



Adaptively Refined Multi-Patch B-splines with Enhanced Smoothness

Florian Buchegger, Bert Juettler, Angelos Mantzaflaris

► To cite this version:

Florian Buchegger, Bert Juettler, Angelos Mantzaflaris. Adaptively Refined Multi-Patch B-splines with Enhanced Smoothness. Applied Mathematics and Computation, 2016, 272, pp.159-172. 10.1016/j.amc.2015.06.055 . hal-02272215

HAL Id: hal-02272215

<https://inria.hal.science/hal-02272215>

Submitted on 27 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptively Refined Multi-Patch B-splines with Enhanced Smoothness

Florian Buchegger^{a,*}, Bert Jüttler^a and Angelos Mantzaflaris^b

^aInstitute of Applied Geometry, Johannes Kepler University, Linz, Austria

^bRICAM, Austrian Academy of Sciences, Linz, Austria

Abstract

A spline space suitable for Isogeometric Analysis (IgA) on multi-patch domains is presented. Our construction is motivated by emerging requirements in isogeometric simulations. In particular, IgA spaces should allow for adaptive mesh refinement and they should guarantee the optimal smoothness of the discretized solution, even across interfaces of adjacent patches.

Given a domain manifold \mathcal{M} consisting of individual patches (isomorphic to the unit square or cube) that are glued together along interfaces, we present a construction of multi-patch B-splines defined on them. Their smoothness is enhanced by locally modifying or merging basis functions around the boundary of each patch. The resulting multi-patch B-splines with enhanced smoothness (MPBES) possess the property of local linear independence and form a nonnegative partition of unity. Moreover, their span can be characterized as the linear space of all piecewise polynomial functions on the domain manifold that possess certain smoothness properties.

Subsequently, adaptively refined MPBES are obtained by generalizing the construction of truncated hierarchical (TH) B-splines. More precisely, a nested sequence of spaces spanned by MPBES is considered, corresponding to steps of local enrichment. In addition, an inversely nested sequence of subdomains (which are submanifolds of \mathcal{M}) is used to specify the local refinement level of functions in these spaces. Finally, truncated hierarchical MPBES are obtained by means of the selection and truncation mechanism of THB-splines. The desired properties of linear independence and convex partition of unity are maintained.

The paper presents several numerical examples which demonstrate potential applications of the new basis in isogeometric analysis.

Keywords: isogeometric analysis, multipatch domains, adaptively refined multilevel splines

1. Introduction

The promising concept of Isogeometric Analysis (IgA) is based on the use of multivariate spline functions for representing simultaneously the geometry of a physical domain and the unknown quantities that are to be computed by a numerical simulation [7]. On the

*Corresponding Author. Postal address: JKU, Altenberger Str. 69, 4040 Linz, Austria. Phone: +43 732 2468 4080. Email address: florian.buchegger@jku.at

one hand, this facilitates the exchange of data between software tools used for geometric design and numerical simulation (analysis), since both processes are based on the same mathematical technology. On the other hand, it has been observed that the increased smoothness of the spline functions has a beneficial effect on the accuracy and robustness of the analysis.

Multivariate spline functions can be obtained efficiently by invoking tensor-product constructions, and they are therefore straightforwardly defined on domains that are high-dimensional boxes. However, such domains are not suitable for the representation of more *complex geometric shapes* and the functions defined on them. One may distinguish between two main approaches to address this problem.

The first one uses *several patches* to represent the computational domain, which *are coupled* across the interfaces. Several possibilities for coupling the functions exist. These include the use of mortar methods, FETI-type techniques, Nitsche's method, and the discontinuous Galerkin method [5, 13, 16, 18].

The second approach is based on a *single global parameterization* of the physical domain. This is the case when using T-splines with extraordinary vertices (EVs) where special basis functions have to be used in their vicinity [22]. Isogeometric Analysis based on subdivision splines also falls into this category. Again, the presence of EVs enforces the use of particular basis functions, which are typically piecewise polynomial functions consisting of an infinite number of polynomial segments [6, 27, 29]. It should be noted that the most important subdivision schemes generate relatively simple basis functions (box splines) consisting of a few polynomial pieces in the vicinity of regular (i.e., non-extraordinary) vertices. The use of special basis functions may cause difficulties concerning numerical quadrature or a reduction of the approximation power in the vicinity of EVs.

The interesting concept of spline forests [23] can be seen as a *blend* between the two main approaches. The domain is represented by several patches, but the basis functions along the interfaces are identified to create globally continuous (C^0 -smooth) basis functions.

Besides the need to deal with domains of general shape, another challenge in IgA is the issue of *adaptive refinement*, since this is not trivially supported by tensor-product constructions. Here, two major approaches can be identified.

Firstly, the use of *T-splines* naturally provides the possibility of adaptive refinement [1, 9] though it is necessary to impose restrictions on the mesh to maintain desirable properties such as linear independence [2, 19], which entail certain complications with the refinement algorithm. It should also be noted that the refinability of the special functions at EVs has not yet been investigated.

Secondly, the classical technology of hierarchical B-splines [10, 17] has been employed for IgA and is enriched with new constructions and results [11, 12, 21, 25]. In addition to these two main approaches, there also exist other constructions such as LR-splines or PHT-splines [14, 26].

In our present work we shall extend the hierarchical space construction to general topologies by proper gluing of interfaces. However, our results extend naturally to more general spaces of functions, such as subdivision splines or box splines, and also to more general domains, which are topological manifolds [27, 28, 29].

Our construction, which is inspired by the concept of spline forests, uses a global approach to generate a B-spline basis on multi-patch domains that possesses increased smoothness across interfaces (MPBES). More precisely, we are able to guarantee maximal

smoothness in the interior of the patches and in the kernel of the interfaces, which is the region that is sufficiently far away from vertices and edges in the two- and three-dimensional case, respectively. The use of standard B-splines guarantees optimal approximation power and makes it possible to rely on standard methods for numerical quadrature.

We use recent results from the theory of LR splines [4] to prove the desired properties of local linear independence and partition of unity. Moreover we characterize the span of the MPBES by the smoothness properties of the piecewise polynomial functions contained in it. As a benefit of the local linear independence, we can then invoke the framework of truncated hierarchical generating systems [28] which produces a truncated hierarchical B-spline basis for the space of adaptively refined spline functions on multi-patch domains.

The paper is organized as follows. The concept of domain manifolds and the definition of smooth functions defined on it are recalled in Section 2. The B-splines with enhanced smoothness on multi-patch domains (MPBES) are introduced in the next section, which also studies their properties. Based on these results we use the hierarchical approach and combine it with the truncation procedure to obtain a basis for adaptively refined spline spaces on multi-patch domains in Section 4. Numerical examples illustrate the construction and the properties of MPBES and their hierarchical generalization. Finally we conclude this paper and provide a comparison to spline forests.

2. Domain Manifolds

More complicated domains in IgA cannot be represented by a single geometry mapping, but a collection of several such mappings is required. This is in accordance to the design process, that typically results in a large collection of patches describing a geometric object. We recall the concept of a *domain manifold* (cf. [29]) which is used throughout the paper in order to describe a multi-patch structure.

More precisely, the domain manifold – which may or may not possess a boundary – consists of *patches* (topological quadrilaterals or cubes) that are joined across *interfaces*. Each patch is represented by a parameterization, whose local domain is the n -dimensional unit cube $U = [0, 1]^n$. We denote with $U^\circ = (0, 1)^n$ and $\partial U = U \setminus U^\circ$ the interior and the boundary of the cube. In addition, the set of all points of U with one of their coordinates set to 1 or 0 is called a face of U .

In order to identify the individual patches that form the domain manifold, we introduce a finite index set $\mathcal{P} \subset \mathbb{Z}_+$. We use it to define the *pre-manifold* $U \times \mathcal{P}$, which is the union of all patches $U \times \{p\}$, $p \in \mathcal{P}$.

The adjacency between the patches is defined by identifying faces, as follows. We consider a set of *interface indices* $D \subseteq \mathcal{P} \times \mathcal{P}$. For each pair $(p, p') \in D$ we identify a face of the patches with indices p and p' . We assume that $p \neq p'$. The faces are selected by specifying a displacement. More precisely, for each $(p, p') \in D$ we have an associated displacement $\delta_p^{p'}: \mathbb{R}^n \rightarrow \mathbb{R}^n$. These displacements are isometries, that map one face of U to another face, while the remaining points of U are mapped outside of U . Thus, $\delta_p^{p'}(U) \cap U$ is a face.

For dimension $n = 2$, there are 32 possible displacements, given by a composition of a translation, a rotation and a reflection. The translations put the domain north, south, west or east of the reference domain, where it can be rotated and reflected. Figure 1 shows an instance of a displacement δ , which is a combination of a translation to the east and a rotation by π .

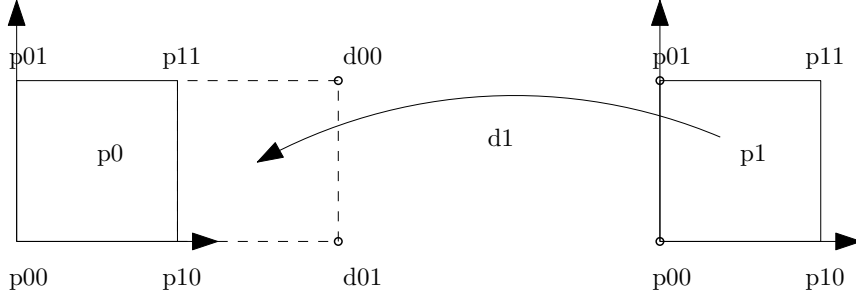


Figure 1: Displacement of a patch p , $n = 2$.

For dimension $n = 3$, there are 288 possible displacements.

Two requirements have to be satisfied by the interface indices and associated displacements. The first requirement (i) ensures that the neighborhood relation is symmetric and possesses compatible displacements. Thus if one patch is a neighbor of another patch, also this other patch is a neighbor of the first patch across the same interface. The second requirement (ii) ensures that every face is identified with at most one other face:

- (i) $(p, p') \in D \Rightarrow (p', p) \in D$ and $\delta_p^{p'} = (\delta_{p'}^p)^{-1}$
- (ii) $\delta_p^{p'}(U) \cap U = \delta_p^{p''}(U) \cap U \Rightarrow p' = p''$

The interface indices and associated displacements specify the *neighbor relation* \simeq on the pre-manifold $U \times \mathcal{P}$. Two points (x, p) and (x', p') , which belong to two different patches, are identified if (p, p') is an interface index and the associated displacement maps x to x' :

$$(x, p) \simeq (x', p') \Leftrightarrow \exists (p, p') \in D : x' = \delta_p^{p'}(x).$$

Clearly, this is only possible if both x and x' are located on the boundary of U .

The *extended neighbor relation* $\hat{\simeq}$ is the transitive, reflexive closure of \simeq . Note that the symmetry is preserved by the closure. Using the fact that $\hat{\simeq}$ is an equivalence relation we define the *domain manifold* \mathcal{M} as the set of equivalence classes,

$$\mathcal{M} = (U \times \mathcal{P}) / \hat{\simeq}.$$

The elements $m \in \mathcal{M}$ are sets of the form $m = \{(x, p), (x', p'), \dots\}$ that possess finitely many elements. It may happen that some faces of patches are not identified with another face. In this case, the domain manifold \mathcal{M} is a manifold with a boundary.

Note that $p \neq p'$ for any two elements (x, p) and (x', p') in $m \in \mathcal{M}$. The points in U° are the only elements in their equivalence class, whereas the points in the interior of the faces generate equivalence classes consisting of two elements, see Fig. 2. Equivalence classes with more than two elements can be present also. In the two-dimensional case ($n = 2$), the number of elements in the equivalence classes of an (inner) corner point of U is equal to its valence.

For later reference we recall the notion of a function on a manifold. A function $f : U \times \mathcal{P} \rightarrow \mathbb{R}$ on the pre-manifold is a function on \mathcal{M} if it takes the same values for equivalent points. Consequently, f has local representations f_p , $p \in \mathcal{P}$, satisfying

$$f : U \times \mathcal{P} \rightarrow \mathbb{R} : (x, p) \mapsto f_p(x)$$

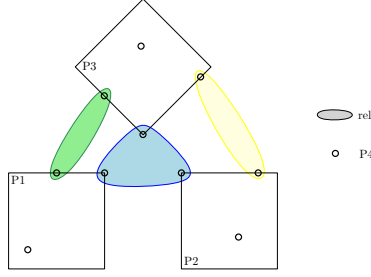


Figure 2: Instances of equivalence classes with 1-3 elements, $n = 2$.

and

$$(x, p) \simeq (x', p') \Rightarrow f(x, p) = f(x', p').$$

Moreover, a function f on \mathcal{M} is *continuous* if

- $\forall p \in \mathcal{P}: f_p|_{U^\circ} \in C(U^\circ)$, so f_p is continuous on each patch, $p \in \mathcal{P}$, and
- the functions f_p , which are defined on U , continuously extend the restrictions $f_p|_{U^\circ}$.

Likewise we also define the notion of higher order smoothness of a continuous function f at certain points (x, p) in the interior of the manifold \mathcal{M} . We do so by considering two cases, depending on the location of the point x :

- If $x \in U^\circ$ is contained in the interior of a patch we use the standard definition,

$$f \text{ is } C^k\text{-smooth at } (x, p) \Leftrightarrow \exists \epsilon: f_p|_{N_\epsilon(x)} \in C^k(N_\epsilon(x)),$$

where $N_\epsilon(x)$ denotes the ϵ -neighborhood of the point $x \in \mathbb{R}^n$.

- Assume that x is located on the boundary ∂U but is contained in the interior of a face, where this face is an interface with another patch with index p' . We define the function \hat{f} as the combination of f_p and $\delta_p^{p'} \circ f_{p'}$, considered on the domain $U \cup \delta_p^{p'}(U)$. Its smoothness determines the smoothness of f at (x, p) ,

$$f \text{ is } C^k\text{-smooth at } (x, p) \Leftrightarrow \exists \epsilon: \hat{f}|_{N_\epsilon(x)} \in C^k(N_\epsilon(x)).$$

It is possible to define higher order smoothness at regular (non extra-ordinary) edges (for $n = 3$) and vertices (for $n = 2, 3$) as well, where the regularity is identified by their valence.

3. Multi-patch B-splines with enhanced smoothness

We define a B-spline basis on the individual patches and then “glue” them together to obtain a global set of basis functions, which are defined on the entire domain manifold. Numerical examples illustrate the properties of this basis.

3.1. B-splines on single patches

In order to define the B-splines on each patch we construct an appropriate LR -mesh, see [3] for more information. Essentially, such a mesh is a *partition* Π of a domain into boxes and a function μ , which specifies the *knot multiplicity* across the interfaces of these boxes. The domain is an axis-aligned box U^+ in \mathbb{R}^n obtained by suitably enlarging the unit cube U .

More precisely, we construct an LR -spline space on the extended local domain $U^+ \subset \mathbb{R}^n$ which will then be restricted to U . First we choose a degree d . To control the dimension of the space, we specify the number of knot-spans in the interval $[0, 1]$ in each dimension to be $\ell \in \mathbb{N}$. Then we denote the density to be $h = \frac{1}{\ell}$. For each choice of d and ℓ we consider the extended domain $U^+ = [-hd, 1 + hd]^n$.

We introduce a partition of the interval $[-hd, 1 + hd]$ into knot spans,

$$Y = \{[hk, h(k+1)]: k = -d, \dots, \ell + d - 1\}.$$

Each knot hj , $j = -d, \dots, \ell + d$, is contained in two knot spans. The boundary knots belong to one knot span only. We then define the partition Π of the whole domain $U^+ = [-hd, 1 + hd]^n$ into boxes:

$$\Pi = \{y_1 \times \dots \times y_n: y_i \in Y\}.$$

These boxes define the set of all interfaces Γ ,

$$\Gamma = \{b \cap b' | b, b' \in \Pi \text{ where } b \neq b' \text{ and } \text{vol}_{n-1}(b \cap b') \neq 0\},$$

where vol_{n-1} denotes the $(n-1)$ -dimensional volume (if $n = 3$, this is the area).

The function μ , which specifies the knot multiplicities across interfaces, is defined on this set, cf. Fig. 3. We wish to use a certain multiplicity μ_0 satisfying $1 \leq \mu_0 < d$ everywhere except for certain interfaces near the boundary, which resemble a “skeleton of boundary”. These interfaces will have knot multiplicity d . In order to simplify the presentation we choose $\mu_0 = 1$.

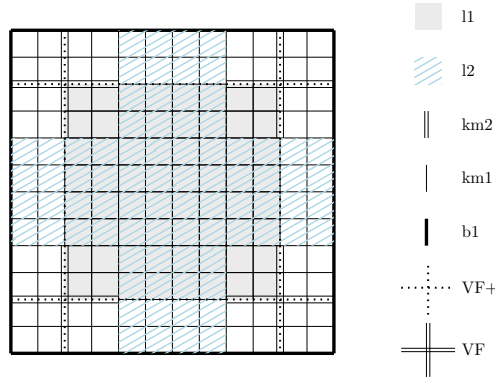


Figure 3: LR-spline after knot insertion, $n = 2$, $d = 2$, $c = 2$.

We define several auxiliary sets that allow us to identify the interfaces with increased knot multiplicity, see Fig. 3. The extended face set F^+ consists of the faces of U and their extensions,

$$F^+ = \bigcup_{\substack{n_1, n_2 \\ n_1 + n_2 = n-1}} \underbrace{\mathbb{R} \times \dots \times \mathbb{R}}_{n_1 \text{ times}} \times \{0, 1\} \times \underbrace{\mathbb{R} \times \dots \times \mathbb{R}}_{n_2 \text{ times}}$$

If we attempt to increase the multiplicity on the set F^+ , we immediately lose C^{d-1} smoothness over the entire interface. Therefore we rather identify the points, where we want to keep the high-order smoothness. To control the size of this set, we have to specify the *smoothness parameter* $c \geq d$ with $c \in \mathbb{N}$. This parameter defines the kernel

$$K = \bigcup_{\substack{n_1, n_2 \\ n_1 + n_2 = n-1}} \underbrace{(ch, 1-ch) \times \cdots \times (ch, 1-ch)}_{n_1 \text{ times}} \times \mathbb{R} \times \underbrace{(ch, 1-ch) \times \cdots \times (ch, 1-ch)}_{n_2 \text{ times}},$$

which covers the interfaces with high-order smoothness. Note that an increase of c decreases the set K that covers the points with high-order smoothness.

The knot multiplicity function μ is defined with the help of the sets F^+ and K ,

$$\mu: \Gamma \rightarrow \mathbb{N}: \quad \gamma \mapsto \mu(\gamma) = \begin{cases} d, & \gamma \subseteq F^+ \setminus K \\ \mu_0 & \text{otherwise.} \end{cases}$$

Figure 3 shows an instance of such an LR-mesh for $n = 2$.

Having defined the LR-mesh, which consists of the partition of the domain into boxes Π and the knot multiplicity function μ on the set of interfaces, we recall how to construct the associated set of LR-splines. First we consider an initial B-spline tensor-product space on the domain by discarding the interior knots (i.e., take Bernstein polynomials on U^+). Then we insert the knot line segments in a suitable ordering and split the basis functions accordingly. More details for this process can be found in [8].

The set of LR B-splines will be denoted by L^+ , since it contains the *local* B-spline basis on the extended domain.

Proposition 1. *The set of LR-splines defined by the LR-mesh (Π, μ) possesses the N_2S property as defined in [4]. Consequently it has the property of local linear independence¹ and forms a non-negative partition of unity.*

Proof. We need to show that the set of LR-splines L^+ possesses the following property:

$$\forall \beta_1, \beta_2 \in L^+: \beta_1 \prec \beta_2 \Rightarrow \beta_1 = \beta_2, \quad (1)$$

where $\beta_1 \prec \beta_2$ is true whenever β_1 is obtained by B-spline refinement of β_2 . This was called the non-nested support (N_2S) property in [4], since there are no LR-splines with nested supports.

If c satisfies $ch \geq \frac{1}{2}$, then the set of LR splines is simply a tensor product B-spline basis, which clearly has this property. The property is also trivially satisfied for dimension $n = 1$. We consider now cases with $n > 1$ and $ch < \frac{1}{2}$.

We split the set L^+ into two disjoint sets. The first one, denoted by L^s , is the set of all functions having only single knots. The second one is denoted with L^m and contains the remaining LR splines. For pairs of functions that belong to one of these sets the property (1) is fulfilled, as they belong to a larger tensor-product B-spline basis. Moreover, no function in L^s can be obtained by LR-refinement of a function contained in L^m , due to the presence

¹A set of functions $(\beta_i)_{i \in \mathcal{I}}$ is said to be linearly independent on an open set Ω if $0 \mid_{\Omega} = \sum_{i \in \mathcal{I}} c_i \beta_i \mid_{\Omega}$ implies $c_i = 0$ whenever the support of β_i overlaps Ω . It has the property of local linear independence if it is linearly independent on *any* open set.

of multiple knots. The remaining case is characterized by $\beta_1 \in L^m$ and $\beta_2 \in L^s$. From [3] we know that

$$\beta_1 \prec \beta_2 \Rightarrow \text{supp } \beta_1 \subseteq \text{supp } \beta_2$$

and that the knot vectors of β_1 can be obtained by inserting knots in the knot vectors of β_2 . If β_1 has multiple knots on the boundary, its knot vectors cannot be obtained in this way from a function in L^s , because inserting boundary knots is impossible. So we need to consider only functions from L_m , which have multiple knots in the interior of the support but no multiple knots on the boundary. Due to our construction, a basis function cannot have more than one multiple knot in each direction. We have to show that the support of β_1 is not strictly contained in β_2 .

First we have a look at functions $\beta_1 \in L_m$, which have multiple knots in all directions. We know that $n > 1$, so the knot lines with multiplicity greater than 1 meet in a point x_C in the inside of $\text{supp}(\beta_1)$. From the construction of the function μ we know, that up to a distance of $c \geq d$ from this point x_C the multiplicity is chosen to be d . So if our function $\beta_2 \in L_s$ contains the support of β_1 , it has to have a multiple knot in its knot vector, which contradicts the fact that it was found in L_s .

Now if we consider a function β_1 , which has s directions without multiple knots, there are only two possibilities. Either at least one of the knot vectors of β_2 of these s directions differs from the corresponding knot vectors of β_1 , which means that $\beta_1 \not\prec \beta_2$, or all knots coincide direction-wise. In the latter case these directions do not influence the argument, which means that the same reasoning applies as in the case of multiple knots in all directions in dimension $n - s$. \square

Figure 4 shows functions β_1 and β_2 which satisfy $\beta_1 \prec \beta_2$ but are not allowed in our setting. Figure 4 (a) is excluded, since the functions were constructed by choosing $c = 1 < d$. Similarly, Figure 4 (b) depicts a case where β_1 has a multiple knot in only one dimension. Also, β_2 is not contained in the basis constructed from Π and μ , as it was split by inserting the multiple knot.

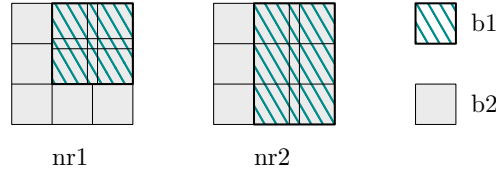


Figure 4: Impossible combinations of β_1 and β_2 , $n = 2$, $d = 2$.

Remark 2. In addition to local linear independence and partition of unity, the construction of the local basis L^+ on the extended domain ensures C^{d-1} -smoothness on $U^\circ \cup (U \cap K)$. More precisely, any function in $\text{span } L^+$ is C^{d-1} -smooth in the interior of the domain U and the derivatives can be extended continuously to $U \cap K$.

All LR splines $\beta \in L^+$ with $\text{supp}(\beta) \cap U^\circ \neq \emptyset$ form the finite *local basis* L on the patch U :

$$L = \{\beta: \beta \in L^+ \text{ and } \text{supp}(\beta) \cap U^\circ \neq \emptyset\}.$$

A two-dimensional example of this local basis is shown in Figure 5, where each basis function is represented by its Greville point. This point is defined as the mean value of the inner local knots for each LR-spline.

Consequently, the functions in $L \times \mathcal{P}$ form the spline space on the union of patches $U \times \mathcal{P}$ that constitute the pre-manifold. These basis functions have the properties stated in Theorem 1.

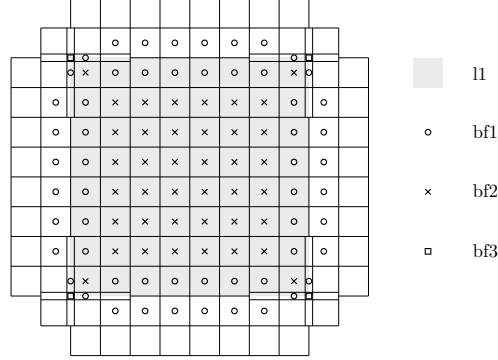


Figure 5: Local basis L , $n = 2$, $d = 2$, $c = 2$ (bf=basis function). The functions are represented by their Greville points.

3.2. Global basis construction and properties

Now we will define a relation on the set of basis functions on the pre-manifold $L \times \mathcal{P}$:

$$(\beta, p) \sim (\beta', p') \Leftrightarrow \beta(x) = (\beta' \circ \delta_p^{p'})(x), \quad \forall x \in \mathbb{R}^n.$$

Again we obtain an equivalence relation \sim by taking the transitive, reflexive closure of \sim . The set of equivalence classes defines the *global basis* B as

$$B = (L \times \mathcal{P}) / \sim.$$

Figure 6 shows an example of basis functions that are identified by the relation \sim in the planar case.

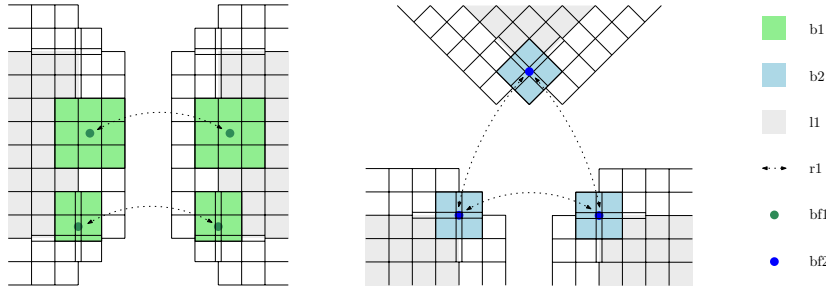


Figure 6: Identifying the basis functions on the boundary, $n = 2$, $d = 2$, $c = 2$.

The basis B consists of elements $\hat{\beta} = \{(\beta, p), (\beta', p'), \dots\}$, where each element is a finite set. These elements can be grouped into three categories. The first category, which consists of *inner basis functions*, contains singletons only. The support of these functions is fully contained within one patch. The second category, which consists of *face basis functions*, contains sets with two elements. The support of these functions is fully contained within two neighboring patches. All other basis functions form the third category.

Note that the restriction $c \geq d$ also implies that $p \neq p'$ holds for any two elements (β, p) and (β', p') in the same equivalence class. Consequently, *each patch cannot contribute more*

If B_1 (resp. B_2) is linearly independent on Ω_1 (resp. Ω_2), then there exist unique coefficients $c_{i,\beta} \in \mathbb{R}$ such that

$$f|_{\Omega_i} = \sum_{\beta \in B_i} c_{i,\beta} \beta|_{\Omega_i}, \quad i = 1, 2$$

and $c_{1,\beta} = c_{2,\beta}$ for all $\beta \in B_1 \cap B_2$.

The proof is implied by the fact that $c_\beta = c_{i,\beta}$ for $\beta \in B_i$, $i = 1, 2$.

Theorem 5. We consider the space S on \mathcal{M} containing all functions that possess the following three properties:

- (i) All functions $f \in S$ are continuous functions on \mathcal{M} ,
- (ii) they are C^{d-1} smooth at all points $(x, p) \in ((K \cap U) \cup U^\circ) \times \mathcal{P}$, and
- (iii) the restrictions of any local representation f_p to an arbitrary element $b \in \Pi$ with $b \subset U$, where Π is the partition of the extended local domains into boxes, is a tensor-product polynomial of degree d .

It follows that the set B forms a basis for the linear space S .

Proof. Given a function $f \in S$, we consider all pairs of neighboring patches \mathcal{D} that share a face. For each pair (p, p') we generate the LR-mesh which is obtained by gluing together the LR-meshes of the two patches, see Figure 8(a). This LR-mesh can simultaneously be seen as a mesh of a hierarchical B-spline with two levels, see again Fig. 8.

The completeness of hierarchical B-splines has been studied in [20]. In particular, the hierarchical spline space is complete if the support of any B-splines basis function of level 0, restricted to $\Omega^0 \setminus \Omega^1$, is a connected set. In our case this condition is satisfied. Since the hierarchical B-splines agree with the LR-splines we obtain a unique representation of the function f on the union of the two patches.

Moreover, due to the local linear independence of the basis and in view of Lemma 4, the coefficients of the representation on the two neighboring patches coincide with the ones in the local representations on each patch separately. Consequently, we are able to find a globally consistent representation of the function f by identifying the matching local coefficients from each patch. \square

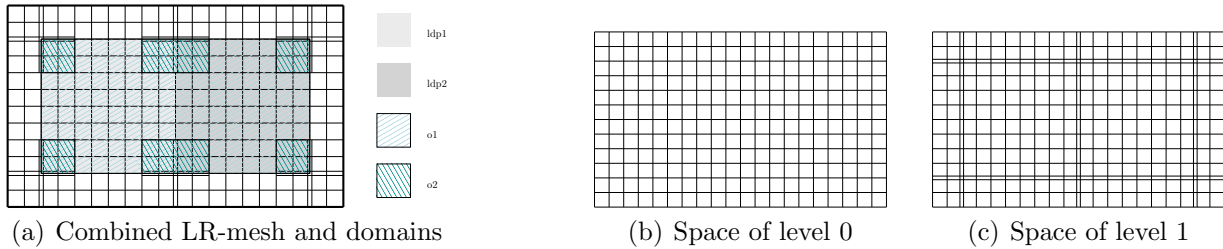


Figure 8: LR-mesh and its corresponding hierarchical mesh, $n = 2$, $d = 2$ and $c = 2$

3.3. Numerical examples

We present three examples to show the possibilities and advantages of this new basis for IgA. First, we look at the parameterization of the smoothed geometry on an L-shaped domain. Second, we solve an elliptic equation on a triangle with an extraordinary vertex. Thirdly, we model a tunnel with a hole, consisting of several patches. The last example will be revisited in the next section to demonstrate mesh adaptivity.

In the theory presented so far, we used reduced smoothness in the vicinity of all $n - 2$ -dimensional interfaces between neighboring patches. This was done mainly to simplify the presentation. In practice, the degree of smoothness can be increased for regular configurations. For instance we can maintain C^{d-1} -smoothness for dimension $n = 2$ for all boundary vertices shared by two patches and for inner vertices possessing valence 4. In addition we can use multiple knots along the boundary of the domain manifold in many cases to obtain interpolating boundary basis functions.

The effect of these modifications is visualized for degree $d = 2$ in Figures 9, 10 and 11. The third example contains an inner vertex with valence 4 where it is possible to maintain C^1 -smoothness.

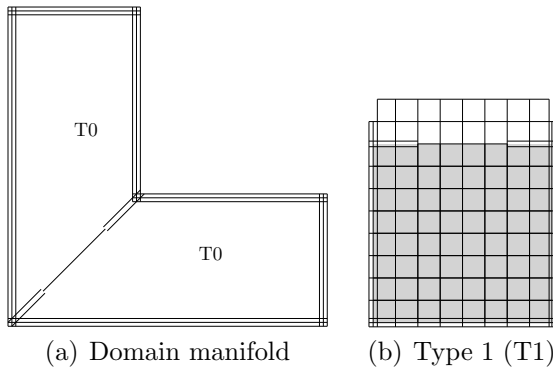


Figure 9: L-shape example with 2 patches of the same type.

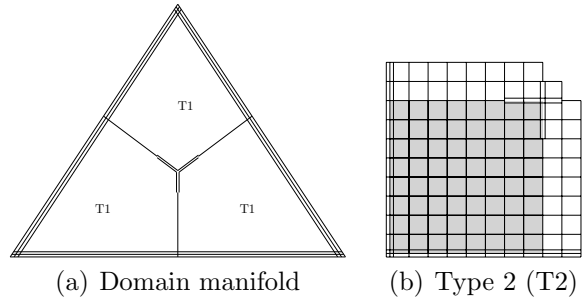


Figure 10: Triangle example with 3 patches of the same type.

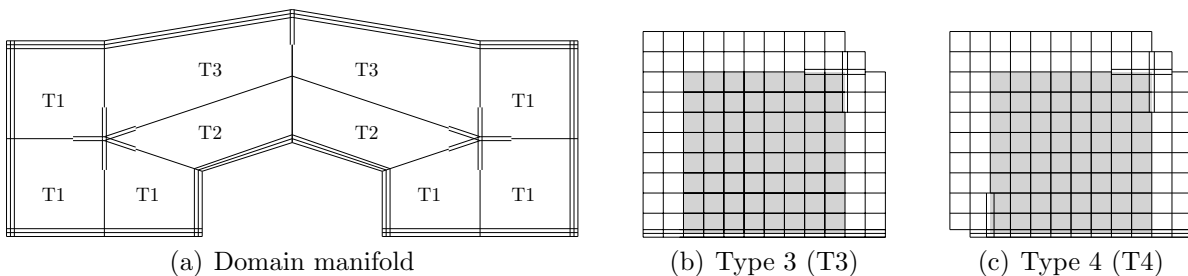


Figure 11: Tunnel example with 10 patches of 3 different types.

We have implemented the new basis for $n = 2$ in the C++ software library **G+SMO**, which provides useful functionality for IgA². After the successful integration of the new basis, we can now use the assembler and solver structure provided there. We also benefit from the existing implementation of (truncated) hierarchical B-splines, which we use in the next section 4 for a hierarchical construction based on our new basis.

²Geometry plus Simulation Modules, gs.jku.at/gismo, see also [15].

L-shape

Our first example resembles an L-shape (see Figure 9), consisting of two quadratic B-spline patches. As the given geometry cannot be represented exactly in the global basis B , we project it into the space S by computing the best approximation with respect to a suitable norm. Such a norm is obtained by summing up contributions from both patches. In this example, we simply used the Euclidean norm for the tensor-product B-spline coefficients on each patch. The projection is performed by solving a sparse system of linear equations. More precisely, it is found by using the QR decomposition of the matrix that expresses the basis B in terms of the tensor-product spline bases of the two patches.

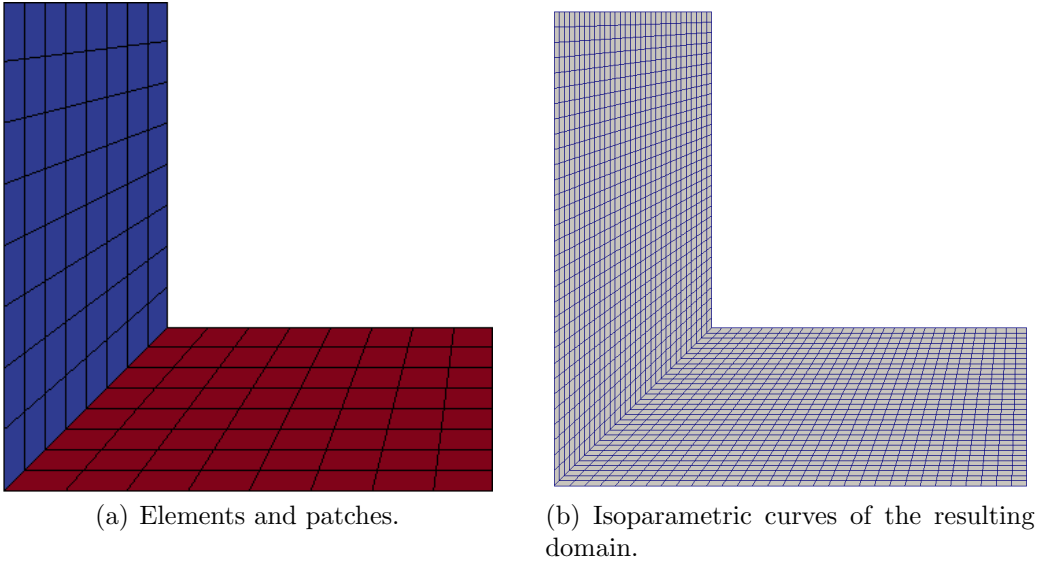


Figure 12: Original parameterization with C^0 multi-patch.

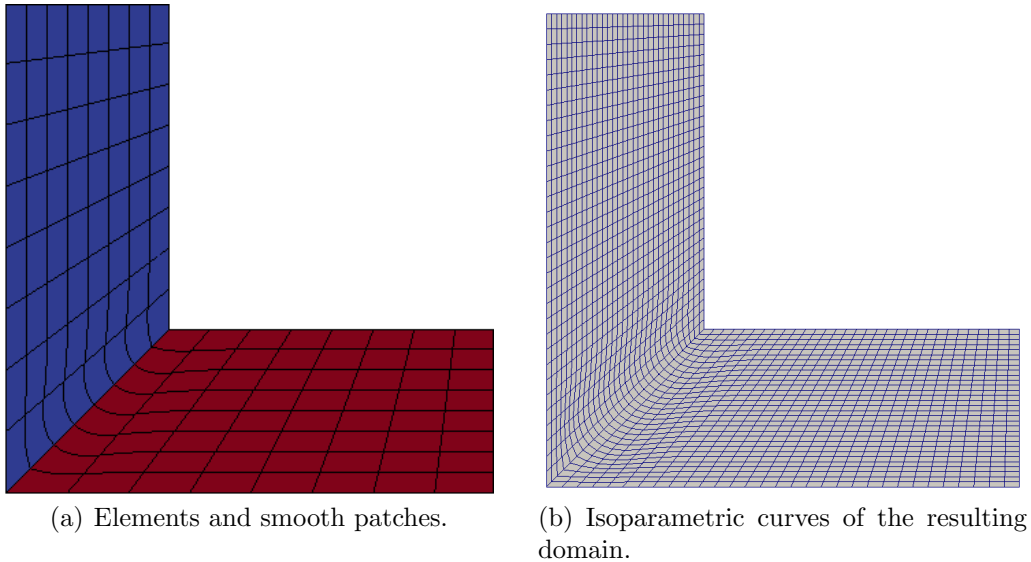


Figure 13: Smooth basis

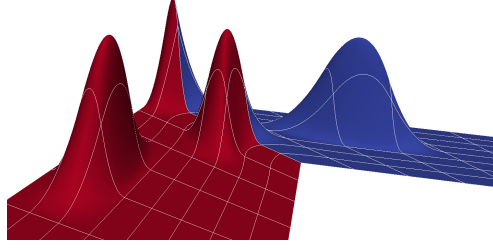


Figure 14: Several basis functions on the smoothed L-shaped domain.

Figure 12 shows the original elements of the basis and a net of isoparametric curves on each patch. Figure 13 shows the elements of the smoothed geometry and again the discretized mesh. We observe that the flow of the mesh is smoother in the second geometry. Except for the vicinity of the C^0 corners, the mesh flows smoother across the interface and the sharp angles, which are present in the original mesh in this area, have disappeared.

Figure 14 shows several basis functions on the domain. We observe the C^0 smoothness of the basis function in the corner, where the knot multiplicities were set to 2. In the middle of the interface, a smooth basis function crosses the common face of the two patches.

Triangle

The domain manifold of our second example consists of three patches, which form a triangle. Figure 10 shows the setup of the patches. In the local basis L , we chose the degree $d = 2$ and $\ell = 9$, which gives us 304 basis functions in the global basis B . The configuration of the patches leads to an extraordinary vertex in the center of the triangle, where C^0 continuity is present. We solve the Poisson equation with Dirichlet boundary conditions for the source function $f = 2 \sin(x) \cos(y)$, which has the exact solution $g = \sin(x) \cos(y)$. The domain and the exact solution are shown in Figure 15.

At the coarsest level we have 304 degrees of freedom. We apply 6 steps of uniform refinement, therefore increasing the number of degrees of freedom up to approx. 1 million. Our experimental results confirm the expected convergence rate 2 in the $H1$ -error (Figure 16(a)) and 3 in the $L2$ -error (Figure 16(b)). Table 1 shows the results of the computations in the global basis B , while Table 2 depicts the same calculations in the C^0 multi-patch setting. We can see that the number of degrees of freedom in the C^0 case is slightly larger and the error with respect to both the norms are slightly lower.

ref. step	0	1	2	3	4	5	6
$L2$ -error	0.010197	0.001010	9.5905e-05	9.9160e-06	1.1024e-06	1.2893e-07	1.5549e-08
$H1$ -error	0.048802	0.010866	0.0024184	0.00057024	0.00013853	3.4153e-05	8.4796e-06
dof	304	1093	4129	16033	63169	250753	999169
times (sec)	0.0087361	0.033335	0.14523	0.70412	4.2031	23.442	161.91

Table 1: Solving the Poisson equation on the smooth triangle with uniform refinement (dof = number of degrees of freedom).

The effect of the increased smoothness of the new basis on the calculated solution is visualized in Figure 17. The left subfigure shows the jump of the derivative in the first coordinate along all the interfaces of the solution obtained using the C^0 smooth basis as opposed to the right subfigure, where we observe this jump only in a vicinity of the extraordinary vertex.

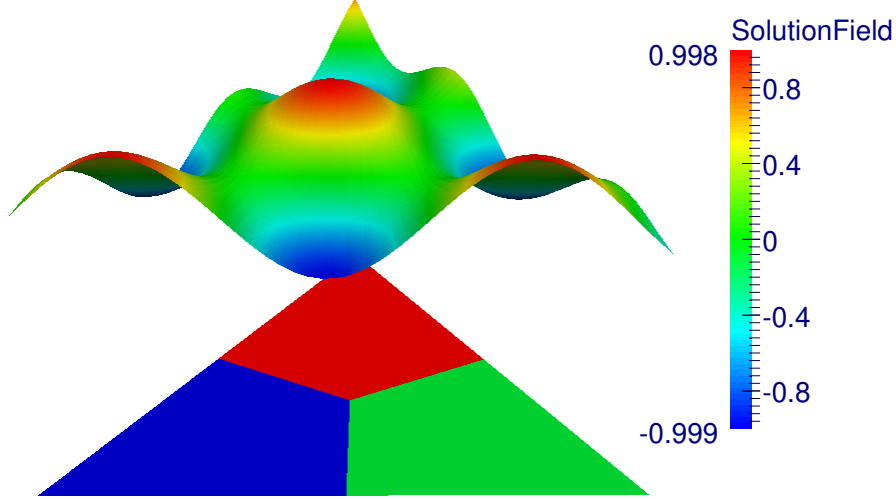


Figure 15: Domain and Solution of the Triangle Example

ref. step	0	1	2	3	4	5	6
$L2$ -error	0.0053265	0.00072434	7.2581e-05	8.2608e-06	9.9254e-07	1.2194e-07	1.5124e-08
$H1$ -error	0.03962	0.010079	0.0023329	0.00056337	0.00013859	3.4385e-05	8.5648e-06
dof	331	1141	4219	16207	63511	251431	1000519
times (sec)	0.018511	0.059474	0.25062	1.4420	8.9197	34.760	163.18

Table 2: Solving the Poisson equation on the C^0 triangle with uniform refinement (dof = number of degrees of freedom).

The computations were done on 64-Bit Gentoo Linux workstation with an Intel i5-4570 quad core CPU with 3.20 GHz and 32 GB of memory. The total processing time (for assembling and solving linear system) are shown in Table 1 for the basis B with increased smoothness and in Table 2 for the C^0 case.

By comparing the computing times in Tables 1 and 2 we observe that one achieves substantially smoother results while keeping the times in the same order of magnitude as in the C^0 case.

4. Hierarchical multi-patch B-splines

In this section we present the extension of our construction to hierarchical B-splines, which support adaptive refinement.

4.1. Construction and properties

Given a domain manifold \mathcal{M} and a degree d we can construct a sequence of nested spline spaces S^L , which is spanned by multi-patch B-splines B^L with enhanced smoothness. The

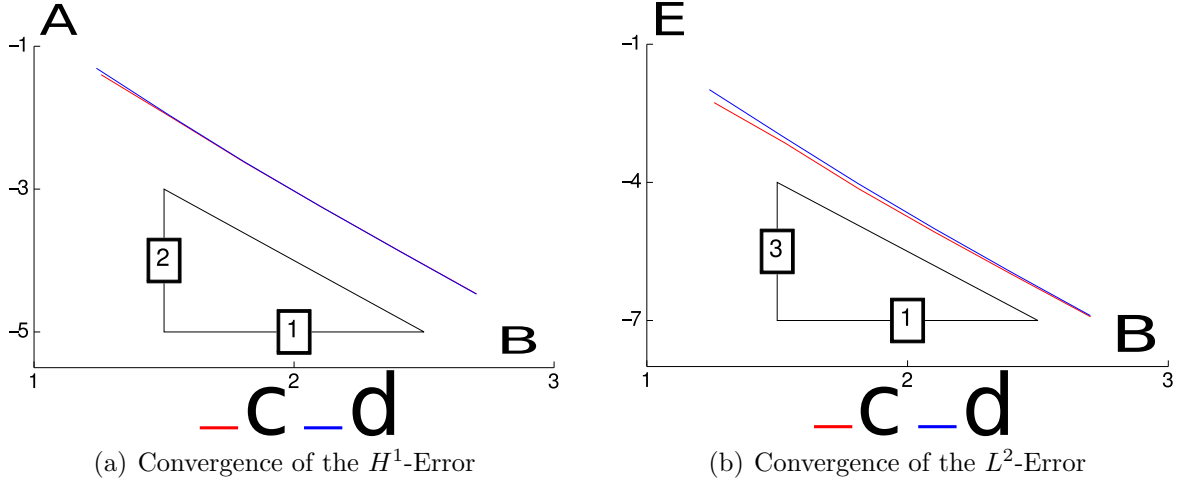


Figure 16: Convergence plots

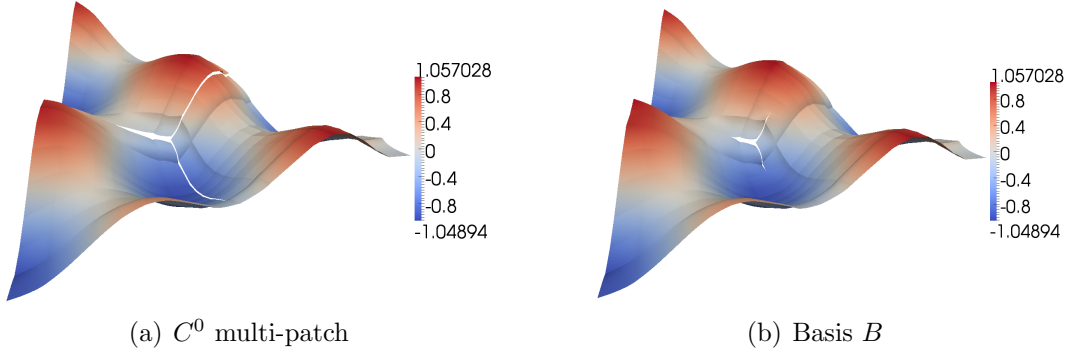


Figure 17: Derivative of the numerical solution with respect to the first coordinate

upper index refers to the level of refinement. For the coarsest level $L = 0$ we choose valid parameters ℓ^0 and c^0 . The parameters for higher levels are obtained by doubling these initial values in each refinement step, i.e., $\ell^L = 2^L \ell^0$ and $c^L = 2^L c^0$. This choice of the parameters leads to nested spaces $S^L = \text{span } B^L$, i.e. $S^{L+1} \supset S^L$. Moreover, the multi-patch B-splines are linked by non-negative refinement matrices. Consequently we can apply the theory presented in [11, 28, 29], as follows:

- The (truncated) hierarchical generating system is defined by specifying a finite, inversely nested (i.e. $\Omega^L \supseteq \Omega^{L+1}$) sequence of subdomains $(\Omega^L)_{L=0,\dots,N}$, which are submanifolds of \mathcal{M} , where $\Omega^0 = \mathcal{M}$ and $\Omega^N = \emptyset$. These subdomains specify the desired adaptive refinement; the latter is typically driven by an *a posteriori* error estimator.
- The hierarchical generating system \mathcal{H} is obtained by collecting all multi-patch B-splines $\beta^L \in B^L$ of all levels L that satisfy

$$\Omega^L \supseteq \text{supp } \beta^L \not\subseteq \Omega^{L+1}.$$

This selection mechanism was established in [17].

- The truncated hierarchical generating system \mathcal{T} is constructed by applying the truncation procedure to the hierarchical generating system \mathcal{H} . This is described in detail in [11, 28].
- Since the bases B^L possess the property of local linear independence, both generating systems \mathcal{H} and \mathcal{T} are linearly independent, hence they are bases of the adaptively refined multilevel spline space on the domain manifold \mathcal{M} .
- Moreover, since the bases B^L form non-negative partitions of unity, the truncated hierarchical generating system has this important property as well.

4.2. Numerical example

The example resembles a tunnel consisting of 10 patches. Figure 11 depicts the arrangement and types of the patches.

We consider again the Poisson equation with the source function

$$f = 10 \tanh(18 - x - 2y)(1 - \tanh(18 - x - 2y) \tanh(18 - x - 2y))$$

and we enforce Dirichlet boundary conditions that correspond to the exact solution $g = \tanh(18 - x - 2y)$. The solution has a very sharp feature, which we expect to be captured by adaptive refinement. The domain and the exact solution are shown in Figure 18.

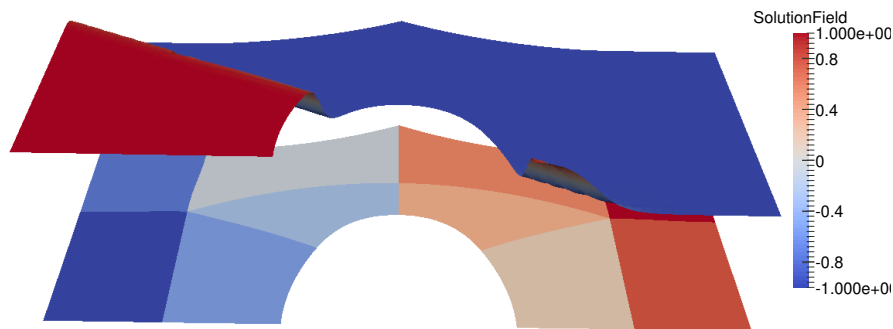


Figure 18: Domain and solution of the tunnel example

We perform seven steps of adaptive refinement, guided by computing the error between the exact and discretized solutions. In each step we refine all elements where the local error exceeds 50% of the maximum error for this solution. Clearly, the number of refined elements varies in each step, depending on the distribution of this error.

Figure 19 shows the advantage of the adaptive refinement with respect to uniform refinement. The use of adaptive refinement significantly increases the rate of convergence with respect to the number of degrees of freedom. The final control grid, which is obtained after seven steps of adaptive refinement is shown in Figure 20.

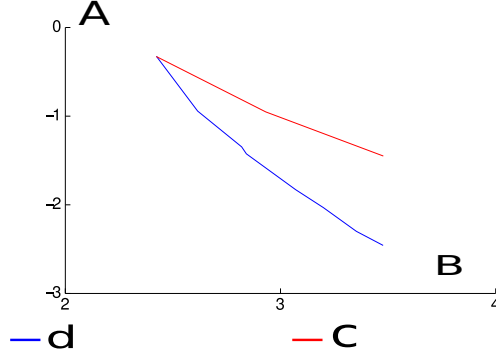


Figure 19: Convergence plot

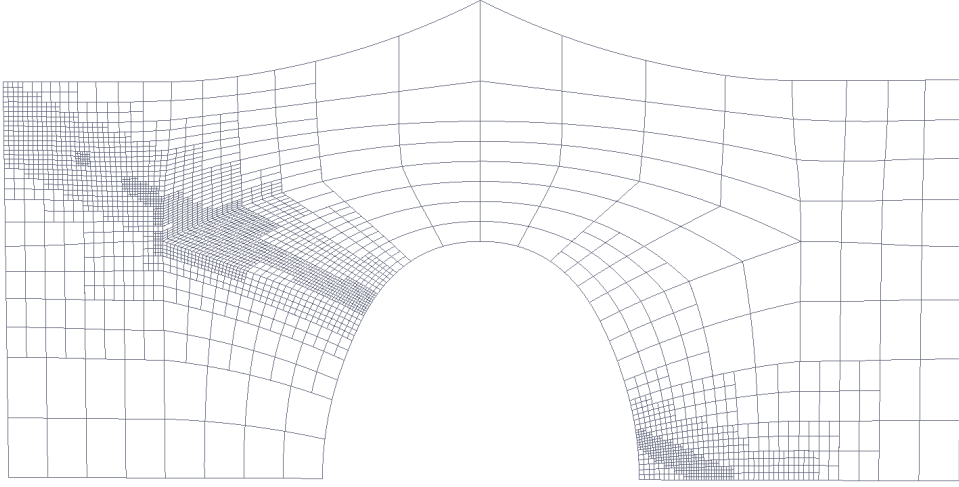


Figure 20: Final control grid after 7 adaptive refinement steps

For this example, the computations were done on a laptop with 64-Bit Debian Linux with an Intel i7-4500 quad core CPU with 1.80 GHz and 16 GB of memory. The computation times, the degrees of freedom and the L^2 and H^1 errors of the seven adaptive refinement steps are reported in Table 3.

ref. step	0	1	2	3	4	5	6	7
L^2 -error	0.1048	0.008024	0.001855	0.001370	0.0003954	0.0001793	7.829e-05	5.152e-05
H^1 -error	0.4678	0.114	0.04503	0.03759	0.01481	0.00928	0.005020	0.00350
dof	266	412	660	694	1176	1584	2252	2985
times (sec)	0.08644	0.1873	0.4209	0.4573	0.9357	1.34	2.139	3.119

Table 3: Solving the Poisson equation on the tunnel with adaptive refinement (dof=number of degrees of freedom).

Figure 21 shows several instances of basis functions on the domain. In particular, in Figure 21(a), a truncated basis function of level 0 is visualized as well as its non-truncated counterpart in Figure 21(b). Moreover, a smooth function crossing the interface and function of third category (see Subsection 3.2) at an extraordinary vertex possessing valence five is depicted.

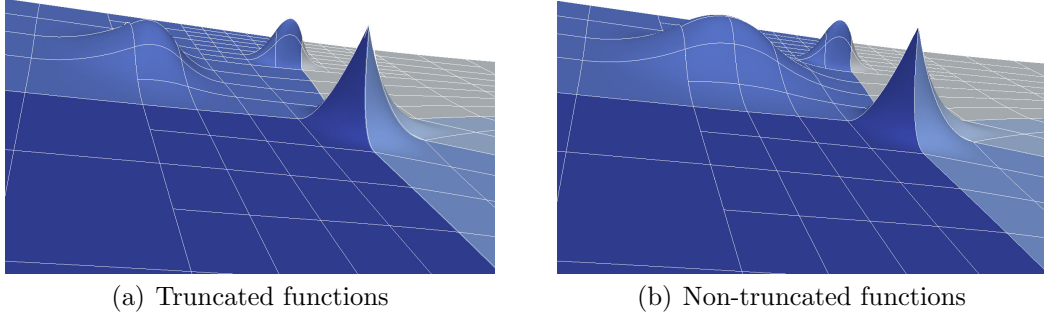


Figure 21: Several instances of basis functions on the tunnel domain after two local refinements

5. Conclusion

Inspired by the concept of spline forests [23] we constructed the new basis of multi-patch B-splines with enhanced smoothness (MPBES) across interfaces. Based on recent results from the theory of LR splines [4] we were able to prove the local linear independence and the partition of unity properties. In addition, we derived a characterization of the space spanned by MPBES by the smoothness properties of the functions contained in it. Finally, relying on the theoretical results concerning local linear independence, we were able to invoke the framework of truncated hierarchical generating systems [28]. This led us to define the truncated hierarchical B-spline basis for the space of adaptively refined spline functions on multi-patch domains. While the theory was presented for the general n -dimensional case, our implementation in the **G+SMO** library [15] and the numerical examples presented in the paper are currently restricted to the two-dimensional setting.

Compared to multi-patch B-splines that are only C^0 -smooth across interfaces (which correspond to isogeometric spline forests), the use of MPBES substantially improves the smoothness properties of the solutions (cf. Figure 17) and of the meshes obtained by discretizing the geometry mappings (see Fig. 13). Surprisingly, the additional computational costs for maintaining the enhanced smoothness are negligible. In fact, the smoothness enhancement reduces the total number of basis functions, thereby slightly reducing the overall computation time, without much impact on the approximation power.

Future work will focus on the extension of our approach to basis functions that possess increased smoothness everywhere. Clearly, this requires to define the smoothness of the functions with respect to the physical domain. Another interesting topic is the generalization to other classes of multivariate splines that provide the possibility of local refinement, such as T-splines [24] and LR splines [8]. The extension of the spline forest approach to T-splines has been already announced in [23], and it would certainly be promising to use higher smoothness across patch interfaces also in this framework. Similar techniques might become available for LR splines too. Clearly, in order to extend the applicability of the specific constructions, it is also interesting to investigate algorithms that convert the different representations of piecewise polynomial functions into each other or into a standard CAD representation, which is currently provided by rational tensor-product splines. The development of such algorithms is expected to benefit from the existing results about the completeness of these spline spaces, see e.g. [20].

Acknowledgement

Supported by the European Commission through FP7, Project EXAMPLE, GA no. 324340, and by the Austrian Science Fund (FWF), NFN S117 “Geometry+Simulation”.

References

References

- [1] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, and T. Sederberg. Isogeometric analysis using T-splines. *Comp. Meth. Appl. Mech. Engrg.*, 199(5–8):229–263, 2010.
- [2] L. Beirão da Veiga, A. Buffa, D. Cho, and G. Sangalli. Analysis-suitable T-splines are dual-compatible. *Comp. Meth. Appl. Mech. Engrg.*, 249–252(0):42–51, 2012.
- [3] A. Bressan. Some properties of LR-splines. *Comp. Aided Geom. Design*, 30(8):778–794, 2013.
- [4] A. Bressan and B. Jüttler. A Hierarchical Construction of LR Meshes in 2D. Technical Report 23, NFN Geometry + Simulation, 2015. www.gs.jku.at/gs_pub.shtml.
- [5] E. Brivadis, A. Buffa, B. Wohlmuth, and L. Wunderlich. Isogeometric mortar methods. *Comp. Meth. Appl. Mech. Engrg.*, 284:292–319, 2015.
- [6] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Int. J. Numer. Meth. in Engrg.*, 47(12):2039–2072, 2000.
- [7] J. Cottrell, T. Hughes, and Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.
- [8] T. Dokken, T. Lyche, and K. Pettersen. Polynomial splines over locally refined box-partitions. *Comp. Aided Geom. Design*, 30(3):331–356, 2013.
- [9] M. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h -refinement with T-splines. *Comp. Meth. Appl. Mech. Engrg.*, 199:264–175, 2010.
- [10] D. Forsey and R. Bartels. Hierarchical B-spline refinement. In *Proc. Siggraph*, pages 205–212, New York, 1988. ACM.
- [11] C. Giannelli, B. Jüttler, and H. Speleers. THB-splines: the truncated basis for hierarchical splines. *Comp. 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] Y. Guo and M. Ruess. Nitsche’s method for a coupling of isogeometric thin shells and blended shell structures. *Comp. Meth. Appl. Mech. Engrg.*, 284:881–905, 2015.
- [14] K. A. Johannessen, T. Kvamsdal, and T. Dokken. Isogeometric analysis using LR B-splines. *Comp. Meth. Appl. Mech. Engrg.*, 269:471–514, 2013.

- [15] B. Jüttler, U. Langer, A. Mantzaflaris, S. E. Moore, and W. Zulehner. Geometry + simulation modules: Implementing isogeometric analysis. *PAMM*, 14(1):961–962, 2014.
- [16] S. Kleiss, C. Pechstein, B. Jüttler, and S. Tomar. IETI – Isogeometric tearing and interconnecting. *Comp. Meth. Appl. Mech. Engrg.*, 247-248:201–215, 2012.
- [17] R. Kraft. Adaptive and linearly independent multilevel B-splines. In *Surface fitting and multiresolution methods*, pages 209–218. Vanderbilt University Press, Nashville, 1997.
- [18] U. Langer, A. Mantzaflaris, S. Moore, and I. Touloupoulos. Multipatch discontinuous Galerkin isogeometric analysis. Technical Report 18, NFN Geometry + Simulation, 2014. available at www.gs.jku.at.
- [19] X. Li, J. Zheng, T. W. Sederberg, T. Hughes, and M. Scott. On linear independence of T-spline blending functions. *Comp. Aided Geom. Design*, 29:63–76, 2012.
- [20] D. Mokriš, B. Jüttler, and C. Giannelli. On the completeness of hierarchical tensor-product B-splines. *J. Comp. Appl. Math.*, 271:53–70, 2014.
- [21] D. Schillinger, J. Evans, A. Reali, M. Scott, and T. Hughes. Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations. *Comp. Meth. Appl. Mech. Engrg.*, 267:170–232, 2013.
- [22] M. Scott, R. Simpson, J. Evans, S. Lipton, S. Bordas, T. Hughes, and T. Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Comp. Meth. Appl. Mech. Engrg.*, 254:197–221, 2013.
- [23] M. Scott, D. Thomas, and E. Evans. Isogeometric spline forests. *Comp. Meth. Appl. Mech. Engrg.*, 269:222–264, 2014.
- [24] T. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and t-nurccs. *ACM Trans. Graph.*, 22(3):477–484, 2003.
- [25] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comp. Meth. Appl. Mech. Engrg.*, 200:3554–3567, 2011.
- [26] P. Wang, J. Xu, J. Deng, and F. Chen. Adaptive isogeometric analysis using rational PHT-splines. *Comp.-Aided Design*, 43(11):1438–1448, 2011. Solid and Physical Modeling 2011.
- [27] X. Wei, Y. Zhang, T. Hughes, and M. Scott. Truncated hierarchical Catmull-Clark subdivision with local refinement. Technical Report 14-31, ICES, University of Texas at Austin, September 2014.
- [28] U. Zore and B. Jüttler. Adaptively refined multilevel spline spaces from generating systems. *Comp. Aided Geom. Design*, 31(7–8):545–566, 2014.
- [29] U. Zore, B. Jüttler, and J. Kosinka. On the linear independence of (truncated) hierarchical subdivision splines. Technical Report 17, NFN Geometry + Simulation, 2014. www.gs.jku.at/gs_pub.shtml.