ÖAW
Austrian Academy
of Sciences

Doctoral Program
Computational Mathematics
Numerical Analysis and Symbolic Computation

JKU
JOHANNES KEPLER
UNIVERSITY LINZ

Local (T)HB-spline projectors via restricted hierarchical spline fitting

Alessandro Giust Bert Jüttler Angelos Mantzaflaris

DK-Report No. 2020 4

03 2020

A–4040 LINZ, ALTENBERGERSTRASSE 69, AUSTRIA

Supported by

Austrian Science Fund (FWF)

FWF
Der Wissenschaftsfonds.

Upper Austria

LAND
OBERÖSTERREICH

DK sponsors:

- **Johannes Kepler University Linz** (JKU)
- **Austrian Science Fund** (FWF)
- **Upper Austria**

Local (T)HB-spline projectors via restricted hierarchical spline fitting

Alessandro Giust[a,*], Bert Jüttler[b], Angelos Mantzaflaris[c]

[a] *Doctoral Program Computational Mathematics, Johannes Kepler University, Linz, Austria*
[b] *Institute of Applied Geometry, Johannes Kepler University, Linz, Austria*
[c] *Université Côte d'Azur, Inria Sophia Antipolis - Méditerranée, France*

Abstract

This paper is devoted to techniques for adaptive spline projection via quasi-interpolation, enabling the efficient approximation of given functions. We employ local least-squares fitting in restricted hierarchical spline spaces to establish novel projection operators for hierarchical splines of degree p. This leads to efficient spline projectors that require $\mathcal{O}(p^d)$ floating point operations and $\mathcal{O}(1)$ evaluations of the given function per degree of freedom, while providing essentially the same accuracy as global approximation. Our spline projectors are based on a unifying framework for quasi-interpolation in hierarchical spline spaces. We present a detailed comparison with the scheme of Speleers and Manni (2016).

Keywords: hierarchical splines, truncation, quasi-interpolation, spline fitting

1. Introduction

Geometric modeling relies on the extensive use of spline technology for the description and manipulation of arbitrary shapes (Farin, 2002). More recently, the introduction of Isogeometric Analysis (IgA) by Hughes et al. (2005) proved the effectiveness of splines also in the context of numerical simulation, where it has been noted that this technology leads to a better accuracy per degree of freedom than classical finite element methods.

The standard approach to multivariate spline functions – both in geometric modeling and isogeometric simulation – relies on the use of the tensor-product construction. As a major drawback, this construction prevents any possibility of employing adaptive (rather than uniform) refinement strategies. This fact has triggered intense research on adaptive spline refinement, resulting in a substantial number of publications. These include the rich literature on T-splines that were introduced by Sederberg et al. (2003) as a generalization of NURBS. The problem of missing adaptivity is addressed by control grids with T-junctions. Properties such as linear independence and suitability for IgA were further analyzed by Li et al. (2012). The related approach of polynomial splines over hierarchical T-meshes (PHT-splines) was introduced by Deng et al. (2008) and has subsequently been employed both in geometric modeling and isogeometric analysis. Another approach was proposed by Dokken et al. (2013), resulting in the framework of LR-splines.

We focus our attention on the hierarchical spline refinement originally introduced by Forsey and Bartels (1988). These spline spaces are constructed in terms of a sequence of subdomains representing different refinement levels, and they can be equipped with the basis of hierarchical B-splines (HB-splines) that was introduced and analyzed by Kraft (1998). Even though HB-splines do not form a partition of unity (PoU), several important properties such as linear independence and approximation power were established in Kraft's seminal work. In order to restore the PoU

[*] Corresponding author
Email addresses: alessandro.giust@dk-compmath.jku.at (Alessandro Giust), bert.juettler@jku.at (Bert Jüttler), angelos.mantzaflaris@inria.fr (Angelos Mantzaflaris)

property, HB-splines were modified via the truncation mechanism described by Giannelli et al. (2012), resulting in the construction of truncated hierarchical B-splines (THB-splines). Besides PoU, these splines provide properties such as strong stability (Giannelli et al., 2014), algebraic completeness under certain assumptions (Mokriš et al., 2014), and better conditioning of the resulting system matrices (Giannelli et al., 2016).

Local and global methods for adaptive spline fitting with hierarchical splines have been an active topic of research ever since the pioneering work of Forsey and Bartels (1995). In particular, global fitting methods were investigated by Greiner and Hormann (1997) and subsequently extended and applied to relevant industrial applications (Kiss et al., 2014).

In this work, we focus on adaptive spline fitting via local schemes for quasi-interpolation (QI). These schemes enable the efficient approximation of given functions. Quasi-interpolants for tensor-product splines were extensively analyzed by de Boor and Fix (1973) and Lyche and Schumaker (1975) and subsequently studied by Sablonnière (2005), while the first extension to hierarchical splines was presented already in the aforementioned work of Kraft (1998), based on a telescopic construction.

Lee et al. (2005) were the first that made use of a QI operator to perform multilevel spline surface reconstruction from scattered data sets. More recently, Speleers and Manni (2016) and Speleers (2017) showed how to construct efficient QI schemes for THB-splines. These results led to renewed interest in quasi interpolants and paved the way to different theoretical and practical results. Bracco et al. (2017, 2018) originally proposed an adaptive scattered data fitting for given data sets and subsequently extended the results to suitable industrial applications. The QI schemes found applications to IgA-related issues such as the formulation of reliable error indicators (Buffa and Giannelli, 2017) and the derivation of efficient quadrature rules (Calabró et al., 2018).

The most advanced QI schemes for hierarchical splines were established in two articles by Speleers and Manni (2016) and Speleers (2017). The first one introduces THB-spline projectors that require $\mathcal{O}(p^d)$ floating point operations and $\mathcal{O}(p^d)$ evaluations of the given function per degree of freedom. The authors also note that these projectors are equivalent to Kraft's HB-spline projector under certain conditions. The second article (Speleers, 2017) derives QI schemes that are no spline projectors but require fewer evaluations (only $\mathcal{O}(1)$ per degree of freedom) of the input function, where polynomial reproduction is used to ensure optimal approximation power.

QI schemes with and without the spline projector property were compared in Table 2 of the article by Speleers and Manni (2016), leading the authors of that paper to conclude that "On average, the projector Q^s performs slightly better than the less complicated quasi-interpolant Q^p". In addition, spline projectors for hierarchical splines are useful for generating (T)HB-spline representations of functions that are known to belong to the hierarchical spline space, such as the Jacobian determinant of domain parameterizations. In principle, these representations could be derived by employing rules for derivatives and products of splines, but the use of a QI scheme is more convenient in practice. Clearly, only QI schemes that are spline projectors guarantee the exact reproduction of these functions.

In this paper, we focus on spline projectors and employ local least-squares fitting in restricted hierarchical spline spaces to establish novel QI schemes for hierarchical splines of any degree p. This leads to efficient spline projectors that require $\mathcal{O}(p^d)$ floating point operations and $\mathcal{O}(1)$ evaluations of the given function per degree of freedom, while providing essentially the same accuracy as global approximation. Numerical and theoretical results are presented in order to compare our spline projectors, which are based on a unifying framework for quasi-interpolation in hierarchical spline spaces, with the scheme of Speleers and Manni (2016). In particular, we establish spline projector QI schemes that require only $\mathcal{O}(1)$ evaluations of the given function per degree of freedom.

The remainder of the paper is organized as follows. Section 2 recalls some known concepts related to hierarchical splines and quasi-interpolation operators. In order to prepare the construction of the fitting-based spline projector, Section 3 presents some results relating to hierarchical splines on full and restricted domains. Based on these observations, Section 4 describes the construction, providing also a detailed analysis of the corresponding computational complexity. The resulting adaptive refinement algorithm is discussed in Section 5, along with the derivation of

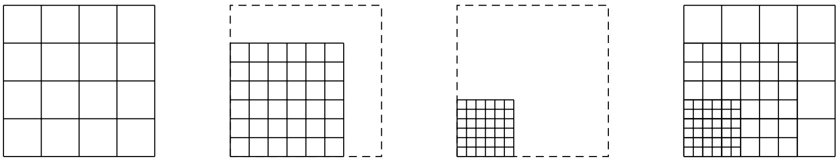

Figure 1: Sequence of subdomains and corresponding hierarchical mesh. From left to right: Ω^0, Ω^1, Ω^2 and final hierarchical mesh

the approximation order of the projector. In Section 6 we describe some experimental tests that illustrate relevant features of our construction. Finally, Section 7 concludes the paper with some remarks and a summary of the achieved results.

2. Preliminaries

We recall the construction of hierarchical spline bases and the use of the truncation mechanism. We also summarize several known results related to quasi-interpolation methods in this context.

2.1. Hierarchical B-splines

In order to define hierarchical B-splines (HB-splines) on a bounded domain $\Omega^0 \subset \mathbb{R}^d$, we consider d-variate *tensor product spline spaces* V^ℓ satisfying

$$V^0 \subset V^1 \subset \cdots \subset V^N,$$

where the upper index $\ell = 0, \ldots, N$ is called the *level*. All spline spaces have the same degree p, and they are defined by d univariate bi-infinite knot sequences $2^{-\ell}\mathbb{Z}$ at level ℓ, one per coordinate direction. The spline spaces are spanned by uniform tensor-product B-splines

$$\mathcal{B}^\ell = \{\beta_i^\ell, i \in \mathbb{Z}^d\}$$

of degree p possessing maximum smoothness $\mathcal{C}^{p-1}$. The restrictions of the spline functions in V^ℓ to *cells of level* ℓ

$$2^{-\ell}(\mathbb{Z}^d + [0,1]^d)$$

are simply tensor-product polynomials of degree p. While the results presented below can be extended to a more general setting (such as splines with non-uniform knot vectors), we try to keep the exposition simple by restricting ourselves to the uniform setting.

In addition, we consider a sequence of *subdomains* $\Omega^\ell \subset \mathbb{R}^d$ satisfying

$$\Omega^0 \supseteq \Omega^1 \supseteq \cdots \supseteq \Omega^N,$$

see Fig. 1. Moreover, it is assumed that $\bar{\Omega}^\ell$ is created by forming the union of finitely many square blocks (which are not required to be mutually disjoint) containing $\lceil \frac{p+1}{2} \rceil$ cells of level $\ell - 1$ per coordinate direction. Consequently, the subdomain Ω^ℓ is a union of the (set-theoretic, i.e., without boundaries) supports of a finite number of B-splines β^ℓ. Note that this includes the assumption that Ω^0 is the union of square blocks of cells of level -1 .

The level ℓ cells contained in

$$\Xi^\ell = \overline{\Omega^\ell \setminus \Omega^{\ell+1}}$$

are said to be *active*. These cells cover Ξ^ℓ, since there are no cells that are only partially refined. The corresponding *hierarchical mesh* (cf. Fig. 1) is formed by all the active cells of all levels.

Hierarchical B-splines (HB-splines) are based on the selection mechanism devised by Kraft (1998): A level ℓ B-spline is selected, i.e., it is included into the hierarchical basis, if its support is contained in the associated domain but not in the next finer one. We obtain the basis

$$\mathcal{H} = \{\beta_i^\ell \ : \ i \in J^\ell, \ \ell = 0, \ldots, N\}$$

with the index sets

$$J^\ell = \{i \in \mathbb{Z}^d \ : \ \operatorname{supp} \beta_i^\ell \subseteq \Omega^\ell \wedge \operatorname{supp} \beta_i^\ell \nsubseteq \Omega^{\ell+1}\} \tag{1}$$

where we set $\Omega^{N+1} = \emptyset$ and we consider supports with respect to the domain Ω^0 (on which we approximate the given function). The selected B-splines are denoted as *active* basis functions. Any active cell of level ℓ belongs to the support of at least one function of that level, since Ω^ℓ is assumed to be the union of supports of level ℓ B-splines.

For future reference we state an assumption that we will need in Theorem 5 (see Section 4.2).

Assumption I (Mesh grading)**.** Only elements of two consecutive levels are present in the support of any selected basis function β_i^ℓ. Consequently, the active cells in the support of any selected basis function

- have level ℓ, or
- have either level ℓ or level $\ell + 1$.

Hierarchical B-splines form a non-negative basis of the *hierarchical spline space*

$$H = \operatorname{span} \mathcal{H} = \{\sigma \in V^N \ : \ \sigma|_{\Omega^0 \setminus \Omega^{\ell+1}} \in V^\ell|_{\Omega^0 \setminus \Omega^{\ell+1}} \ \forall \ell = 0, \ldots, N\}, \tag{2}$$

but they do not form a partition of the unity. This property can be restored by the truncation mechanism of Giannelli et al. (2012). We define the single-level truncation operator $\operatorname{trunc}^\ell : V^\ell \to V^\ell$ as

$$\operatorname{trunc}^\ell(\sigma) = \sum_{i \in \mathbb{Z}^d \setminus J^\ell} c_i^\ell \beta_i^\ell = \sigma - \sum_{i \in J^\ell} c_i^\ell \beta_i^\ell \quad \text{if} \quad \sigma = \sum_{i \in \mathbb{Z}^d} c_i^\ell \beta_i^\ell, \tag{3}$$

and the corresponding multi-level version $\operatorname{Trunc}^\ell : V^\ell \to V^N$

$$\operatorname{Trunc}^\ell(\sum_{i \in \mathbb{Z}^d} c_i^\ell \beta_i^\ell) = \operatorname{trunc}^N \Big(\operatorname{trunc}^{N-1}\big(\ldots(\operatorname{trunc}^\ell(\sum_{i \in \mathbb{Z}^d} c_i^\ell \beta_i^\ell))\ldots\big)\Big) \tag{4}$$

Clearly, in order to apply the single level operators to different levels, the output of each operator has to be represented with respect to the basis of the next level. This can be done easily by the spline refinement rules, due to the nestedness of the spaces V^ℓ.

Truncated hierarchical B-splines (THB-splines) are obtained by applying multi-level truncation (4) to the basis functions of $\mathcal{H}$,

$$\mathcal{T} = \{\tau_i^\ell = \operatorname{Trunc}^{\ell+1}(\beta_i^\ell) \ : \ i \in J^\ell, \ \ell = 0, \ldots, N\} \,.$$

They form another basis, which is a non-negative partition of unity, of the hierarchical spline space.

In addition, they possess the preservation of coefficients (PoC) property: By definition, any function $\sigma \in H$ of the hierarchical spline space has local representations

$$\sigma|_{\Omega^0 \setminus \Omega^{\ell+1}} = \sum_{i \in \mathbb{Z}^d} d_i^\ell \beta_i^\ell|_{\Omega^0 \setminus \Omega^{\ell+1}}, \quad \ell = 1 \ldots, N \tag{5}$$

on each subdomain $\Omega^0 \setminus \Omega^{\ell+1}$. All coefficients of the representation of σ with respect to the truncated basis are simply inherited from these representations,

$$\sigma = \sum_{\ell=0}^{N} \sum_{i \in J^\ell} d_i^\ell \tau_i^\ell \,. \tag{6}$$

In other words, *the coefficient of each THB-spline τ_i^ℓ coincides with the coefficient of its mother B-spline β_i^ℓ in the local representation* (Giannelli et al., 2014).

2.2. Quasi-interpolation in hierarchical spline spaces

We consider linear functionals $\lambda_i^\ell : \mathcal{C}(\mathbb{R}^d) \to \mathbb{R}$, the construction of which will be discussed below. For any index set $J \subseteq \mathbb{Z}^d$ and level ℓ, these define operators

$$Q_J^\ell(f) = \sum_{i \in J} \lambda_i^\ell(f) \beta_i^\ell \tag{7}$$

that transform a continuous function f into a tensor-product spline. In particular, choosing $J = \mathbb{Z}^d$ gives a scheme that generates an approximation in the full tensor-product spline space V^ℓ.

Each value $\lambda_i^\ell(f)$ depends on the values of f in a certain subdomain, which is called the *support* of the linear functional λ_i^ℓ. More precisely, we have that

$$\lambda_i^\ell(f) = 0 \quad \text{if} \quad \operatorname{supp} \lambda_i^\ell \cap \operatorname{supp} f = \emptyset,$$

and $\operatorname{supp} \lambda_i^\ell$ is the smallest set with this property. In order to benefit from the PoC property, linear functionals that satisfy the condition

$$\operatorname{supp} \lambda_i^\ell \subseteq \operatorname{supp} \beta_i^\ell \cap (\Omega^\ell \setminus \Omega^{\ell+1}) \quad \text{if} \quad i \in J^\ell \tag{8}$$

have been considered (Speleers and Manni, 2016). The functionals λ_i^ℓ are said to define *tensor-product spline projectors* if $Q_{\mathbb{Z}^d}^\ell f = f$ for all $f \in V^\ell$. Given functionals λ_i^ℓ with this property, two particular QI schemes for *hierarchical* splines have been studied:

- Kraft (1998) proposed to use the telescopic construction

$$\begin{array}{rcl} K^0 f & = & Q_{J^0}^0 f, \\ K^\ell f & = & K^{\ell-1} f + Q_{J^\ell}^\ell (f - K^{\ell-1} f), \; \ell = 1, \ldots, N \;, \\ K f & = & K^N f \;. \end{array}$$

- Speleers and Manni (2016) exploit the PoC property and define their scheme, which we will call the Speleers-Manni scheme (SMS), in terms of THB-splines,

$$S f = \sum_{\ell=0}^{N} \sum_{i \in J^\ell} \lambda_i^\ell(f) \tau_i^\ell \;.$$

These authors also noted that both schemes are *equivalent*, i.e., they generate the same[1] spline function $Kf = Sf$ for any given f, provided that the one level QI schemes (7) are spline projectors with coefficient functionals that fulfill the condition (8). This condition also ensures that both schemes perform spline projection.

However, both constructions involve the rather strong restriction (8) on the support of the linear functionals λ_i^ℓ. Fig. 2 (left) shows the support of a cubic (T)HB-spline whose support is refined, except for the top left cell. The associated coefficient is determined by the values of the function in that cell. This cell covers only a small portion of the support of the function, and does not even contain its maximum. Nevertheless, the two spline projectors evaluate the associated coefficient solely by examining the value of the given function within that cell.

We employ local fitting (Section 4) based on restricted hierarchical spline spaces (Section 3) for the construction of *spline projectors for the hierarchical space H*. These will take the form

$$P_\beta f = \sum_{\ell=0}^{N} \sum_{i \in J^\ell} \mu_i^\ell(f) \beta_i^\ell \quad \text{and} \quad P_\tau f = \sum_{\ell=0}^{N} \sum_{i \in J^\ell} \nu_i^\ell(f) \tau_i^\ell \tag{9}$$

with linear functionals $\mu_i^\ell(f)$ and $\nu_i^\ell(f)$ for HB- and THB-splines, respectively. For future reference we note that

$$\nu_i^\ell(s) = d_i^\ell \quad \text{for} \quad s \in H \tag{10}$$

where d_i^ℓ is the coefficient appearing in the associated local representation (5), according to PoC (6).

[1] although represented with respect to THB-splines for S and HB-splines for K

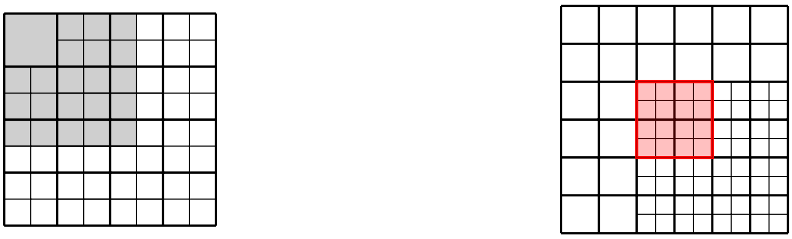

Figure 2: Left: Support of a cubic HB-spline (entire box) and of a THB-spline (grey) of level 0 on a hierarchy with two levels, $N = 1$. The associated coefficient functional is supported on the larger cell (that belongs to $\Omega^0 \setminus \Omega^1$) in the top left corner. Right: Hierarchy of 2 levels Ω^0 and Ω^1, subdomain $\Delta = \hat{\Omega}^0$ highlighted in red. We have $\hat{\Omega}^0 = \Omega^1$, hence $\hat{J}^0 = \emptyset$ and $\hat{J}^1 \not\subseteq J^1$.

3. Hierarchical splines on restricted domains

We consider the *restricted domain*, which is a subset $\Delta \subset \Omega^0$ formed by active cells, i.e., by cells of the hierarchical mesh. We also associate with Δ the nested *restricted subdomains* $\hat{\Omega}^\ell = \Delta \cap \Omega^\ell$, in particular $\hat{\Omega}^0 = \Delta$. Analogously to (2), the restricted spline spaces and subdomains define the *restricted hierarchical spline space* $\hat{H}$. Note that this space $\hat{H}$ is not simply the restriction of the original one to the restricted domain Δ, see Fig. 2 (right). Still we have the following result:

Lemma 1. *The restriction of the hierarchical spline space H to the restricted domain Δ is a subspace of the restricted hierarchical spline space $\hat{H}$, i.e., $H|_\Delta \subseteq \hat{H}$.*

Proof. Consider $s \in H$. By definition, it satisfies

$$s|_{\Omega^0 \setminus \Omega^{\ell+1}} \in V^\ell|_{\Omega^0 \setminus \Omega^{\ell+1}} \ \forall \ell = 0, \ldots, N,$$

see (2). Since

$$\hat{\Omega}^0 \setminus \hat{\Omega}^{\ell+1} = (\Delta \cap \Omega^0) \setminus (\Delta \cap \Omega^{\ell+1}) = \Delta \cap (\Omega^0 \setminus \Omega^{\ell+1}) \subseteq \Omega^0 \setminus \Omega^{\ell+1},$$

it also satisfies

$$s|_{\hat{\Omega}^0 \setminus \hat{\Omega}^{\ell+1}} \in V^\ell|_{\hat{\Omega}^0 \setminus \hat{\Omega}^{\ell+1}} \ \forall \ell = 0, \ldots, N,$$

and thus it is an element of the restricted hierarchical spline space $\hat{H}$. □

We denote with $\hat{\beta}_i^\ell$ and $\hat{\tau}_i^\ell$ the HB- and THB-splines defined by the restricted spline bases with respect to the restricted subdomains $\hat{\Omega}^\ell$. Their indices form the sets

$$\hat{J}^\ell = \{i \in \mathbb{Z}^d \ : \ \widehat{\mathrm{supp}}\, \beta_i^\ell \subseteq \hat{\Omega}^\ell \wedge \widehat{\mathrm{supp}}\, \beta_i^\ell \not\subseteq \hat{\Omega}^{\ell+1}\},$$

which are defined analogously to (1), but based on the hierarchy of nested *restricted* subdomains $\hat{\Omega}^\ell$ and using the supports (which are denoted by $\widehat{\mathrm{supp}}$) with respect to $\hat{\Omega}^0$.

It should be noted that the index sets J^ℓ and $\hat{J}^\ell$ are generally not nested, i.e., $J^\ell \not\supseteq \hat{J}^\ell$ in general. Thus, the restricted index sets are not simply the restrictions of their un-restricted counterparts, see Fig. 2 for a counterexample.

Clearly, the HB-splines are identical on Δ, i.e.

$$\hat{\beta}_i^\ell = \beta_i^\ell|_\Delta, \tag{11}$$

if $i \in J^\ell \cap \hat{J}^\ell$. This, however, is not valid for THB-splines. Nevertheless, the THB-splines satisfy

$$\hat{\tau}_i^\ell|_{\hat{\Omega}^\ell \setminus \hat{\Omega}^{\ell+1}} = \tau_i^\ell|_{\hat{\Omega}^\ell \setminus \hat{\Omega}^{\ell+1}} = \beta_i^\ell|_{\hat{\Omega}^\ell \setminus \hat{\Omega}^{\ell+1}}$$

We have the following simple result.

Lemma 2. *A selected B-spline β_i^ℓ, $i \in J^\ell$, is present in the restricted hierarchical basis, i.e., $i \in \hat{J}^\ell$, if its support is contained in the restricted domain, $\operatorname{supp} \beta_i^\ell \subseteq \Delta$. This also applies to the associated truncated hierarchical B-spline.*

In addition to the spline projectors introduced in (9), we will need the *restricted spline projectors* with respect to restricted domains Δ. The coefficient functionals will be denoted by $\hat{\mu}_i^\ell(f)$ and $\hat{\nu}_i^\ell(f)$ for HB- and THB-splines, respectively.

Given a THB-spline τ_i^ℓ, $i \in \hat{J}^\ell$, we identify a simple sufficient condition[2] on the restricted domain Δ, which guarantees that the coefficients of any hierarchical spline function $s \in H$ are identical.

Proposition 3. *Consider the coefficient of a level ℓ THB-spline τ_j^ℓ with index $j \in J^\ell$. The original and restricted THB-spline coefficients are identical,*

$$\nu_j^\ell(s) = \hat{\nu}_j^\ell(s|_\Delta) \quad \textit{for all} \quad s \in H,$$

if the restricted domain Δ contains the support of the associated HB-spline, i.e., if $\operatorname{supp} \beta_j^\ell \subseteq \Delta$.

Proof. According to Lemma 2, the index $j \in J^\ell$ also belongs to $\hat{J}^\ell$, since the support of β_j^ℓ is contained in the restricted domain Δ.

Any spline function $s \in H$ possesses local representations

$$s|_{\Omega^0 \setminus \Omega^{\ell+1}} = \sum_{i \in \mathbb{Z}^d} d_i^\ell \beta_i^\ell|_{\Omega^0 \setminus \Omega^{\ell+1}}, \quad \text{and} \quad s|_{\hat{\Omega}^0 \setminus \hat{\Omega}^{\ell+1}} = \sum_{i \in \mathbb{Z}^d} \hat{d}_i^\ell \hat{\beta}_i^\ell|_{\hat{\Omega}^0 \setminus \hat{\Omega}^{\ell+1}},$$

on the "rings" of the original and the restricted hierarchical spline space, respectively. The local linear independence of the B-splines, combined with (11), implies that the coefficients are identical, $d_i^\ell = \hat{d}_i^\ell$, if the associated basis functions contribute to this representation, i.e., if $\hat{\beta}_i^\ell|_{\hat{\Omega}^0 \setminus \hat{\Omega}^{\ell+1}} \neq 0$ (which implies $\beta_i^\ell|_{\Omega^0 \setminus \Omega^{\ell+1}} \neq 0$). In particular, this condition is satisfied for the considered index $j \in J^\ell$, hence

$$d_j^\ell = \hat{d}_j^\ell. \tag{12}$$

Indeed, the support of the associated basis function satisfies $\operatorname{supp} \beta_j^\ell \subseteq \Delta = \hat{\Omega}^0$ according to the assumption, and $\operatorname{supp} \hat{\beta}_j^\ell \not\subseteq \hat{\Omega}^{\ell+1}$ since $j \in \hat{J}^\ell$, thus $\hat{\beta}_j^\ell|_{\hat{\Omega}^0 \setminus \hat{\Omega}^{\ell+1}} \neq 0$. Finally we combine (12) with the PoC property, see (10), to conclude

$$\nu_j^\ell(s) = d_j^\ell = \hat{d}_j^\ell = \hat{\nu}_j^\ell(s|_\Delta). \qquad \square$$

We define the *extended support* of a HB-spline,

$$\operatorname{supp}^\star \beta_j^\ell = \operatorname{aabb}\Big(\bigcup \{\operatorname{supp} \beta_i^k \;:\; \operatorname{supp} \beta_i^k \supset \operatorname{supp} \beta_j^\ell,\ i \in J^k, \quad k = 1, \ldots, \ell - 1\}\Big),$$

where aabb denotes the *axis-aligned bounding box.* For future reference we state an assumption that we will need in Theorem 6 (see Section 4.3).

Assumption II (Strong mesh grading)**.** Only elements of two consecutive levels are present in the extended support of any selected basis function β_i^ℓ. Consequently, the active cells in the extended support of any selected basis function

- have level ℓ, or
- have either level ℓ or level $\ell - 1$, or
- have either level ℓ or level $\ell + 1$.

[2] While the detailed investigation of weaker conditions is beyond the scope of the present paper, we note that it suffices if Δ contains at least one active cell of level ℓ that belongs to $\operatorname{supp} \beta_j^\ell$, according to PoC.

Now we extend the sufficient condition[3] stated in the previous proposition to HB-splines.

Proposition 4. *Consider a level ℓ HB-spline β_j^ℓ with index $j \in J^\ell$. The original and restricted HB-spline coefficients are identical,*

$$\mu_j^\ell(s) = \hat{\mu}_j^\ell(s|_\Delta) \quad \textit{for all} \quad s \in H, \tag{13}$$

if the restricted domain Δ contains the extended support of the HB-spline, i.e., if $\operatorname{supp}^\star \beta_j^\ell \subseteq \Delta$.

Proof. According to Lemma 2, the index $j \in J^\ell$ also belongs to $\hat{J}^\ell$, since the support of β_j^ℓ, which is a subset of the extended support, is contained in the restricted domain Δ. Any given function $s \in H$ possesses two representations on Δ,

$$s|_\Delta = \sum_{k=0}^{N} \sum_{i \in J^k} \nu_i^k(s) \tau_i^k|_\Delta \quad \text{and} \quad s|_\Delta = \sum_{k=0}^{N} \sum_{i \in \hat{J}^k} \hat{\nu}_i^k(s|_\Delta) \hat{\tau}_i^k, \tag{14}$$

with respect to the restrictions of the globally defined THB-splines and with respect to the THB-splines defined by the restricted spline bases, respectively. The previous result guarantees the equality $\nu_i^k(s) = \hat{\nu}_i^k(s|_\Delta)$ of the coefficients whenever the associated B-splines satisfy $\operatorname{supp} \beta_i^k \subseteq \Delta$.

Both versions of the THB-splines admit representations in terms of HB-splines, from which they are constructed via truncation, see (3). These representations take the form

$$\tau_i^k|_\Delta = \beta_i^k|_\Delta - \sum_{k'=k+1}^{N} \sum_{i' \in J^{k'}} c_{i,i'}^{k,k'} \beta_{i'}^{k'}|_\Delta \quad \text{and} \quad \hat{\tau}_i^k = \hat{\beta}_i^k - \sum_{k'=k+1}^{N} \sum_{i' \in \hat{J}^{k'}} \hat{c}_{i,i'}^{k,k'} \hat{\beta}_{i'}^{k'} \tag{15}$$

with certain truncation coefficients $c_{i,i'}^{k,k'}$ and $\hat{c}_{i,i'}^{k,k'}$ The following observations are in order.

(i) The corresponding truncation coefficients are identical, i.e. $c_{i,i'}^{k,k'} = \hat{c}_{i,i'}^{k,k'}$, if the support of the associated HB-spline satisfies $\operatorname{supp} \beta_i^k \subseteq \Delta$.

(ii) The truncation coefficients $c_{i,i'}^{k,k'}$ and $\hat{c}_{i,i'}^{k,k'}$ are non-zero only if the supports satisfy $\operatorname{supp} \beta_{i'}^{k'} \subseteq \operatorname{supp} \beta_i^k$ and $\operatorname{supp} \hat{\beta}_{i'}^{k'} \subseteq \operatorname{supp} \hat{\beta}_i^k$, respectively,

Substituting (15) into (14) transforms the THB-spline representation into the HB-spline representation. After re-arranging the sums we arrive at

$$s|_\Delta = \sum_{k=0}^{N} \sum_{i \in J^k} \mu_i^k(s) \beta_i^k|_\Delta \quad \text{and} \quad s|_\Delta = \sum_{k=0}^{N} \sum_{i \in \hat{J}^k} \hat{\mu}_i^k(s|_\Delta) \hat{\beta}_i^k.$$

In particular, the two coefficients considered in (13) evaluate to

$$\mu_j^\ell(s) = \nu_j^\ell(s) - \underbrace{\sum_{k=0}^{\ell-1} \sum_{i \in J^\ell} c_{i,j}^{k,\ell} \nu_i^k(s)}_{(\star)} \quad \text{and} \quad \hat{\mu}_j^\ell(s|_\Delta) = \hat{\nu}_j^\ell(s|_\Delta) - \underbrace{\sum_{k=0}^{\ell-1} \sum_{i \in \hat{J}^\ell} \hat{c}_{i,j}^{k,\ell} \hat{\nu}_i^k(s|_\Delta)}_{(\star)}$$

In both sums, the two coefficients $c_{i,j}^{k,\ell}$ and $\hat{c}_{i,j}^{k,\ell}$ are nonzero only if

$$\operatorname{supp} \beta_i^k \supseteq \operatorname{supp} \beta_j^\ell \quad \text{and} \quad \operatorname{supp} \hat{\beta}_i^k \supseteq \operatorname{supp} \hat{\beta}_j^\ell,$$

according to (ii). Moreover, under these conditions, the coefficients even take the same values, $c_{i,j}^{k,\ell} = \hat{c}_{i,j}^{k,\ell}$, since $\operatorname{supp} \beta_i^k \subseteq \operatorname{supp}^\star \beta_j^\ell \subseteq \Delta$, as noted in ($i$). Finally we note that the latter condition also ensures that the associated THB-spline coefficients are identical, $\nu_i^k(s) = \hat{\nu}_i^k(s|_\Delta)$, according to Proposition 3. These three observations imply that the two double sums ($\star$) take the same value. We complete the proof by using Proposition 3 once more, which confirms $\nu_j^\ell(s) = \hat{\nu}_j^\ell(s|_\Delta)$. □

[3] Again we note that the investigation of necessary and sufficient conditions is beyond the scope of this paper.

4. Defining spline projectors by local fitting

We recall the local fitting procedure and use it for defining THB- and HB-spline projectors.

4.1. Local fitting

We consider a linearly independent system of functions $\Phi = \{\phi_k : k \in K\}$ defined on a given restricted domain Δ, where K is a finite index set. We assume that the domain is equipped with nodes (i.e., sampling points) $u_s \in \Delta$ with associated positive weights w_s, $s \in S$, where S is another finite index set.

Discrete least-squares fitting considers the over-constrained linear system $Ac = f$ with

$$A = (\sqrt{w_s}\phi_k(u_s))_{s\in S, k\in K}, \quad c = (c_k)_{k\in K}, \quad \text{and} \quad f = (\sqrt{w_s} f(u_s))_{s\in S},$$

which is solved in the least-squares sense, resulting in $c = (A^T A)^{-1} A^T f$. A unique least-squares solution exists for certain configurations of nodes that are determined by the system Φ. The resulting function

$$\sigma = \sum_{s\in S} c_s \phi_s$$

minimizes the Euclidean norm of the residual vector $\|(\sqrt{w_s}(f(u_s) - \sigma(u_s))_{s\in S}\|$ among all functions in span Φ.

Two specific cases will be considered in the next two sections. Clearly, the result depends on the choice of nodes and weights. If we consider spline functions ϕ_k, then the resulting function is the best approximation with respect to the L^2 norm if the quadrature rule defined by nodes and weights exactly integrates the products $\phi_k \phi_{k'}$ and $f\phi_k$, and it produces an approximation otherwise.

We focus on two possibilities to define the nodes and weights:

(a) We consider the Gauss points and weights on the restricted mesh. More precisely, we choose $(p+1)^d$ Gauss points per element, and we scale the Gauss weights by the volume of the element.

(b) We consider a tensor-product grid of interpolation points, which is unisolvent for the finest tensor product space that is active on Δ. All the weights are set to 1.

Appendix A describes the *smart sampling* method, which efficiently generates the points needed for the second choice (b). It will be shown that the total number of required function evaluations does not exceed $10^d|\mathcal{H}|$.

4.2. The THB-spline projector

We define the functionals ν_i^ℓ that generate the coefficients of the THB-spline projector. Given a fixed level ℓ_0 and an index $i_0 \in J^{\ell_0}$, we choose the restricted domain $\Delta = \operatorname{supp} \beta_{i_0}^{\ell_0}$ and we consider the THB-splines $\hat{\tau}_i^\ell$ defined on it. These functions form the system Φ. Now, given a function f, we perform local fitting as described in the previous section and assign the value of the coefficient associated with $\hat{\tau}_{i_0}^{\ell_0}$ to $\nu_{i_0}^{\ell_0}(f)$.

For the first choice (a) of nodes, the least-squares solution of the linear system is unique, since the matrix $A^T A$ is simply the mass matrix of the system Φ. The uniqueness is also given for the second choice (b), since one may consider the inner product defined by the tensor-product grid of interpolation points instead of the L^2 inner product. This procedure defines a spline projector (9) according to Proposition 3.

We analyze the computational costs per coefficient, focusing on the second choice (b) of nodes: The restricted domain Δ consists of at most $(p+1)^d$ cells of level ℓ_0. The active cells belong to the levels ℓ_0 and $\ell_0 + 1$ of the subdomain hierarchy, due to Assumption I about the mesh grading. The number of possible cases is thus bounded by $2^{(p+1)^d}$, since cells of level $\ell_0 + 1$ are created by refining blocks of cells of level ℓ_0.

Figure 3 lists the 44 possible configurations for $p = d = 2$. The upper bound is rather pessimistic.

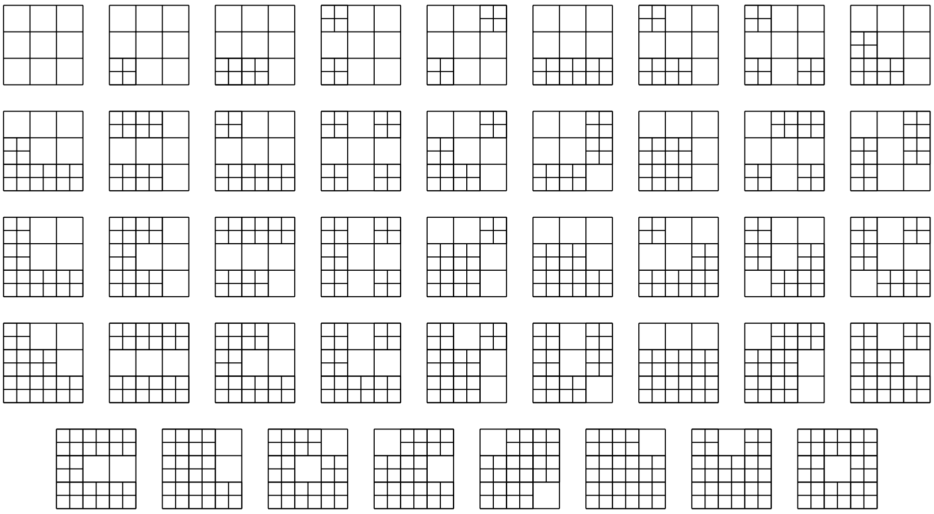

Figure 3: Possible configurations of the restricted domain Δ for $p = 2$ in the case of THB-splines (up to symmetries).

Assumption I also implies that the grid of nodes can be chosen such that it contains no more than $C_g\, p^d$ points, where the constant C_g does not depend on the polynomial degree p. See Appendix A for more details.

We may assume that all the matrices $(A^T A)^{-1} A^T$ that may occur have been pre-computed and stored in a look-up table. In fact, it suffices to store the row vectors that correspond to the coefficients $\nu_{i_0}^{\ell_0}(f)$. In each specific situation, finding the right entry in the table via binary search requires not more than $\log 2^{(p+1)^d} = \mathcal{O}(p^d)$ steps.

In addition, we need to evaluate f at a tensor-product grid of nodes containing $\mathcal{O}(p^d)$ nodes, and to compute the inner product with the pre-computed row vector. If smart sampling is employed, then the total number of function evaluations (which are assumed to take constant time) is $\mathcal{O}(|\mathcal{H}|)$.

We summarize the results:

Theorem 5. *Consider the functionals ν_i^ℓ defined by local fitting with respect to the restricted domain* $\operatorname{supp} \beta_i^\ell$*, using a unisolvent set of nodes.*

(i) The linear operator P_τ is a THB-spline projector.

(ii) When using the second choice of nodes, the evaluation of the spline coefficients requires $\mathcal{O}(p^d)$ flops and $\mathcal{O}(1)$ function evaluations per degree of freedom if Assumption I is satisfied.

In contrast to this, the first choice (a) leads to higher computational costs, since the values of f at $\mathcal{O}(p^{2d})$ nodes are used. Even though the first choice should give a better approximation of the (local) L^2 projection, our experiments did not lead us to observe significant differences between the computational results for the two different choices of the nodes.

In addition to this result, we also obtain a bound

$$|\nu_{i_0}^{\ell}(f)| \le C_\tau \|f\|_{\infty, \operatorname{supp} \beta_i^\ell} \tag{16}$$

with the constant

$$C_\tau = C_g\, p^d \max_{\text{all cases}} \|(A^T A)^{-1} A^T\|_\infty$$

where $\|.\|_\infty$ is the element-wise maximum norm of a matrix. Recall that $C_g\, p^d$ is an upper bound on the number of evaluation points.

4.3. The HB-spline projector

Now we define the functionals μ_i^ℓ that generate the coefficients of the HB-spline projector.

Given a fixed level ℓ_0 and an index $i_0 \in J^\ell$, we choose the restricted domain Δ as the extended support $\operatorname{supp}^\star \beta_{i_0}^{\ell_0}$ and we consider the HB-splines $\hat{\beta}_i^\ell$ defined on it. These functions form the system Φ. Similar to the previous case, we have the two main possibilities (a) and (b) for defining interpolation nodes and weights.

Now, given a function f, we perform local fitting as described in the previous section and assign the value of the coefficient associated with $\hat{\beta}_{i_0}^{\ell_0}$ to $\mu_{i_0}^{\ell_0}(f)$. The same arguments as for THB-splines prove the uniqueness of the solution. This procedure defines a spline projector (9) according to Proposition 4.

We analyze the computational costs per coefficient, focusing on the second choice of nodes. Any restricted domain Δ contains at most two levels, due to Assumption II. We distinguish between two cases: First, if $\Delta = \operatorname{supp} \beta_{i_0}^{\ell_0}$, only cells of levels ℓ_0 and $\ell_0 + 1$ are present. The number of possible cases does not exceed $2^{(p+1)^d}$, since there are $(p+1)^d$ cells of level ℓ_0. Second, if $\Delta \neq \operatorname{supp} \beta_{i_0}^{\ell_0}$, only cells of levels ℓ_0 and $\ell_0 - 1$ are present. In this situation, the number of cells of level $\ell_0 - 1$ does not exceed $\lceil \frac{3p+3}{2} \rceil^d$. Consequently, the number of possible cases does not exceed $2^{\lceil \frac{3p+3}{2} \rceil^d}$.

Again, Assumption II also implies that the grid of nodes can be chosen such that it contains no more than $C_g\, p^d$ points, where the constant C_g does not depend on the polynomial degree p. See Appendix A for more details.

Once more, we may assume that all possible matrices $(A^T A)^{-1} A^T$ (more precisely, the row vectors that corresponds to the coefficient $\mu_{i_0}^{\ell_0}(f)$) have been pre-computed and stored in a look-up table. In each specific situation, finding the right entry in the table via binary search requires $\mathcal{O}(p^d)$ steps. In addition, we need to evaluate f at a tensor-product grid containing $\mathcal{O}(p^d)$ nodes, and to compute the inner product with the pre-computed row vector.

We summarize the results:

Theorem 6. *Consider the functionals μ_i^ℓ defined by local fitting with respect to the restricted domain $\operatorname{supp}^\star \beta_i^\ell$, using a unisolvent set of nodes.*

(i) The linear operator P_β is a HB-spline projector.

(ii) When using the second choice of nodes, the evaluation of the spline coefficients requires $\mathcal{O}(p^d)$ flops and $\mathcal{O}(1)$ function evaluations per degree of freedom if Assumption II is satisfied.

Similar to the case of THB-splines, we additionally obtain a bound

$$|\mu_i^\ell(f)| \leq C_\beta \|f\|_{\infty, \operatorname{supp}^* \beta_i^\ell}. \tag{17}$$

with the constant

$$C_\beta = C_g p^d \max_{\text{all cases}} \|(A^T A)^{-1} A^T\|_\infty.$$

5. Approximation using adaptive refinement

Given a function $f : \Omega^0 \to \mathbb{R}$, we compute a hierarchical spline approximation as follows:

1. The number of levels is initialized by $N = 0$.
2. Compute the spline approximation $P_\gamma f$ with respect to one of two bases, i.e., for $\gamma = \beta$ and $\gamma = \tau$, see (9). The coefficient functions are defined by local fitting, see Section 4.
3. Mark the elements $e \in E$ of the hierarchical mesh associated with P_γ that are identified by the refinement indicator and collect the marked elements, and the elements in the neighborhood whose size is specified by the extension parameter, in the set $E^\star$.
4. Terminate the iteration if $E^\star = \emptyset$. Otherwise create the refined subdomain hierarchy, as follows:
   - Increase N by 1 and initialize Ω^N by the empty set.
   - Increase the level of each marked element $e \in E^\star$ by one. More precisely, add e to $\Omega^{\text{level}(e)+1}$ where $\text{level}(e) = \max\{\ell \,:\, e \subseteq \Omega^\ell\}$.

- Restore the mesh grading by applying a suitable closure operation to the subdomain hierarchy. See Buffa et al. (2016) for a detailed discussion of mesh refinement algorithms.

5. Continue with Step 2.

The refinement is driven by the maximum error. Given a tolerance ε, we mark an element e if

$$\|f - P_\gamma f\|_{\infty,e} \geq \varepsilon, \tag{18}$$

where either $\gamma = \beta$ or $\gamma = \tau$. In addition, the refinement depends on the extension parameter, which controls the locality of the refinement process. While the refinement region needs to be large enough to create sufficiently many new degrees of freedom, it should also remain as local as possible in order to keep the data volume small. Moreover, it is desirable to avoid the frequent re-computation of individual coefficients. A more detailed analysis of the choice of the refinement strategies is beyond the scope of the present paper, see also Kiss et al. (2014).

Given a hierarchical mesh, which is defined by a subdomain hierarchy, we define for each element e the index sets that contains indices of active basis functions,

$$J_\gamma^{\ell,e} = \{i \in J^\ell \ : \ \gamma_i^\ell|_e \neq 0\}, \quad \ell = 0, \ldots, N$$

for the cases of HB-splines ($\gamma = \beta$) and THB-splines ($\gamma = \tau$). Clearly we have $J_\tau^{\ell,e} \subseteq J_\beta^{\ell,e}$. We use these sets to define the *support extension*

$$\operatorname{ext}(e) = \bigcup_{\substack{\ell=0,\ldots,N \\ i \in J_\tau^{\ell,e}}} \operatorname{supp} \beta_i^\ell,$$

and the *extended support extension*

$$\operatorname{ext}^\star(e) = \bigcup_{\substack{\ell=0,\ldots,N \\ i \in J_\beta^{\ell,e}}} \operatorname{supp}^\star \beta_i^\ell.$$

Again we get $e \subseteq \operatorname{ext}(e) \subseteq \operatorname{ext}^\star(e)$. After these preparations we can now state our result concerning the approximation order, which is based on the fact that spline projectors reproduce polynomials:

Theorem 7. *Given a function $f \in \mathcal{C}^{p+1}(\Omega)$, the element-wise maximum error can be estimated by*

$$\|f - P_\gamma f\|_{\infty,e} \leq C \operatorname{diam}(e)^{p+1} \max_{|\alpha|=p+1} \|D^\alpha f\|_{\infty,\operatorname{ext}^{[\star]}(e)} \tag{19}$$

where either $\gamma = \beta$ for HB-splines or $\gamma = \tau$ for THB-splines, where the maximum norm on the right-hand side considers the restriction to the extended support extension $\operatorname{ext}^\star(e)$ or to the (standard) support extension $\operatorname{ext}(e)$, respectively.

Proof. Denoting by $T_e f$ the Taylor polynomial of degree p of f at the center of e, the triangle inequality gives

$$\|f - P_\gamma f\|_{\infty,e} \leq \|f - T_e f\|_{\infty,e} + \|P_\gamma(f - T_e f)\|_{\infty,e}$$

since P_γ reproduces polynomials of degree p. A bound for the first term is obtained by the standard error estimate for Taylor expansions,

$$\|f - T_e f\|_{\infty,e} \leq C_p \operatorname{diam}(e)^{p+1} \max_{|\alpha|=p+1} \|D^\alpha f\|_{\infty,e}$$

where the constant is independent of f.

We now analyze the second term in the case of THB-splines, obtaining

$$\|P_\tau(f - T_e f)\|_{\infty,e} = \|\sum_{\ell=0}^{N} \sum_{i \in J^\ell} \nu_i^\ell(f - T_e f)\tau_i^\ell\|_{\infty,e} \leq \max_{\ell=0,\ldots,n} \max_{i \in J_\tau^{\ell,e}} |\nu_i^\ell(f - T_e f)|$$

since the THB-splines form a convex partition of unity. Using (16) and again the standard estimate for Taylor series gives

$$|\nu_i^\ell(f-T_e f)| \leq C_\tau \|f-T_e f\|_{\infty,\mathrm{ext}(e)} \leq C_\tau C_p \mathrm{diam}(\mathrm{ext}(e))^{p+1} \max_{|\alpha|=p+1} \|D^\alpha f\|_{\infty,\mathrm{ext}(e)}$$

According to Assumption I (mesh grading), the support extension satisfies

$$\mathrm{diam}(\mathrm{ext}(e)) \leq (4p+2)\mathrm{diam}(e).$$

Combining this observation with the previous four inequalities confirms (19) for THB-splines with

$$C = C_p(1+(4p+2)^{p+1}C_\tau).$$

The case of HB-splines can be dealt with similarly: Analyzing the second term gives

$$\|P_\beta(f-T_e f)\|_{\infty,e} = \|\sum_{\ell=0}^{N}\sum_{i\in J^\ell} \mu_i^\ell(f-T_e f)\beta_i^\ell\|_{\infty,e} \leq 2 \max_{\ell=0,\dots,n} \max_{i\in J_\tau^{\ell,e}} |\mu_i^\ell(f-T_e f)|$$

since B-splines of no more than two levels are present in each cell (according to the mesh grading assumptions) and the B-splines within each level form a convex partition of unity. Using (17) and the standard result for Taylor series gives

$$|\mu_i^\ell(f-T_e f)| \leq C_\beta \|f-T_e f\|_{\infty,\mathrm{ext}^\star(e)} \leq C_\beta C_p \mathrm{diam}(\mathrm{ext}^\star(e))^{p+1} \max_{|\alpha|=p+1} \|D^\alpha f\|_{\infty,\mathrm{ext}^\star(e)}$$

According to the Assumption II (strong mesh grading), the extended support extension satisfies

$$\mathrm{diam}(\mathrm{ext}^\star(e)) \leq (4p+2)\mathrm{diam}(e).$$

Combining this observation with the previous four inequalities confirms (19) with

$$C = C_p(1+2(4p+2)^{p+1}C_\beta). \qquad \square$$

6. Experimental results

We perform several numerical experiments to validate and underline some relevant features of our approach. The entire implementation has been carried out in C++ with the use of the **G+Smo** library (Mantzaflaris et al., 2019).

Example 1. We perform adaptive approximation of the function

$$f(x,y) = 1 - \tanh\left(\frac{\sqrt{x^2+y^2}-0.3}{0.05\sqrt{2}}\right), \quad (x,y)\in[-1,1]^2,$$

which was taken from the work of Hennig et al. (2018). The approximation obtained with our scheme is compared with the global fitting strategy described by Kiss et al. (2014), and with uniform refinement. The value of the extension parameter (see Step 3) is set to p. The algorithms terminate if the maximum error is below a certain threshold (in this example ε = 1e-4).

Table 1 summarizes the obtained results. The results obtained by global fitting and by local fitting-based quasi-interpolation with HB and THB-splines are quite similar (We even obtain the same meshes and virtually the same maximum error, except for depth 5). In fact, a slightly smaller number of degrees of freedom suffices to reach the desired accuracy via quasi-interpolation. As to be expected, many more degrees of freedom are needed to achieve the same accuracy when using uniformly refined tensor-product (TP) splines. The convergence rates are depicted in Fig. 4.

		Quasi interpolation			Global fitting			
p	depth	DoF	error (HB)	error (THB)	DoF (THB)	error (HB & THB)	DoF (TP)	error (TP)
2	1	324	1.684e-1	1.684e-1	324	1.243e-1	324	1.243e-1
	2	1012	2.213e-2	2.213e-2	1128	1.811e-2	1156	1.811e-2
	3	2076	2.016e-3	2.016e-3	2512	1.545e-3	4356	1.545e-3
	4	4648	1.688e-4	1.688e-4	4948	1.485e-4	16900	1.485e-4
	5	7248	9.103e-5	9.103e-5	7420	7.227e-5	66564	1.903e-5
3	1	361	1.343e-1	1.343e-1	361	9.676e-2	361	9.676e-2
	2	1225	1.491e-2	1.491e-2	1225	1.142e-3	1225	1.142e-2
	3	2593	5.780e-4	5.780e-4	2977	3.526e-4	4489	3.527e-4
	4	4753	5.019e-5	2.845e-5	4921	3.913e-5	17161	1.972e-5

Table 1: Example 1 – Maximum errors and degrees of freedom for approximation via quasi interpolation or global fitting with HB-splines, THB-splines and tensor-product (TP) splines.

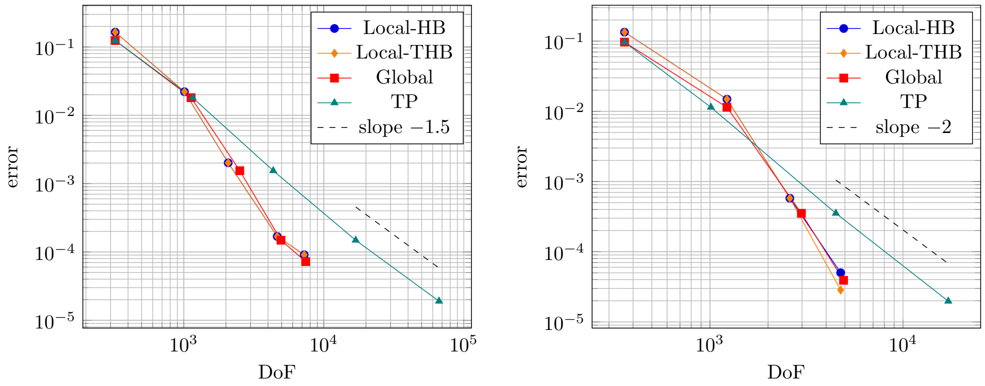

Figure 4: Example 1 – Convergence rates for $p = 2$ (left) and $p = 3$ (right)

		HB		THB	
p	depth	DoF	error	DoF	error
2	1	324	1.684e-1	324	1.684e-1
	2	760	2.213e-2	760	2.213e-2
	3	1580	2.016e-3	1580	2.016e-3
	4	3672	1.688e-4	3696	1.688e-4
	5	5360	8.933e-5	5384	8.933e-5

Table 2: Example 1 – Results for biquadratic HB- and THB-splines with extension parameter 1

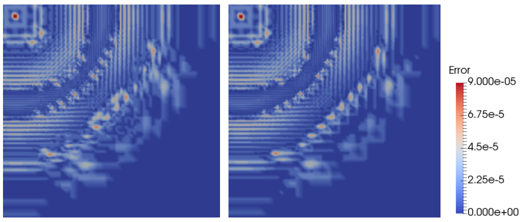

Figure 5: Error distribution for the HB- (left) and THB-spline (right) approximations for depth 5, cf. Table 2

While we got identical meshes for HB- and THB-splines in this experiment, this is not always the case. Table 2 reports the result for biquadratic splines using the extension parameter 1, where the generated hierarchical meshes and the associated numbers of degrees of freedom for HB- and THB-splines are slightly different. Even though the maximum errors are again identical, the distribution of the error (shown in Fig. 5) is not.

Example 2. We approximate the function

$$f(x,y) = \frac{1}{9}(\tanh(9y-9x)+1) + \frac{1}{1.5\exp((10x-6)^2+(10y+7)^2)}, \quad (x,y) \in [-1,1]^2$$

that has been considered by Bracco et al. (2018), using adaptive spline refinement. We set again $\varepsilon = 1\text{e-}4$. Here we exploit the observation that many of the coefficients have been computed during previous steps of the iterative refinement process. This is in contrast with global approximation strategies that require the solution of one large linear system per iteration. Table 3 shows the error and the number of computed coefficients per iteration. It is worth noting than even though the use of HB-splines leads to a larger restricted domain Δ associated with each basis function, this does not have much impact on the locality of the approach. (Again, the refinement processes creates the same meshes for HB- and THB-splines, and also the values of the error are virtually identical in almost all cases.)

		Quasi interpolation			Computed coeffs.	
p	depth	DoF	error (HB)	error (THB)	HB	THB
2	1	324	1.185e-1	1.185e-1	324	324
	2	1044	2.088e-2	2.088e-2	1016	1016
	3	2678	1.421e-3	1.421e-3	2408	2384
	4	3376	1.245e-4	1.245e-4	1064	1030
	5	3785	9.688e-5	9.688e-5	616	616
3	1	361	1.660e-1	1.660e-1	361	361
	2	1150	1.081e-2	1.081e-2	1135	1135
	3	3040	2.220e-4	2.220e-4	2826	2793
	4	3421	4.408e-5	2.781e-5	617	617

Table 3: Example 2 – Maximum errors and computed coefficients

Example 3. We explore the impact of Assumptions I and II, and of the extension parameter. The latter parameter controls the size of the neighborhood of a marked element that is subject to refinement, see Step 3 of the algorithm in Section 5. We perform adaptive approximation of the function

$$f(x,y) = \frac{1}{5\exp((10x)^4 + (10y)^4)}$$

with $\Omega = [-1,1]^2$ and ε = 1e-4. We use either HB- or THB-splines and vary the degree and the extension parameter. We compare the average size of the linear systems.

Three observations are in order. First, the use of mesh grading leads to a smaller average size of the linear systems. According to the theory, Assumptions I and II ensure that the size varies between $(p+2)^d$ and $(2p+3)^d$ for THB-splines, and between $(p+2)^d$ and $(p+2\lfloor\frac{3p+3}{2}\rfloor)^d$ for HB-splines. In practice, the average size is far less than the upper bound. Second, increasing the extension parameter has a similar effect as mesh grading. Third, the linear systems for HB-splines have a larger size.

		THB-splines				HB-splines			
		without grading		with grading		without grading		with grading	
p	ext.	#lin. syst.	avg. size	#lin. syst.	avg. size	#lin. syst.	avg. size	#lin. syst.	avg. size
2	1	5244	27.27^2	5520	26.96^2	5244	45.68^2	5520	39.34^2
	2	6888	26.80^2	6960	26.76^2	6888	38.16^2	6960	37.72^2
3	1	3160	52.73^2	3412	51.88^2	3160	103.51^2	3412	82.64^2
	2	4024	52.67^2	4228	52.35^2	4024	89.39^2	4228	82.44^2
	3	5320	50.95^2	5344	50.94^2	5320	79.38^2	5344	78.70^2
4	1	3976	94.33^2	5412	82.64^2	4672	347.01^2	5864	125.00^2
	2	4608	89.35^2	5740	82.92^2	4608	192.62^2	5740	128.97^2
	3	6264	83.48^2	6620	80.93^2	6264	144.80^2	6620	116.22^2
	4	7128	82.69^2	7324	81.25^2	7128	130.39^2	7324	115.27^2

Table 4: Example 3 – Number and average size of the linear systems during the refinement process

We compare our QI scheme with SMS (Speleers and Manni, 2016), which was constructed in terms of THB-splines. In order to do so, we apply our scheme to two given functions, both defined in $[-1,1]^2$, that were considered in that paper, adopting the same setting. More precisely,

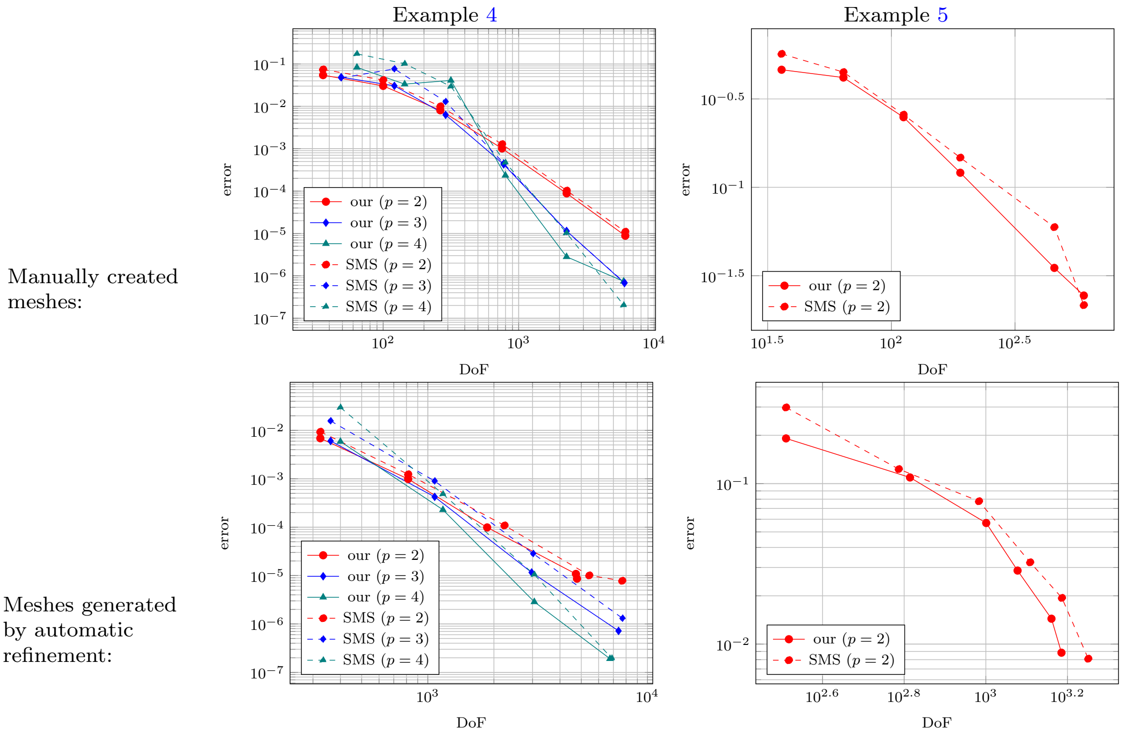

Figure 6: Comparison of the maximum error for the Examples 4 (left) and 5 (right) with manually created meshes and errors with respect to a fixed grid (top row) and for meshes created by automatic refinement (bottom row).

- we adopt *manually created sequences* of refined meshes (see Fig. 3 of Speleers and Manni, 2016), and
- we evaluate the maximum error on a *fixed regular grid* of 150 × 150 points.

In addition, we also compare the results created by the automatic mesh refinement algorithm.

Example 4. The first function is

$$f(x, y) = \frac{\tanh(9y - 9x) + 1}{9}$$

Figure 6 (left column) compares the approximations obtained by the two schemes. Our method results in a slightly better or comparable accuracy in almost all the cases[4]. Note that the finest grid (level 6), when considered on the entire domain, consists of 128 × 128 points. Using only 150 × 150 points for error estimation may not suffice. The higher accuracy of our method is also present for meshes created by automatic refinement. We used $\varepsilon = 1e - 5$ for $p = 2$ and $\varepsilon = 1e - 6$ for $p \geq 3$.

Example 5. The same setting as in the previous example is applied to the function

$$f(x, y) = \frac{2}{3\exp\left(\sqrt{(10x-3)^2+(10y-3)^2}\right)} + \frac{2}{3\exp\left(\sqrt{(10x+3)^2+(10y+3)^2}\right)} + \frac{2}{3\exp\left(\sqrt{(10x)^2+(10y)^2}\right)}$$

The obtained results are shown in the right column of Figure 6. Also in this case, our method reveals a slightly better accuracy in almost all cases. For the manually created meshes (top row),

[4] Speleers (2017) applies his QI schemes (which reproduce polynomials but not splines) to this function also. While the (manually created) sequence of refined meshes is slightly different in that paper, the results still indicate that sacrificing the spline projector property leads to a slight increase of the error, while the rate of convergence remains the same.

the grid of points used to estimate the error might lead to inaccurate results. We used $\varepsilon = 1e - 2$ for the automatic mesh refinement algorithm (bottom row).

Example 6. Finally, we present an example that reveals another significant advantage of out method. We fit the function from Example 3

$$f(x, y) = \frac{1}{5 \exp((10x)^4 + (10y)^4)}$$

on $\Omega = [-1, 1]^2$ using automatic refinement with threshold ε = 1e-4. Figure 7 shows the errors and the DoFs obtained with the two schemes (our scheme and SMS) using THB-splines of degree 3. SMS leads to non–symmetric meshes, even though the function is symmetric, see Fig. 8. This is due to the choice of performing quasi-interpolation via interpolation in only one of cells that form the support of the selected basis function, in agreement with (8).

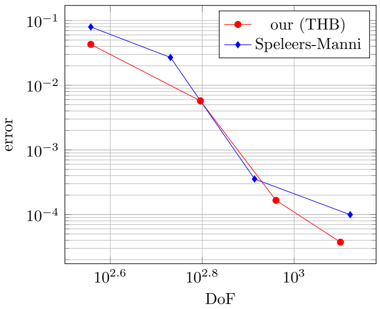

Figure 7: Maximum error obtained by adaptive refinement using our QI scheme and SMS for $p = 3$.

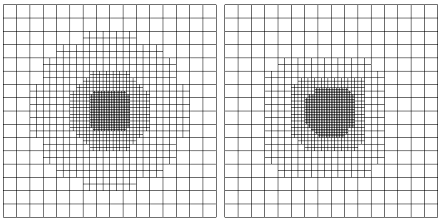

Figure 8: Meshes obtained at the finest level, see Figure 7, using our QI scheme (left) and SMS (right) for $p = 3$.

Example 7. Finally we present a simple one-dimensional example that illustrates the differences between our scheme for THB-splines and SMS. We use both schemes to approximate $y = |x|$ by quadratic hierarchical splines on [-1,1] defined on two levels with knot spans of length 1/8 and 1/16, where refinement took place on the interval $[-1/8, 1/8]$, see Fig. 9. SMS (blue) preserves the original approximation outside of the refined domain, whereas our scheme recomputes the coefficients of all functions in the support extension of the refined domain. In the first case, the error is more concentrated and larger, while it is more spread out and thus lower for the second method.

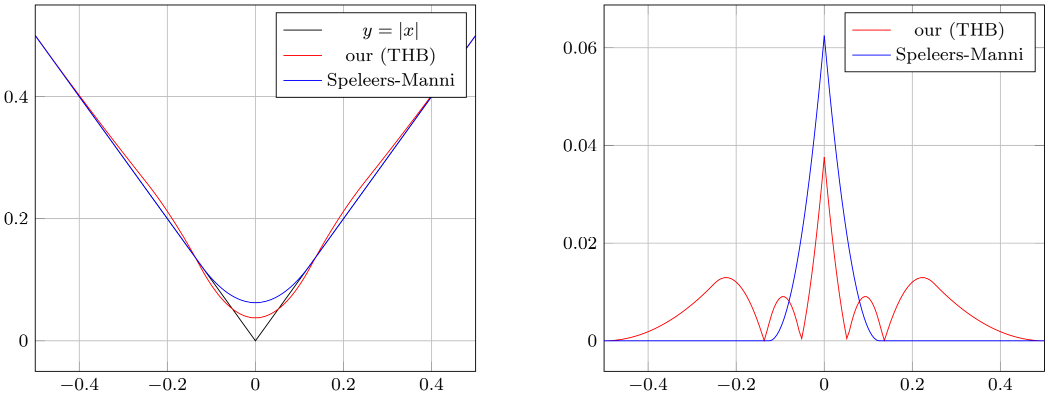

Figure 9: Approximation (left) and comparison of the error (right)

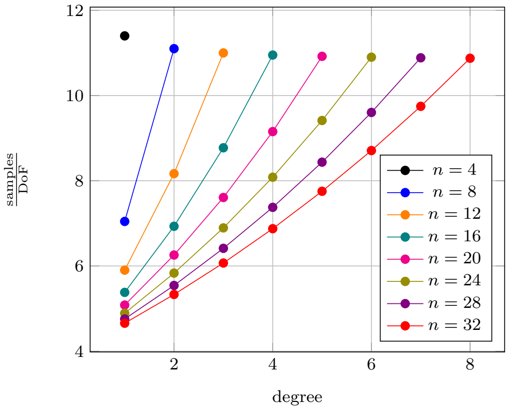

Figure 10: A domain $\Omega^0 \subset \mathbb{R}^2$ consisting of $2n \times 2n$ cells of level 0 is repeatedly refined by selecting the central $n \times n$ cells of each level ℓ to form the next level subdomain. Assumption I is satisfied if $n \geq 4p$. The plot reports the limit of the number of sample points per degree of freedom as $N \to \infty$ for various instances of n. For any fixed degree p, that value tends to 4 as $n \to \infty$.

Remark. Finally we compare the computational costs of our spline projector for THB-splines with SMS:

- First we analyze the number of required sample points. We give a rather conservative (but degree-independent) upper bound (10^d) for the number of such points per degree of freedom in Proposition 9. In practice, the ratio is closer to the lower bound of 2^d, which is realized when considering uniform meshes ($N = 0$), see Fig. 10. As noted by Speleers (2017) for $d = 2$, SMS requires "on average p^d" (and not more than $(p+1)^d$) samples per degree of freedom.

- Both schemes generate the THB-spline coefficients by evaluating linear combinations of samples values with predefined weights, and the number of floating point operations (flops) amounts to twice the number of sample points. Our method requires $2(4(p+1)+1)^d$ flops per coefficient, thus approximately 4^d times as many as SMS, which needs $2(p+1)^d$ flops per coefficient. Clearly, the evaluation of the THB-spline coefficients is ideally suited for parallel computing. Without using pre-computed inverse matrices, the computational costs would amount to $\mathcal{O}(p^{3d})$ flops per degree of freedom for both methods, again with a smaller constant for SMS and with the option of parallelizing the computations.

7. Conclusion

We proposed a novel QI scheme based on a local least-squares fitting. This defines two new spline projectors, one each for HB- and for THB-splines, which are formulated within a unifying framework. Moreover, we provided a detailed analysis of the computational complexity, which showed that the costs are comparable to Kraft (1998) for HB-splines and Speleers and Manni (2016) for THB-splines. We also analyzed the required number of sampled function values, showing that $\mathcal{O}(1)$ evaluations are sufficient to obtain spline projectors. Naturally, the computational costs grow with the degree p and with the dimension d, thereby imposing limitations on the applicability of the method, which can, however, be addressed to some extent via parallel computing. The theoretical results ensure the efficiency of the proposed scheme and provides meaningful indications on how to apply it effectively. The approximation properties of the projector were analyzed, and an adaptive refinement strategy was presented. The interdependence between the theoretical results achieved in the paper is summarized in Fig. 11. Numerical experiments were performed, showing that one obtains an accuracy comparable to the one of global approximation techniques.

Furthermore, a comparison with the QI scheme proposed by Speleers and Manni (2016) was carried out. Our construction

- is based on a unifying framework for HB- and THB-splines, while SMS is limited to THB-splines,
- guarantees the spline projector property with $\mathcal{O}(1)$ evaluations per degree of freedom,
- was shown experimentally to preserve certain symmetries of the given function better than SMS, and
- seems to give slightly lower errors in almost all computational experiments that we performed.

While it was possible to maintain overall the computational complexity, these advantages come at the price of higher computational costs, i.e., of a larger constant in the complexity estimate. Additionally, more computer memory is needed to store the look-up table that is used to generate the spline coefficients.

Future work will be devoted to the extension of the proposed approximation scheme to scattered data sets, both concerning the theory and the algorithmic aspects. In fact, local approximation strategies facilitate the use of parallel computing approaches. This might be further addressed to increase the efficiency of the current implementation. Future work may also include the extension to non-uniform meshes, where the use of pre-computed look-up tables is not possible. Finally, the application of the proposed scheme to IgA related problems, such as matrix assembly for hierarchical splines will be addressed, extending the results of Pan et al. (2019).

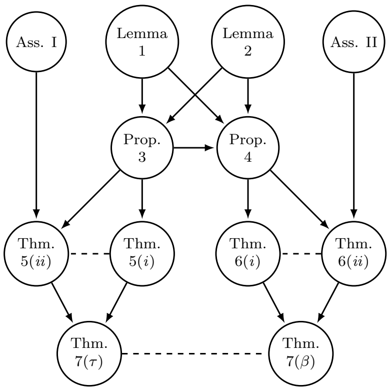

Figure 11: Dependencies between the assumptions and results in this paper

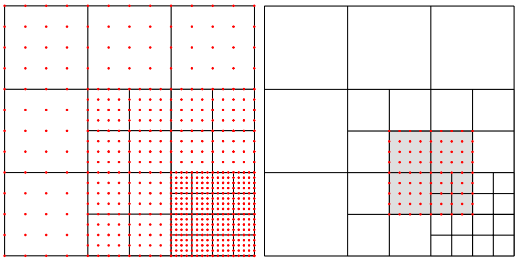

Figure 12: Left: Full set of candidate sample points. Right: grid of sample points on a restricted domain Δ (grey) for $p = 1$.

Acknowledgements

The research was funded by the Austrian Science Fund (FWF): W1214-N15, project DK3. The support is gratefully acknowledged.

Appendix A. Smart sampling

Any restricted domain Δ contains active cells of at most two levels, and it can be covered by (active and in-active) cells of the finer one among these two levels. We construct the grid on Δ by splitting each of these covering cells into 2^d cells via dyadic subdivision and placing the grid points at the cells' vertices. Consequently, all the grid points are vertices of the mesh that is created by subdividing each active cell into 4^d cells via 4-adic subdivision. These vertices form the set of candidate sample points, see Fig. 12.

Proposition 8. *For the restricted domains Δ considered in Section 4.2 and 4.3, the grid contains $\mathcal{O}(p^d)$ nodes and is unisolvent for the restricted hierarchical spline space $\hat{H}$.*

Proof. Any restricted domain Δ consist of $\mathcal{O}(p^d)$ cells of the finer level and we sample at most 3^d nodes per cell. The unisolvency is guaranteed since these grids are unisolvent for the full space of tensor-product splines defined on the finer grid, and the restricted hierarchical spline space is a subspace thereof. The unisolvency for tensor-product splines follows from the Schoenberg-Whitney conditions and the fact that the Δ consists of at least $(p+1)^d$ cells of the finer level. □

Proposition 9. *The total number of grid points (and hence the number of required function evaluations) does not exceed* $10^d|\mathcal{H}|$.

Proof. First we note that the total number of grid points is bounded by $5^d n_{\text{ac}}$, where n_{ac} denotes the number of active cells. Second, we consider all pairs of (T)HB-splines and active cells, where the cells are contained in the support of the (T)HB-spline. The number of such pairs is bounded from below by $(p+1)^d n_{\text{ac}}$, since the (T)HB-splines span the full space of tensor-product polynomials on each active cell. It is also bounded from above by $2^d(p+1)^d|\mathcal{H}|$, according to Assumption I. Consequently, $n_{\text{ac}} \leq 2^d|\mathcal{H}|$ and the result follows. □

Finally we note that this counting – which is based on set of candidate sample points – leads to an over-estimation of the required number of evaluations. Indeed, it suffices to perform 4-adic subdivision only for active cells that are "close" to the cells of next finer level. More precisely, in order to generate the sample points for THB-splines,

- the active cells within the offset region[5] of the subdomain $\Omega^{\ell+1}$ formed by p layers of cells of level ℓ need to be subdivided into 4^d cells via 4-adic subdivision,
- it suffices to subdivide all other cells into 2^d cells only, via binary subdivision.

The vertices of the refined mesh form the set of sample points. A similar observation applies to HB-splines.

References

de Boor, C., Fix, M., 1973. Spline approximation by quasi-interpolants. Journal of Approximation Theory 8, 19 – 45.

Bracco, C., Giannelli, C., Großmann, D., Sestini, A., 2018. Adaptive fitting with THB-splines: Error analysis and industrial applications. Computer Aided Geometric Design 62, 239–252.

Bracco, C., Giannelli, C., Sestini, A., 2017. Adaptive scattered data fitting by extension of local approximations to hierarchical splines. Computer Aided Geometric Design 52-53, 90 – 105. Geometric Modeling and Processing 2017.

Buffa, A., Giannelli, C., 2017. Adaptive isogeometric methods with hierarchical splines: Optimality and convergence rates. Mathematical Models and Methods in Applied Sciences 27, 2803–2807.

Buffa, A., Giannelli, C., Morgenstern, P., Peterseim, D., 2016. Complexity of hierarchical refinement for a class of admissible mesh configurations. Computer Aided Geometric Design 47, 83–92.

Calabró, F., Falini, A., Sampoli, M.L., Sestini, A., 2018. Efficient quadrature rules based on spline quasi-interpolation for application to IGA-BEMs. Journal of Computational and Applied Mathematics 338, 153 – 167.

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

Dokken, T., Lyche, T., Pettersen, K.F., 2013. Polynomial splines over locally refined box-partitions. Computer Aided Geometric Design 30, 331 – 356.

Farin, G., 2002. Curves and surfaces for CAGD: a practical guide. Morgan Kaufmann.

Forsey, D., Bartels, R., 1988. Hierarchical B-spline refinement. Comput. Graph. 22, 205–212.

Forsey, D., Bartels, R., 1995. Surface fitting with hierarchical splines. ACM Transactions on Graphics 14, 134–161.

Giannelli, C., Jüttler, B., Kleiss, S.K., Mantzaflaris, A., Simeon, B., Špeh, J., 2016. 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.

Giannelli, C., Jüttler, B., Speleers, H., 2012. THB-splines: the truncated basis for hierarchical splines. Computer Aided Geometric Design 29, 485–498.

Giannelli, C., Jüttler, B., Speleers, H., 2014. Strongly stable bases for adaptively refined multilevel spline spaces. Advances in Computational Mathematics 40, 459–490.

[5] This region is swept out the by the level ℓ HB-splines with a support intersecting the subdomain $\Omega^{\ell+1}$.

Greiner, G., Hormann, K., 1997. 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, Vanderbilt University Press, Nashville, TN. pp. 163–172.

Hennig, P., Ambati, M., De Lorenzis, L., Kästner, M., 2018. Projection and transfer operators in adaptive isogeometric analysis with hierarchical B-splines. Computer Methods in Applied Mechanics and Engineering 334, 311–336.

Hughes, T., Cottrell, J., Bazilevs, Y., 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering 194, 4135 – 4195.

Kiss, G., Giannelli, C., Zore, U., Jüttler, B., Großmann, D., Barner, J., 2014. Adaptive CAD model (re-)construction with THB-splines. Graphical Models 76, 278–288.

Kraft, R., 1998. Adaptive und linear unabhängige Multilevel B-Splines und ihre Anwendungen. Ph.D. thesis. Universität Stuttgart.

Lee, B.G., Lee, J.J., Kwon, K.R., 2005. Quasi-interpolants based multilevel B-spline surface reconstruction from scattered data, in: Gervasi, O., et al. (Eds.), Computational Science and Its Applications – ICCSA 2005, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 1209–1218.

Li, X., Zheng, J., Sederberg, T.W., Hughes, T.J., Scott, M.A., 2012. On linear independence of T-spline blending functions. Computer Aided Geometric Design 29, 63 – 76.

Lyche, T., Schumaker, L., 1975. Local spline approximation. Journal of Approximation Theory 15, 294 – 325.

Mantzaflaris, A., Giust, A., others (see website), 2019. G+Smo (Geometry plus Simulation modules) v0.8.1. github.com/gismo.

Mokriš, D., Jüttler, B., Giannelli, C., 2014. On the completeness of hierarchical tensor-product B-splines. Journal of Computational and Applied Mathematics 271, 53 – 70.

Pan, M., Jüttler, B., Giust, A., 2019. Fast Formation of Isogeometric Galerkin Matrices via Integration by Interpolation and Look-up. Technical Report 88. NFN Geometry + Simulation.

Sablonnière, P., 2005. Recent progress on univariate and multivariate polynomial and spline quasi-interpolants, in: Mache, D.H., Szabados, J., de Bruin, M.G. (Eds.), Trends and Applications in Constructive Approximation, Birkhäuser Basel, Basel. pp. 229–245.

Sederberg, T.W., Zheng, J., Bakenov, A., Nasri, A., 2003. T-splines and T-NURCCs. ACM Trans. Graph. 22, 477–484.

Speleers, H., 2017. Hierarchical spline spaces: quasi-interpolants and local approximation estimates. Advances in Computational Mathematics 43, 235–255.

Speleers, H., Manni, C., 2016. Effortless quasi-interpolation in hierarchical spaces. Numerische Mathematik 132, 155–184.

Technical Reports of the Doctoral Program “Computational Mathematics”

2020

2020-01 N. Smoot: *A Single-Variable Proof of the Omega SPT Congruence Family Over Powers of 5* Feb 2020. Eds.: P. Paule, S. Radu

2020-02 A. Schafelner, P.S. Vassilevski: *Numerical Results for Adaptive (Negative Norm) Constrained First Order System Least Squares Formulations* March 2020. Eds.: U. Langer, V. Pillwein

2020-03 U. Langer, A. Schafelner: *Adaptive space-time finite element methods for non-autonomous parabolic problems with distributional sources* March 2020. Eds.: B. Jüttler, V. Pillwein

2020-04 A. Giust, B. Jüttler, A. Mantzaflaris: *Local (T)HB-spline projectors via restricted hierarchical spline fitting* March 2020. Eds.: U. Langer, V. Pillwein

2019

2019-01 A. Seiler, B. Jüttler: *Approximately $\mathcal{C}^1$-smooth Isogeometric Functions on Two-Patch Domains* Jan 2019. Eds.: J. Schicho, U. Langer

2019-02 A. Jiménez-Pastor, V. Pillwein, M.F. Singer: *Some structural results on D^n-finite functions* Feb 2019. Eds.: M. Kauers, P. Paule

2019-03 U. Langer, A. Schafelner: *Space-Time Finite Element Methods for Parabolic Evolution Problems with Non-smooth Solutions* March 2019. Eds.: B. Jüttler, V. Pillwein

2019-04 D. Dominici, F. Marcellán: *Discrete semiclassical orthogonal polynomials of class 2* April 2019. Eds.: P. Paule, V. Pillwein

2019-05 D. Dominici, V. Pillwein: *A sequence of polynomials generated by a Kapteyn series of the second kind* May 2019. Eds.: P. Paule, J. Schicho

2019-06 D. Dominici: *Mehler-Heine type formulas for the Krawtchouk polynomials* June 2019. Eds.: P. Paule, M. Kauers

2019-07 M. Barkatou, A. Jiménez-Paster: *Linearizing Differential Equations Riccati Solutions as D^n-Finite Functions* June 2019. Eds.: P. Paule, M. Kauers

2019-08 D. Dominici: *Recurrence coefficients of Toda-type orthogonal polynomials I. Asymptotic analysis.* July 2019. Eds.: P. Paule, M. Kauers

2019-09 M. Neumüller, M. Schwalsberger: *A parallel space-time multigrid method for the eddy-current equation* Nov 2019. Eds.: U. Langer, R. Ramlau

2019-10 N. Smoot: *An Implementation of Radus Ramanujan-Kolberg Algorithm* Nov 2019. Eds.: P. Paule, S. Radu

2019-11 S. Radu, N. Smoot: *A Method of Verifying Partition Congruences by Symbolic Computation* Dec 2019. Eds.: P. Paule, V. Pillwein

2019-12 J. Qi: *How to avoid collision of 3D-realization for moving graphs* Dec 2019. Eds.: J. Schicho, M. Kauers

The complete list since 2009 can be found at
`https://www.dk-compmath.jku.at/publications/`

Doctoral Program
"Computational Mathematics"

Director:

Dr. Veronika Pillwein
Research Institute for Symbolic Computation

Deputy Director:

Prof. Dr. Bert Jüttler
Institute of Applied Geometry

Address:

Johannes Kepler University Linz
Doctoral Program "Computational Mathematics"
Altenbergerstr. 69
A-4040 Linz
Austria
Tel.: ++43 732-2468-6840

E-Mail:

office@dk-compmath.jku.at

Homepage:

http://www.dk-compmath.jku.at