} \in \Delta^R$.

Similarly to the previous section we define index sets for the truncated PB-splines of level R, which are derived from PB-splines of level ℓ,

$$\mathcal{T}^{\ell,R} = \{j \in \mathcal{J}^\ell : \tau_j^\ell \in T^R\}, \text{ hence } T^R = \bigcup_{\ell=R}^{N} \{\tau_j^\ell : j \in \mathcal{T}^{\ell,R}\}.$$

In order to prove that this construction is well-defined, i.e., that the local representation (17) exists and is unique, we will show that the truncated PB-splines of level ℓ are a basis of the patchwork spline space $P(\Delta^\ell)$.

Recall [12] that the representation of a function

$$f(\mathbf{x}) = \sum_{\ell=R}^{N} \sum_{i \in \mathcal{T}^{\ell,R}} d_i^\ell \tau_i^\ell(\mathbf{x}), \quad \mathbf{x} \in \Delta^R,$$

with respect to the system of functions T^R *preserves the coefficients* of the local tensor-product representations

$$f(\mathbf{x}) = \sum_{i \in \mathcal{K}^{\ell,\ell}} c_i^\ell \beta_i^\ell(\mathbf{x}), \quad \mathbf{x} \in \pi^\ell, \ \ell = R, \dots, N, \tag{20}$$

on the patches π^ℓ if $d_i^\ell = c_i^\ell$ holds for all $i \in \mathcal{T}^{\ell,R}$.

Theorem 6. *The equations* (16)–(17) *define non-negative and linearly independent truncated PB-splines T^R that span (a superspace of) the patchwork spline space $P(\Delta^R)$ and preserve the coefficients of the local tensor-product representations of any function $f \in P(\Delta^R)$ if FSC, CBA and SMC are satisfied.*

The proof of the theorem is given in the appendix. In particular, preservation of coefficients implies that the TPB-splines form a partition of unity. While the theorem ensures that $P(\Delta^R)$ is contained in the span of the TPB-splines, it can be even shown that the two spaces are even identical, but we postpone this to a future publication.

9. Generating feasible patchwork hierarchies

A collection of patches and associated spline spaces will be called a *feasible patchwork hierarchy* if the corresponding patchwork spline space $P(\Omega)$ satisfies FSC and CBA. Consequently, this space can be equipped with a basis of (T)PB-splines.

The algorithm is based on an infinite but countable catalog $\hat{\mathcal{V}} = (\hat{V}^k)_{k=0,1,\dots}$ of candidate spline spaces

$$\hat{V}^k = \mathrm{span}\hat{B}^k, \quad k = 0, 1, \dots,$$

which are defined on the d-dimensional unit cube. Each space is spanned by a tensor-product spline basis

$$\hat{B}^k = (\hat{\beta}_j^k)_{j \in \hat{\mathcal{J}}^k},$$

the elements of which are enumerated by an associated index set $\hat{\mathcal{J}}^k$. The index k of the catalog defines a natural ordering, but it is not required that the spaces are nested.

Algorithm 1 Generate a feasible patchwork hierarchy

Input: N polytopes σ and indices $s(\sigma)$, a catalog $\hat{\mathcal{V}}$
Output: $\pi^1, \dots, \pi^N$ with $V^1, \dots, V^N$

```
1  for ℓ = 1 to N do
2      k_min ← |V̂| + 1
3      for all polytopes σ ∈ Σ do
4          k ← s(σ);
5          while not[FEASIBLE(k, ℓ, σ) and V^{s(σ)} ⊆ V^k] do
6              k ← k + 1
7          end while
8          if k < k_min then
9              k_min ← k; σ_min ← σ
10         end if
11     end for
12     π^ℓ ← σ_min; V^ℓ ← V̂^{k_min}
13     Delete the selected σ_min from Σ.
14 end for
```

Example 5. In the bivariate case ($d = 2$), for given uniform degrees $\mathbf{p} = (p, p)$, we consider dyadically refined spline spaces with maximum smoothness,

$$M^p_{q,r} = \mathcal{S}^p_{\underbrace{[1,\dots,1]}_{q \text{ times}}} \otimes S^p_{\underbrace{[1,\dots,1]}_{r \text{ times}}}.$$

We restrict ourselves to spaces of maximum refinement level ν and impose a bound ϱ on the difference of the refinement levels, i.e., and $\max(q, r) \le \nu$ and $|q - r| \le \varrho$. The latter bound controls the aspect ratio of the elements. We define the catalog as

$$\hat{\mathcal{V}} = (\hat{V}_s)_s = (M^p_{0,0}, M^p_{1,0}, M^p_{0,1}, M^p_{2,0}, M^p_{1,1}, M^p_{0,2}, \dots, M^p_{\nu,\nu}),$$

where we order the spaces lexicographically with respect to $(q + r, q)$. Fig. 7 shows an instance of the catalog. ◊

Algorithm 2 Feasibility check

```
1  function FEASIBLE(k, ℓ, σ)
2      for all i < ℓ with π̂^{i,i} ∩ σ ≠ ∅ do
3          if V^i ⊈ V̂^k then return false end if
4      end for
5      for all j ∈ Ĵ^k with supp β̂^k_j ∩ σ ≠ ∅ do
6          safe ← true
7          for all i < ℓ with supp β̂^k_j ∩ π^i ≠ ∅ do
8              safe ← false
9              for i' = i to ℓ − 1 do
10                 if supp β̂^k_j ∩ σ̄ ∩ π^{i'} ≠ ∅ then
11                     safe ← true; break
12                 end if
13             end for
14             if safe == false then return false end if
15         end for
16     end for
17     return true
18 end function
```

The *input* of Algorithm 1 consists of a finite subdivision Σ of the domain into $N = |\Sigma|$ polytopes σ (which will become the patches), with associated initial spline spaces $\hat{V}^{s(\sigma)}$ from the catalog. We assume that the final space in the catalog contains all other spaces in order to guarantee that the algorithm succeeds in all cases. Moreover, it is required that the entire boundary $\partial\sigma$ of each polytope is contained in the union of the knot hyperplanes $\Xi^{s(\sigma)}$ of the associated initial spline space. This will be denoted as *Full Boundary Alignment (FBA)*. In particular, it implies that the polytopes have axis-aligned sides[2].

The *output* is a sequence of patches π^ℓ, covering the domain Ω, along with associated spline spaces V^ℓ from the given catalog. Each patch is one of the given polytopes and the spline spaces V^ℓ satisfy

$$V^\ell \supseteq \hat{V}^{s(\sigma)} \text{ if } \pi^\ell = \sigma. \tag{21}$$

Consequently, the space associated with a patch is a superspace of the associated initial space. While this already implies CBA, the algorithm additionally ensures that FSC is satisfied.

The index ℓ in the main loop of Algorithm 1 indicates the level of the patch that we want to add next to the sequence of to patches. In order to determine the polytope that will become the patch π^ℓ we go through all remaining polytopes in Σ and select the smallest index of a feasible spline space from our catalog $\hat{\mathcal{V}}$ for each polytope.

Algorithm 2 analyzes the feasibility of a spline space for a polytope σ (see line 5 of Algorithm 1): We first check compatibility with shadows $\hat{\pi}^{i,i}$ for all $i < \ell$, thereby ensuring the R-tail shadow compatibility of the already selected patches π^i for $R = 1, \dots, \ell - 1$. Second, we ensure R-tail shadow compatibility of $\sigma = \pi^\ell$ in lines 5 to 16.

It is guaranteed that Algorithm 1 succeeds in generating a feasible patchwork hierarchy if the final space of the catalog contains all other spaces, which is satisfied for the catalog described in Example 5.

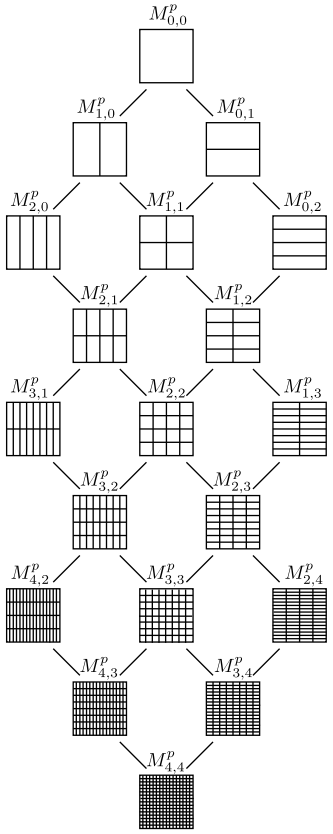

Figure 7: The catalog $\hat{\mathcal{V}}$ from Example 5 with $\varrho = 2$ and $\nu = 4$.

10. Adaptive surface fitting

We approximate point data $f_i \in \mathbb{R}^3$, $i = 0, \dots, m$, with associated parameter values $(u_i, v_i) \in [0, 1]^2$. The patchwork hierarchy is initialized with a small number of patches obtained by n-adic subdivision and relatively coarse spline spaces with uniform knots. We alternate fitting and refinement steps until the maximum error is below the user-defined threshold ε or the maximum number of iterations is reached.

[2]For d=2, they are polygons with horizontal and vertical edges.

Fitting step. We perform standard regularized least-squares fitting as described, e.g., in [18], for generating (T)PB-spline surfaces.

Refinement step. The adaptive refinement is guided by the threshold ε and uses a Mark–Subdivide–Refine method:

- *Mark*: Patches containing points with an error exceeding the threshold ε are marked for refinement. If a point belongs to several patches, we mark the one with the lowest level.
- *Subdivide*: The marked patches are subdivided into smaller ones by using n-adic subdivision.
- *Refine*: The initial spline space for each newly generated patch is such that it contains the space of the original patch and satisfies FBA. Additionally, if a newly generated patch contains points with an error exceeding the threshold ε, we refine the initial spline space further by applying n-adic knot refinement either in u- or v-direction. The direction is chosen using a local fitting approach. Finally, we apply Algorithm 1 to obtain a feasible patchwork hierarchy again.

Note that PB-splines and TPB-splines defined by the same patchwork hierarchy give equivalent results, since the spaces spanned by them are identical.

Example 6. We used the adaptive fitting algorithm for generating the surface and the corresponding PB-spline mesh in Fig. 1. We sampled 258×265 data points on a non-uniform grid from the function

$$\begin{aligned} f(u,v) = 0.1(&B_{0,7}(u)\,(\sin(120u)\sin(2\pi u)) \\ &+B_{1,7}(u)\,(2\sin(120u)\sin(2\pi u)) \\ &+B_{7,7}(u)\,(2-2(1+0.4\sin(60v))|\cos(2\pi v)|)), \end{aligned}$$

with $B_{i,7}(u)$ being the i–th Bernstein polynomial of degree 7. The values of the function f vary between -0.07 and 0.2. We create an initial patchwork hierarchy that consists of four patches, obtained from dyadic subdivision of the domain, and corresponding initial spline spaces $V = D^{(2,2)}_{([1,1,1],[1,1,1])}$. Applying the adaptive mesh refinement algorithm with $\varepsilon = 5e{-}4$ to the initial patchwork hierarchy results in a feasible patchwork hierarchy with 7858 degrees of freedom, see Fig. 1, center and right. For the corresponding fitting result we obtain a maximum error of $4.95e{-}4$ and an average error of $1.03e{-}4$. In order to visualize the surface with a commercial tool, we had to subdivide the PB-spline patches to classical tensor-product spline patches. The resulting surface can be seen in Fig. 1 on the left, where the black lines indicate the boundaries of the B-spline patches. ◊

We present a second example and use it to compare the approximation results for different spline constructions: PB-splines, hierarchical B-splines and tensor-product splines.

Example 7. The function we considered in this example is constructed in a similar manner as in Example 6, however, we use univariate Bernstein polynomials in both directions and a different coefficient matrix. We sampled $90,000$ data points and for generating the PB-spline mesh we applied the adaptive refinement algorithm with $\varepsilon = 1e{-}3$ to the same initial patchwork hierarchy as before. The HB-spline mesh is also obtained by automatic adaptive refinement, whereas the tensor-product spline mesh has been manually defined by imitating the behavior of the function to prevent uniform refinement across the entire domain. Fig. 8 visualizes the approximating surface (for PB-splines only, since all results look very similar), as well as the adaptive PB-spline mesh, the adaptive HB-spline mesh and the manually generated tensor-product spline mesh. Table 1 reports the numbers of degrees of freedom and some error statistics. It is observed that by using PB-splines we could not only save a significant percentage of degrees of freedom, but we also obtained a slightly better fitting result. ◊

	no. of dof	% of dof	maximum error	average error
tensor-product splines	2916	265 %	3.08e-3	1.5e-4
HB-splines	1860	169 %	3.08e-3	1.42e-4
PB-splines	1102	100 %	1.08e-3	1.32e-4

Table 1: Comparison of the results for Example 7.

Finally, we apply the surface approximation algorithm to real data from a turbine blade of an aircraft engine and compare again the results of PB-splines, truncated hierarchical B-splines and tensor-product splines.

Example 8. We considered $210,479$ data points on the airfoil of an aircraft engine, where the corresponding parameter values are obtained by Floater's parameterization method, see [8] and [7].

All three methods started from the initial spline space $V = D^{(2,2)}_{([1,1,1],[1,1])}$ where we have roughly twice as many knots in u- than in v-direction. The patchwork hierarchy evolved from a single patch covering the entire unit square. We applied the adaptive fitting algorithm with $\varepsilon = 1e{-}6$ and stopped when the maximum error was reasonably small and approximately the same for all three results. This resulted in 7 refinement steps for PB-splines, 6 steps for THB-splines and 5 steps for tensor-product splines.

Table 2 compares the number of degrees of freedom with respect to the PB-splines, the maximum error, the average error and the percentage of points with an error smaller or equal to ε for the three methods. The results emphasize once more the advantage of using adaptive spline constructions. When comparing PB- and THB-splines we notice that we could still save some degrees of freedom by using the PB-splines, although the difference is not as significant as in Example 7.

Fig. 9 shows the resulting PB-spline mesh together with close-up views of the PB- and the THB-spline mesh. We also present the result of a reflection line analysis of the PB-spline surface. Additionally, we intersected the different approximation surfaces with a plane, which is indicated by the red rectangle around the blade, and zoomed into the

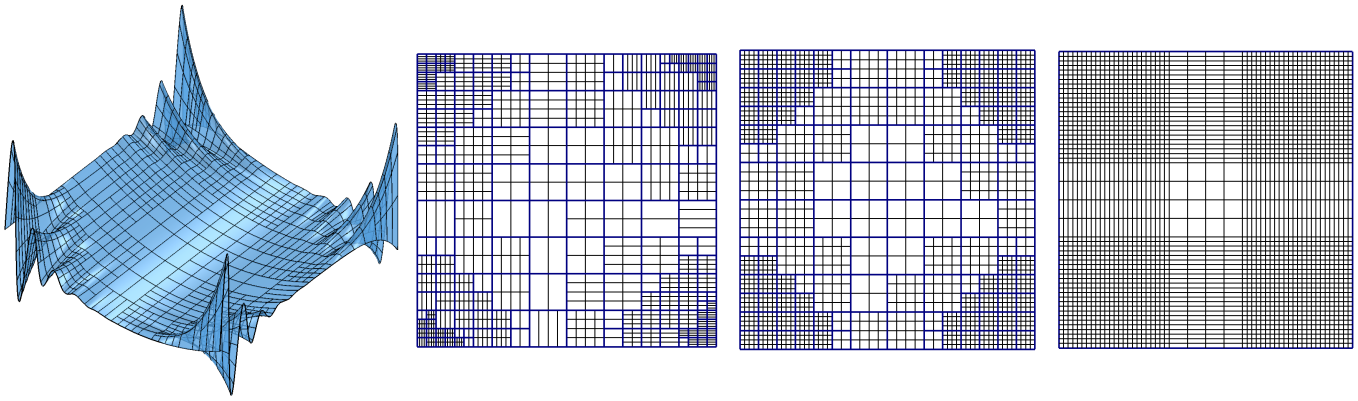

Figure 8: From left to right: surface obtained for Example 7, automatically generated mesh (patches and knot lines) for PB-splines, automatically generated mesh for HB-splines and user-defined mesh for tensor-product splines.

	no. of dof	% of dof	maximum error	average error	% of data below ε
tensor-product splines	33540	278%	5.49e-5	1.60e-6	89.2%
THB-splines	13682	114%	3.11e-5	5.82e-7	92.1%
PB-splines	12050	100%	5.79e-5	4.05e-7	94.5%

Table 2: Comparison of results for Example 8.

region near the leading edge, using a frontal view. The intersection curves are depicted in Fig. 9 on the right, where the red, green and blue curve correspond to PB-, THB-splines and tensor-product splines, respectively. The gray curve represents a stripe of the original triangulated point data that was obtained by cutting it with two parallel planes.

These intersection curves reveal significant differences between the three surfaces. While both the THB- and the tensor-product spline surface oscillate in the vicinity of the leading edge, the PB-spline surface follows the triangulated data quite well. As a possible explanation we note that – even if THB- and PB-splines use a similar amount of degrees of freedom – the PB-spline construction seems to allocate them more efficiently to the areas where they are required. ◊

11. Conclusion

We generalized the concept of hierarchical spline constructions by introducing (T)PB-splines. In particular, this new framework increases the flexibility of the available refinement strategies. For instance, it allows the insertion of vertical vs. horizontal knot lines in different areas of the bivariate (parameter) domain. If the resulting patches and spline spaces form a feasible patchwork hierarchy, then the corresponding (T)PB-splines are a basis of the patchwork spline space $P(\Omega)$. The independent refinement strategies lead to a reduced number of degrees of freedom compared to hierarchical B-splines or classical tensor-product splines. Moreover, when performing data fitting we were able to improve the quality of the approximation surface by using PB-splines.

Future work will complete the presented theory by investigating the algebraic completeness and by further characterizing the space spanned by the TPB-splines. For industrial applications it is necessary to provide an efficient implementation and therefore, we will elaborate on data structures and algorithms in greater detail. Furthermore, error estimators for guiding anisotropic refinement can be considered. Subsequent research topics might include the application of the presented construction to isogeometric analysis and the use of (T)PB-splines for geometric design, as well as a detailed comparison of PB- and truncated PB-splines with respect to their numerical properties, analogous to the comparison between HB- and THB-splines in [10].

Acknowledgment

Supported by project NFN S117 “Geometry + Simulation” of the Austrian Science Fund and the EC projects “MOTOR”, GA no. 678727 and “CHANGE”, GA no. 694515. The authors would like to thank Dr. U. Zore and Dr. D. Grossmann.

Appendix

Proof of Lemma 2. The constraining boundary $\Gamma^\ell = \bigcup_{\varphi\in\mathcal{F}^\ell}\varphi$ consists of facets φ from a certain facet set $\mathcal{F}^\ell$, due to CBA. Each facet φ has some dimension $\delta\in\{1,\dots,d-1\}$. After a suitable reordering of the coordinates, it takes the form

$$\varphi = [\kappa^1_{j^1},\kappa^1_{j^1+1}]\times\cdots\times[\kappa^\delta_{j^\delta},\kappa^\delta_{j^\delta+1}]\times\{\kappa^{\delta+1}_{j^{\delta+1}}\}\times\cdots\times\{\kappa^d_{j^d}\}$$

for certain indices $j^1,\dots,j^d$, with $\kappa^i_{j^k}$ being the j^k-th knot of the knot vector in direction i of the space V^ℓ.

First, we show that the selected B-splines belong to the constrained spline space. For any index $j\in\mathcal{K}^\ell$ it follows from the definition of $\mathcal{K}^\ell$ that the B-spline β^ℓ_j vanishes on the constraining boundary. Thus it satisfies the homogeneous boundary conditions with respect to the patch π^ℓ since the B-spline has smoothness $\mathbf{s}^\ell(\mathbf{x})$.

Second, we show that we need only selected B-splines for representing the restriction $f|_{\pi^\ell}$ of any function from the constrained spline space V^ℓ_0. Clearly, any $f\in V^\ell_0\subseteq V^\ell$ possesses a representation

$$f(\mathbf{x}) = \sum_{\substack{j\in\mathcal{J}^\ell\\ \operatorname{supp}\beta^\ell_j\cap\pi^\ell\neq\emptyset}} c^\ell_j\beta^\ell_j(\mathbf{x}),\quad \mathbf{x}\in\pi^\ell, \tag{22}$$

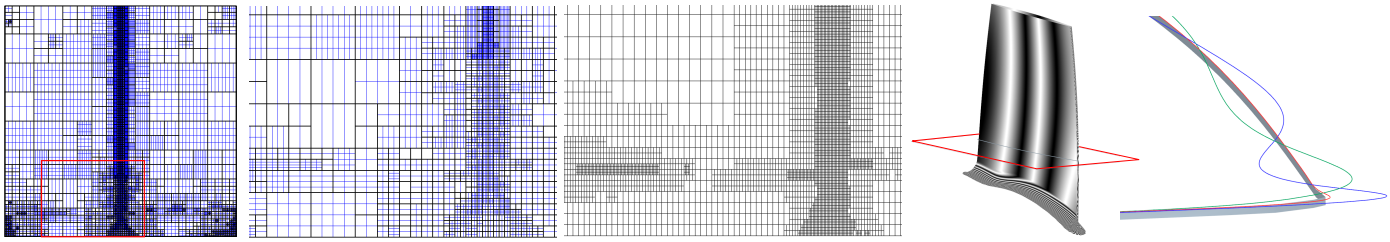

Figure 9: Approximation of airfoil data (Example 8). From left to right: The automatically generated PB-spline mesh, details of the PB- and THB-spline mesh, the PB-spline surface (entire blade with reflection lines), intersection lines for PB- (red), THB- (green) and tensor-product splines (blue) together with a stripe of the triangulated data (gray).

with certain real coefficients c_j^ℓ. Consider a B-spline β_k^ℓ with an index $k \in \mathcal{J}^\ell \setminus \mathcal{K}^\ell$ and $\mathrm{supp}\beta_k^\ell \cap \pi^\ell \neq \emptyset$. Consequently, there exists a facet $\varphi \in \mathcal{F}^\ell$ such that $\beta_k^\ell|_\varphi \neq 0$. Without loss of generality we assume that the dimension of φ is maximal, i.e., it is chosen such that it is not contained in any other facet $\varphi' \in \mathcal{F}^\ell$. The knot vectors of the tensor-product spline space V^ℓ induce a subdivision of $(0,1)^d$ into cells (axis-aligned boxes). There exist cells $\zeta \subseteq \overline{\pi^\ell}$ and $\zeta' \subseteq \overline{(0,1)^d \setminus \pi^\ell}$ which intersect in φ, i.e., $\varphi = \zeta \cap \zeta'$, since the dimension of φ is maximal.

The functions $f|_\zeta$ and $0|_{\zeta'}$ have contact on φ as described in Definition 2.3 of [23], as f satisfies homogeneous boundary conditions on $\varphi \subseteq \Gamma^\ell$. We consider the local representations of these two functions on the cells ζ and ζ',

$$f(\mathbf{x}) = \sum_{\substack{j \in \mathcal{J}^\ell \\ \mathrm{supp}\beta_j^\ell \cap \zeta \neq \emptyset}} c_j^\ell \beta_j^\ell(\mathbf{x}), \ \mathbf{x} \in \zeta, \ \text{and } 0 = \sum_{\substack{j \in \mathcal{J}^\ell \\ \mathrm{supp}\beta_j^\ell \cap \zeta' \neq \emptyset}} 0 \ \beta_j^\ell(\mathbf{x}'), \ \mathbf{x}' \in \zeta'.$$

The index sets of both sums include k as the B-spline β_k^ℓ takes non-zero values on $\varphi = \zeta \cap \zeta'$. We use the Contact Characterization Lemma [23, Lemma 2.4] to conclude that $c_k^\ell = 0$. Thus, only B-splines β_j^ℓ with $j \in \mathcal{K}^\ell$ contribute to the representation in (22), as the remaining terms in the sum have zero coefficients. Finally, we note that the selected B-splines form a basis as B-splines possess the property of local linear independence (i.e., they are linearly independent on any open subset). □

Proof of Theorem 6. We use the induction principle for decreasing values of the level ℓ, starting at $\ell = N$, until we arrive at $\ell = R$. For each level ℓ we shall prove the following six facts:

(H0) T^ℓ is well defined by equations (16)-(17).

(H1) T^ℓ is linearly independent on Δ^ℓ.

(H2) $\mathrm{span}T^\ell \supseteq P(\Delta^\ell) = \mathrm{span}K^\ell$.

(H3) The functions in T^ℓ are non-negative.

(H4) We have $\mathrm{supp}\tau_i^k \subseteq \mathrm{supp}\beta_i^k$ for all $\tau_i^\ell \in T^\ell$.

(H5) The representations of functions with respect to T^ℓ preserve the coefficients of the local tensor-product representations of any function $f \in P(\Delta^\ell)$.

All statements are true for $\ell = N$ since $T^N = K^N$. In the induction step we assume that they are valid for $\ell = L+1$ and prove that the statements are true for $\ell = L$ as well.

Proof of (H0). The local representation

$$\beta_i^L(\mathbf{x}) = \sum_{\tau_j^\ell \in T^{L+1}} b_{j,i}^{\ell,L} \tau_j^\ell(\mathbf{x}), \quad \mathbf{x} \in \Delta^{L+1},$$

exists, since $\mathrm{span}T^{L+1} = \mathrm{span}K^{L+1}$ by the induction hypothesis (H2) and $\beta_i^L \in P(\Delta^{L+1})$ for $i \in \mathcal{K}^{L,L}$ due to FSC and the definition of $P(\Delta^{L+1})$. Therefore, T^L is well defined by equations (16)-(17).

Proof of (H1). We consider the representation

$$0 = \sum_{i \in \mathcal{K}^{L,L}} c_i^L \tau_i^L(\mathbf{x}) + \sum_{\tau_j^\ell \in \mathrm{Sel}^L(T^{L+1})} c_j^\ell \tau_j^\ell(\mathbf{x}), \quad \mathbf{x} \in \Delta^L,$$

of the null function. When restricted to patch π^L we get

$$0 = \sum_{i \in \mathcal{K}^{L,L}} c_i^L \beta_i^L(\mathbf{x}), \quad \mathbf{x} \in \pi^L,$$

since the functions in $\mathrm{Sel}^L(T^{L+1})$ vanish on π^L, according to the definition of the selection mechanism. Therefore, the coefficients c_i^L are all zero since the functions β_i^L for $i \in \mathcal{K}^{L,L}$ are non-zero and linearly independent on π^L. The remaining functions $\tau_j^\ell \in \mathrm{Sel}^L(T^{L+1})$ are linearly independent on Δ^{L+1} according to the induction hypothesis (H1), hence the coefficients satisfy $c_j^\ell = 0$ as $\Delta^{L+1} \subseteq \Delta^L$.

Proof of (H2). Recall that

$$K^L = \{\beta_i^L : i \in \mathcal{K}^{L,L}\} \cup \mathrm{Sel}^L(K^{L+1}),$$

according to Lemma 5. We prove that functions from both subsets of K^L admit a representation with respect to T^L. This is clear for β_i^L with $i \in \mathcal{K}^{L,L}$, since solving (18) for β_i^L leads to

$$\beta_i^L = \tau_i^L + \sum_{\tau_j^\ell \in \mathrm{Sel}^L(T^{L+1})} b_{j,i}^{\ell,L} \tau_j^\ell.$$

The proof for functions $\beta_j^\ell \in \mathrm{Sel}^L(K^{L+1})$ from the second subset requires more work: We know that $\beta_j^\ell \in P(\Delta^{L+1})$ by Corollary 4 and $\mathrm{Sel}^L(K^{L+1}) \subseteq K^{L+1}$, thus we can use the induction hypothesis (H2) and find a representation

$$\beta_j^\ell(\mathbf{x}) = \sum_{\tau_i^m \in T^{L+1}} b_{i,j}^{m,\ell} \tau_i^m(\mathbf{x}), \quad \mathbf{x} \in \Delta^{L+1}.$$

For any non-zero coefficient $b_{i,j}^{m,\ell}$ in this representation, the associated basis functions satisfy $\mathrm{supp}\beta_i^m \subseteq \mathrm{supp}\beta_j^\ell$, due

to the induction hypothesis (H5) and the refinement property of B-splines. We now use the induction hypothesis (H4) to conclude that $\tau_i^m \in \mathrm{Sel}^L(T^{L+1})$ if $b_{i,j}^{m,\ell} \neq 0$, since

$$\mathrm{supp}\tau_i^m \subseteq \mathrm{supp}\beta_i^m \subseteq \mathrm{supp}\beta_j^\ell \text{ and } \beta_j^\ell \in \mathrm{Sel}^L(K^{L+1}).$$

Moreover, the definition (12) of Sel^L implies that β_j^ℓ vanishes on π^L. Combining both observations gives the representation

$$\beta_j^\ell(\mathbf{x}) = \sum_{\tau_i^m \in \mathrm{Sel}^L(T^{L+1})} b_{i,j}^{m,\ell} \tau_i^m(\mathbf{x}), \quad \mathbf{x} \in \Delta^L.$$

Consequently, these functions are contained in $\mathrm{span}(\mathrm{Sel}^L(T^{L+1})) \subset \mathrm{span}T^L$.

Proof of (H3). The induction hypothesis for (H3) implies that all functions $\tau_j^\ell \in \mathrm{Sel}^L(T^{L+1})$ are non-negative. In order to establish this property for τ_i^L we consider the piecewise representation (19). On the one hand, the function τ_i^L is non-negative on π^L since

$$\tau_i^L|_{\pi^L} = \beta_i^L|_{\pi^L} \geq 0.$$

On the other hand, it is a non-negative linear combination of non-negative functions on Δ^{L+1}. Indeed, the functions are non-negative due to the induction hypothesis (H3), and the coefficients are non-negative due to (H5) and the B-spline refinement properties.

Proof of (H4). It is implied by the induction hypothesis (H4) for $k > L$. For $k = L$ it follows immediately by comparing the representations (19) and (17) of τ_i^L and β_i^L, respectively, due to the induction hypotheses (H3) and (H5). Note that the B-spline refinement properties again ensure non-negative coefficients.

Proof of (H5). Finally, we show the property of preservation of coefficients. Therefore, we consider a function $f \in P(\Delta^L)$ which has representations (20) for $R = L$. Due to (H2) we may represent it as

$$\begin{aligned} f(\mathbf{x}) &= \sum_{\ell=L}^{N} \sum_{i \in \mathcal{T}^{\ell,L}} d_i^\ell \tau_i^\ell(\mathbf{x}) \\ &= \sum_{i \in \mathcal{K}^{L,L}} d_i^L \tau_i^L(\mathbf{x}) + \sum_{\tau_i^\ell \in \mathrm{Sel}^L(T^{L+1})} d_i^\ell \tau_i^\ell(\mathbf{x}), \end{aligned} \tag{23}$$

for $\mathbf{x} \in \Delta^L$. First we prove the preservation property for the coefficients of level L. Restricting (23) to the patch π^L gives the sum

$$f(\mathbf{x}) = \sum_{i \in \mathcal{K}^{L,L}} d_i^L \tau_i^L(\mathbf{x}) = \sum_{i \in \mathcal{K}^{L,L}} c_i^L \beta_i^L(\mathbf{x}), \quad \mathbf{x} \in \pi^L,$$

where only functions τ_i^L contribute, according to (19). Comparing this representation with (20) confirms $d_i^L = c_i^L$ for all $i \in \mathcal{K}^{L,L}$.

The preservation property for the coefficients of higher levels remains to be shown. On the one hand, when considering the restriction of f to Δ^{L+1} we obtain that

$$\begin{aligned} f(\mathbf{x}) = &\sum_{i \in \mathcal{K}^{L,L}} \sum_{\tau_j^\ell \in \mathrm{nSel}^L(T^{L+1})} d_i^L b_{j,i}^{\ell,L} \tau_j^\ell(\mathbf{x}) + \\ &\sum_{\tau_j^\ell \in \mathrm{Sel}^L(T^{L+1})} d_j^\ell \tau_j^\ell(\mathbf{x}), \quad \mathbf{x} \in \Delta^{L+1}, \end{aligned} \tag{24}$$

by using the piecewise representation (19) of τ_i^L. On the other hand, we invoke the induction hypothesis (H5) and conclude that the coefficients of the representation

$$f(\mathbf{x}) = \sum_{\tau_i^\ell \in T^{L+1}} \hat{d}_i^\ell \tau_i^\ell(\mathbf{x}), \quad \mathbf{x} \in \Delta^{L+1}, \tag{25}$$

possess the preservation property, i.e. $\hat{d}_i^\ell = c_i^\ell$ for all $i \in \mathcal{K}^{\ell,L+1}$, $\ell = L+1, \ldots, N$. Since T^{L+1} is the disjoint union of $\mathrm{Sel}^L(T^{L+1})$ and $\mathrm{nSel}^L(T^{L+1})$, and due to the linear independence, we compare the coefficients in (24) and (25), which finally confirms that

$$d_i^\ell = \hat{d}_i^\ell = c_i^\ell, \text{ if } \tau_i^\ell \in \mathrm{Sel}^L(T^{L+1})$$

for $\ell = L+1, \ldots, N$. □

References

[1] P. B. Bornemann and F. Cirak. A subdivision-based implementation of the hierarchical B-spline finite element method. *Comput. Meth. Appl. Mech. Engrg.*, 253:584–598, 2013.

[2] A. Buffa and C. Giannelli. Adaptive isogeometric methods with hierarchical splines: error estimator and convergence. *Math. Meth. Appl. Sc.*, 26:1–25, 2016.

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

[4] N. Engleitner, B. Jüttler, and U. Zore. Partially nested hierarchical refinement of bivariate tensor-product splines with highest order smoothness. In: *Mathematical Methods for Curves and Surfaces (Tønsberg 2016)*, Lecture Notes in Computer Science, Springer, in press.

[5] E. J. Evans, M. A. Scott, X. Li, and D. C. Thomas. Hierarchical T-splines: analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Comput. Meth. Appl. Mech. Engrg.*, 284:1–20, 2015.

[6] G. Farin, J. Hoschek, and M.-S. Kim, editors. *Handbook of Computer Aided Geometric Design*. Elsevier, 2002.

[7] Michael S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19 – 27, 2003.

[8] M.S. Floater. Parametrization and smooth approximation of surface triangulations. *Comput. Aided Geom. Design*, 14(3):231–250, 1997.

[9] D. R. Forsey and R. H. Bartels. Hierarchical B-spline refinement. *Comput. Graphics*, 22:205–212, 1988.

[10] C. Giannelli et al. THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Comput. Meth. Appl. Mech. Engrg.*, 299:337–365, 2016.

[11] C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: The truncated basis for hierarchical splines. *Comput. Aided Geom. Design*, 29:485–498, 2012.

[12] C. Giannelli, B. Jüttler, and H. Speleers. Strongly stable bases for adaptively refined multilevel spline spaces. *Adv. Comput. Math.*, 40(2):459–490, 2014.

[13] G. Greiner and K. Hormann. Interpolating and approximating scattered 3D-data with hierarchical tensor product B-splines. In A. Le Méhauté, C. Rabut, and L.L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, pages 163–172. Vanderbilt University Press, Nashville, TN, 1997.

[14] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Meth. Appl. Mech. Engrg.*, 194:4135–4195, 2005.

[15] T. Kanduč, C. Giannelli, F. Pelosi, and H. Speleers. Adaptive isogeometric analysis with hierarchical box splines. *Comput. Meth. Appl. Mech. Engrg.*, 316:817–838, 2017.

[16] H. Kang, F. Chen, and J. Deng. Hierarchical B-splines on regular triangular partitions. *Graph. Models*, 76(5):289–300, 2014.

[17] H. Kang, J. Xu, F. Chen, and J. Deng. A new basis for PHT-splines. *Graphical Models*, 82:149–159, 2015.

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

[19] R. Kraft. *Adaptive und linear unabhängige Multilevel B-Splines und ihre Anwendungen*. PhD thesis, Univ. Stuttgart, 1998.

[20] G. Kuru, C. Verhoosel, K. van der Zeeb, and E.van Brummelen. Goal-adaptive isogeometric analysis with hierarchical splines. *Comput. Meth. Appl. Mech. Engrg.*, 270:270–292, 2014.

[21] X. Li, J. Deng, and F. Chen. Polynomial splines over general T-meshes. *Visual Comput.*, 26:277–286, 2010.

[22] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, and M. A. Scott. On linear independence of T-spline blending functions. *Comput. Aided Geom. Design*, 29:63–76, 2012.

[23] D. Mokriš, B. Jüttler, and C. Giannelli. On the completeness of hierarchical tensor-product B-splines. *J. Comput. Appl. Math.*, 271:53–70, 2014.

[24] D. Schillinger et al. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Comput. Meth. Appl. Mech. Engrg.*, 249–252:116–150, 2012.

[25] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCS. *ACM Trans. Graphics*, 22:477–484, 2003.

[26] H. Speleers, P. Dierckx, and S. Vandewalle. Quasi-hierarchical Powell-Sabin B-splines. *Comput. Aided Geom. Design*, 26:174–191, 2009.

[27] H. Speleers and C. Manni. Effortless quasi-interpolation in hierarchical spaces. *Numerische Mathematik*, 132(1):155–184, 2016.

[28] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput. Meth. Appl. Mech. Engrg.*, 200:3554–3567, 2011.

[29] X. Wei, Y. Zhang, T. J. R. Hughes, and M. A. Scott. Extended truncated hierarchical Catmull-Clark subdivision. *Comput. Meth. Appl. Mech. Engrg.*, 299:316–336, 2016.

[30] A. Yvart, S. Hahmann, and G.-P. Bonneau. Hierarchical triangular splines. *ACM Trans. Graphics*, 24:1374–1391, 2005.

[31] U. Zore and B. Jüttler. Adaptively refined multilevel spline spaces from generating systems. *Comput. Aided Geom. Design*, 31:545–566, 2014.

[32] U. Zore, B. Jüttler, and J. Kosinka. On the linear independence of truncated hierarchical generating systems. *J. Comput. Appl. Math.*, 306:200–216, 2016.