Enhancing Isogeometric Analysis by a Finite Element-Based Local Refinement Strategy

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

[a] *Institute of Applied Geometry, Johannes Kepler University Linz, Faculty of Natural Sciences and Engineering, Altenberger Straße 69, 4040 Linz, Austria*
[b] *Institute of Computational Mathematics, Johannes Kepler University Linz, Faculty of Natural Sciences and Engineering, Altenberger Straße 69, 4040 Linz, Austria*

Abstract

While isogeometric analysis has the potential to close the gap between Computer Aided Design and Finite Element Methods, the underlying structure of NURBS (non-uniform rational B-splines) is a weakness when it comes to local refinement. We propose a hybrid method that combines a globally C^1-continuous, piecewise polynomial finite element basis with rational NURBS-mappings in such a way that an isoparametric setting and exact geometry representation are preserved. We define this basis over T-meshes with a hierarchical structure that allows locally restricted refinement. Combined with a state-of-the-art a posteriori error estimator, we present an adaptive refinement procedure. This concept is successfully demonstrated with the Laplace equation, advection-diffusion problems and linear elasticity problems.

Keywords: Isogeometric analysis, NURBS, adaptive refinement, local refinement, T-mesh

1. Introduction

Geometry representations in Computer Aided Design (CAD) software and Finite Element Method (FEM) have been developed independently and differently. Hence, when FEM is applied to domains constructed with CAD tools, which is often the case in practice, a transformation of the geometry data is necessary. One issue in this context is whether these transformations can be carried out automatically and efficiently, or if manual user interaction is necessary, and if so, to what extent. Another issue is that transforming geometry data is bound to require some approximations, which results in inaccuracies and geometry errors.

In classic FEM, the geometry is approximated by polygonal, i.e. by piecewise linear elements. It is obvious that these elements fail to exactly represent curved geometries and that a good approximation of such geometries might require an undesirably high number of elements. With higher-order isoparametric FEM it is possible to obtain a better geometry approximation by using elements that may have curved boundaries. Either way, the geometry representation remains inexact, and refining complicated parts of the boundary of the computational domain requires costly communication with the CAD programme in order to preserve the best possible geometry representation.

Isogeometric analysis (IgA) was introduced by Hughes et al. [21] in order to overcome these difficulties. It is based on the fact that it is a standard in CAD to use Non-Uniform Rational B-Splines (NURBS) as basis functions for the geometry representation, and that these NURBS have properties that make them suitable as FEM shape functions. The direct use of geometry data from CAD eliminates the need for costly and inaccurate data transformation. Exact geometry representation is given in the first, coarsest mesh, and preserved throughout the refinement process, which makes further communication with the CAD programme unnecessary.

*Corresponding author, Tel.: +43 732 2468 1671, Fax: +43 732 2468 29162

The potential of this approach has already been shown by applying IgA to, for example, plane linear incompressible elasticity [3], gradient elasticity [19], structural vibrations [14], electromagnetics [10], arterial blood flows [7, 37], incompressible turbulent flows [1], or trimmed CAD surfaces [23]. Efficient quadrature for NURBS-based IgA has been investigated by Hughes et al. [22], locking-phenomena of NURBS-based finite elements by Echter and Bischoff [18], and the robustness of IgA under severe mesh distortion by Lipton et al. [26].

NURBS can be refined by *knot insertion*, which corresponds to h-refinement in FEM, or by *order elevation*, which corresponds to p-refinement in FEM. As an additional, NURBS-specific method, a combination of these two can be applied, namely *k-refinement* (see, e.g. [13, 21]). Cottrell et al. [13] compared h-, p-, and k-refinement and showed examples where k-refinement produced a significantly better accuracy on degree-of-freedom basis than the other methods. When it comes to adaptive, i.e. local refinement, however, NURBS show a weakness, because they are defined over a grid-like structure that can not be refined locally.

In the context of geometry modeling, the discussion of local refinement of B-splines dates back to Forsey and Bartels [20]. Sederberg et al. [32, 33] introduced *T-splines* for local refinement of rational spline surfaces and gap-free merging of several spline surfaces with different knot vectors. The application of T-splines in IgA has been investigated by Bazilevs et al. [5], Dörfel et al. [17], and Uhm and Youn [36] and the results show that T-splines have the potential to provide a tool for local refinement. One drawback of T-splines is that a specific structure of the underlying mesh has to be preserved (see [5, 17, 32, 33] for details). In special cases, this might require the insertion of additional knots in the vicinity of the actual area that has to be refined. In a worst-case example shown in [17], these additional knots trigger a snowball-effect leading to almost global refinement and thereby jeopardizing the gained advantages.

Another question is that of linear independence of T-splines and the characterization of the corresponding function spaces. Under certain smoothness conditions, Deng et al. [15] provide a formula for the dimension of the spanned spline space over T-meshes. Buffa et al. [9] discuss a very specific refinement process that ensures linear independence of T-splines, but also give an example of linearly dependent T-splines.

Sufficient and necessary conditions for linear independence of T-splines have been presented recently by Li et al. [25]. Recent work [24, 31] shows that linear independence and a locally restricted refinement can be obtained by using the restricted set of *analysis suitable T-splines*.

Hughes et al. [21] already mention that, while NURBS are in most widespread use, they are "not a requisite ingredient in isogeometric analysis". For example, Manni et al. [27] propose the use of generalized B-splines as an alternative to NURBS-based IgA. Deng et al. [16] introduced polynomial splines over hierarchical T-meshes (PHT-splines), which allow local refinement. Recently, Nguyen-Thanh et al. [29] applied piecewise bicubic, globally C^1-continuous PHT-splines to two-dimensional elasticity problems in the isogeometric framework.

Similar to the concept in [29], we propose the use of basis functions which are piecewise bicubic and globally C^1-continuous on the parameter domain, and which are transformed to the physical domain by the NURBS geometry mapping. We define a hierarchical T-mesh with hanging nodes that is structured by relations between hanging nodes at T-shaped junctions and primary nodes at X-shaped junctions.

As a difference to [29], we use a nodal-based representation of the basis functions. We extend the Bogner-Fox-Schmit rectangle to hierarchical T-meshes (this element was also considered by Stogner et al. [34]). This way, the procedure for obtaining the local representation of a function is straight-forward and requires only a minimum of information to be stored, which makes it convenient to implement. The refinement of this T-mesh is restricted locally, without any unwanted fill-in effects, and linear independence of the used basis functions is guaranteed throughout the refinement process. Compared to analysis-suitable T-splines (see again [24, 25, 31]), our method provides a more local refinement with a simpler refinement algorithm. Indeed, the refinement of analysis-suitable T-splines still requires additional knot insertions beyond the ones marked by the error estimator. On the other hand, our approach suffers from a reduced smoothness of the spline space.

This finite element basis allows us to combine it with the a posteriori error estimator discussed by Dörfel et al. [17]. Hence, we present an adaptive local refinement procedure.

Instead of a simple push-forward to the physical domain, we incorporate the denominator of the rational NURBS-mapping into the transformation of the basis functions, such that an isoparametric setting is

preserved. The benefits of IgA with regard to exact geometry representation on the first, coarsest mesh are inherited and also preserved.

The proposed method is tested with advection-dominated advection-diffusion problems, the Laplace equation, and linearized elasticity problems.

The remainder of this paper is organized as follows: Hierarchical T-meshes and additional preliminary work are discussed in Section 2. The proposed hybrid method is defined in Section 3. The applied adaptive refinement procedure and the a posteriori error estimator are described in Section 4. Finally, the results of numerical tests are presented in Section 5, and a conclusion is presented in Section 6.

2. Preliminaries

As preliminary work, we describe hierarchical T-meshes. In order to fix notation and for the sake of completeness, we also define B-spline- and NURBS-mappings, and we recall and compare some main aspects of IgA and classic FEM.

2.1. Hierarchical T-Meshes

We define a *T-mesh* $\mathcal{T}$ as a subdivision of a rectangular domain $Q \subset \mathbb{R}^2$ into rectangular, non-degenerated, axis-aligned *cells* K by a grid that may possess T-junctions such as illustrated in the example in Fig. 1. The gridpoints of the mesh are called *nodes* and denoted by $\nu^{(i)}$. The gridlines are called *edges*, and the edge from node $\nu^{(i)}$ to $\nu^{(j)}$ will be denoted by $(\nu^{(i)}\nu^{(j)})$.

Nodes on the boundary of Q are called *boundary nodes* (e.g. $\nu^{(1)}$, $\nu^{(2)}$, and $\nu^{(5)}$ in Fig. 1), all other nodes are called *interior nodes* (e.g. $\nu^{(6)}$, $\nu^{(7)}$, and $\nu^{(8)}$ in Fig. 1). Except for the nodes at the corners of the domain Q, we require all nodes to be the vertices of at least three edges, i.e. we do not allow L-shaped junctions, nodes on straight lines, edges with loose ends, or isolated nodes. We distinguish between the following two main types of nodes:

- *Hanging nodes*: Interior nodes at T-shaped junctions (e.g. $\nu^{(7)}$, $\nu^{(8)}$, $\nu^{(9)}$, and $\nu^{(10)}$ in Fig. 1).
- *Primary nodes*: Boundary nodes and interior nodes at X-shaped junctions (such as, e.g. $\nu^{(1)}$, $\nu^{(2)}$, $\nu^{(6)}$, and $\nu^{(14)}$ in Fig. 1).

  The set of indices of primary nodes will be denoted by $\mathcal{I}_X$, their total number by N_X.

We define a *principal edge (p-edge)* as the longest possible line segment that only contains hanging nodes in its interior. We then define the two endpoints of a p-edge to be the *parent nodes* of all hanging nodes contained in the p-edge. For example, in Fig. 1, $(\nu^{(11)}\nu^{(14)})$ is a p-edge; $\nu^{(11)}$ and $\nu^{(14)}$ are the parent nodes of $\nu^{(12)}$ and $\nu^{(13)}$. $(\nu^{(6)}\nu^{(12)})$ is also a p-edge, while $(\nu^{(12)}\nu^{(13)})$ and $(\nu^{(14)}\nu^{(19)})$ are not.

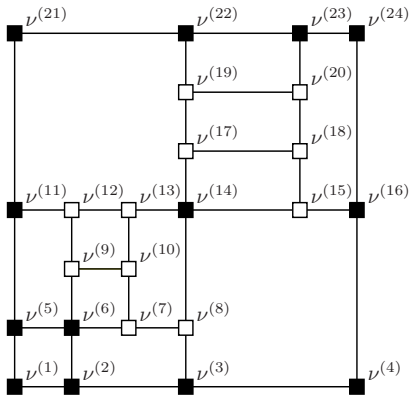


Figure 1: Example of hierarchical T-mesh (obtained from mesh in Fig. 2(d) by inserting edge $(\nu^{(9)}\nu^{(10)})$).
■ primary node, □ hanging node.

We focus on a specific class of T-meshes, namely *hierarchical T-meshes*, which we define as follows: We start from an initial mesh $\mathcal{T}_0$ of Q consisting of rectangular, axis-aligned cells *without* hanging nodes, such as the one in Fig. 2(a). We then refine cells by *edge insertion*: We split a cell into two smaller cells by inserting one axis-aligned edge as illustrated in Fig. 2. The hierarchical T-mesh is structured by assigning *levels* to nodes and requiring the following to be fulfilled:

(T1) The level of a primary node is always 1.
(T2) The level of a hanging node is always higher than the levels of its two parent nodes.

For example, in Fig. 1, the hanging nodes are structured as follows:
Level 2: $\nu^{(8)}$, $\nu^{(12)}$, $\nu^{(13)}$, $\nu^{(15)}$, $\nu^{(17)}$, $\nu^{(19)}$.
Level 3: $\nu^{(7)}$, $\nu^{(9)}$, $\nu^{(18)}$, $\nu^{(20)}$.
Level 4: $\nu^{(10)}$.

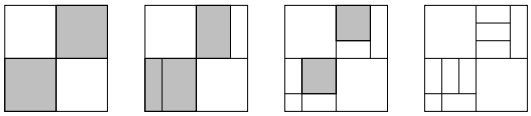


Figure 2: Example of hierarchical T-mesh and three steps of edge insertions into the shaded cells.

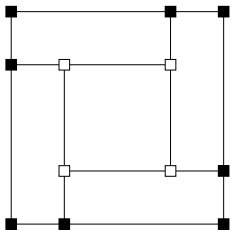


Figure 3: Non-hierarchical T-mesh. (T2) is violated by circular dependencies of hanging nodes marked by □.

Fig. 3 shows an example of a T-mesh that is not hierarchical. The circular dependencies of the hanging nodes cause a violation of (T2).

In the initial subdivision, in which we only have primary nodes, we assign level 1 to every node. If a new hanging node is created during refinement, (T2) is easily fulfilled by assigning any level that is higher than the levels of its parent nodes (a natural choice would be to add 1 to the higher one of the two parent nodes' levels). If an already existing hanging node is changed into a primary node, we assign level 1 to it, thereby fulfilling (T1) while (T2) remains valid. Note that, if a p-edge $(\nu^{(i)}\nu^{(j)})$ contains a hanging node $\nu^{(k)}$, and this node is changed into a primary node, then the p-edge is split into two smaller p-edges $(\nu^{(i)}\nu^{(k)})$ and $(\nu^{(k)}\nu^{(j)})$. In this case, it might be necessary to re-assign the parent nodes of other hanging nodes lying on the original p-edge $(\nu^{(i)}\nu^{(j)})$.

We say that a T-mesh $\mathcal{T}_2$ is a *refinement of* $\mathcal{T}_1$, if $\mathcal{T}_2$ can be obtained from $\mathcal{T}_1$ through one or multiple consecutive edge insertions.

2.2. NURBS Geometry Mapping

We briefly recall the definition of B-spline basis functions and NURBS mappings. Let p be a non-negative integer and let $s = (s_1, \ldots, s_m)$ be a sequence of knots in the parameter space with $s_i \leq s_{i+1}$ for all i. The

$n = m - p - 1$ univariate *B-spline basis functions* $B^s_{i,p} : [s_1, s_m] \to \mathbb{R}$, $i = 1, \ldots, n$, are defined recursively as follows:

$$\begin{aligned} B^s_{i,0}(\xi) &= \begin{cases} 1 & \text{for} \quad s_i \le \xi < s_{i+1} \\ 0 & \text{else} \end{cases} \\ B^s_{i,p}(\xi) &= \frac{\xi - s_i}{s_{i+p} - s_i} B^s_{i,p-1}(\xi) \\ &\quad + \frac{s_{i+p+1} - \xi}{s_{i+p+1} - s_{i+1}} B^s_{i+1,p-1}(\xi). \end{aligned}$$

p is called the *degree* of the B-spline basis functions and s the *knot vector*.

Whenever a zero denominator appears in the definition above, the corresponding function $B^s_{i,p}$ is zero and the whole fraction is considered to be zero. We only consider *open knot vectors*, i.e. knot vectors in which the first and last knot appear $p + 1$-times.

Let $B^s_{i,p}$ and $B^t_{j,q}$ be two families of B-spline basis functions defined by the *degrees* p and q, and the open knot vectors $s = (s_1, \ldots, s_{n_s+p+1})$ and $t = (t_1, \ldots, t_{n_t+q+1})$, respectively. Let $w_{ij} \in \mathbb{R}_+$, $(i, j) \in \{1, \ldots, n_s\} \times \{1, \ldots, n_t\}$, be positive *weights*. We define the *bivariate NURBS basis functions* as follows:

$$R_{ij}(\xi_1, \xi_2) = \frac{R^N_{ij}(\xi_1, \xi_2)}{R^D(\xi_1, \xi_2)} \tag{1}$$

where the numerator R^N_{ij} and the denominator R^D are given by:

$$R^N_{ij}(\xi_1, \xi_2) = B^s_{i,p}(\xi_1)\, B^t_{j,q}(\xi_2)\, w_{ij}, \tag{2}$$

$$\begin{aligned} R^D(\xi_1, \xi_2) &= \sum_{k=1}^{n_s} \sum_{l=1}^{n_t} B^s_{k,p}(\xi_1)\, B^t_{l,q}(\xi_2)\, w_{kl} \\ &= \sum_{k=1}^{n_s} \sum_{l=1}^{n_t} R^N_{kl}(\xi_1, \xi_2). \end{aligned} \tag{3}$$

Given a *control net* of *control points* $P_{ij} \in \mathbb{R}^2$, $(i, j) \in \{1, \ldots, n_s\} \times \{1, \ldots, n_t\}$, the two-dimensional *NURBS-surface* $G : Q \to \Omega$ is defined by

$$G(\xi_1, \xi_2) = \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} R_{ij}(\xi_1, \xi_2)\, P_{ij}, \tag{4}$$

where $Q = [s_1, s_{n_s+p+1}] \times [t_1, t_{n_t+q+1}]$.

For a more detailed definition and introduction of NURBS mappings and B-spline basis functions, and for a discussion of the properties of these functions and mappings we refer to, e.g. [5, 13, 14, 21] and the references therein. Note that NURBS basis functions of degree p are, in general, globally C^{p-1}-continuous. If a knot is appears $k + 1$ times, the continuity is reduced by k. For example, at a double knot $s_i = s_{i+1}$, a NURBS basis function of degree 3 is C^1-continuous.

For our purposes, we assume that the geometry mapping is continuous and bijective (i.e. not self-penetrating), which are natural assumptions for CAD-applications.

2.3. Isogeometric Analysis and Isoparametric FEM

Let L be a second order elliptic operator on the Lipschitz domain $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$. We want to find a solution u of the problem

$$Lu = f \quad \text{on } \Omega \tag{5}$$

for a given f, with Neumann boundary conditions on Γ_N, and Dirichlet boundary conditions $u = g_D$ on Γ_D.

The *variational form* (or weak form) obtained from (5) is written as follows:

Find $u \in V_g$ such that
$a(u, v) = \langle F, v \rangle \quad$ for all $v \in V_0$

with $V_0 = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\} \subset H^1(\Omega)$ and $V_g = \{v \in H^1(\Omega) : v|_{\Gamma_D} = g_D\} \subset H^1(\Omega)$, where $H^1(\Omega)$ denotes the standard Sobolev space of order 1. Assuming that $a(\cdot, \cdot)$ is a bounded and coercive bilinear form and that F is a bounded linear form, the theorem of Lax-Milgram guarantees existence and uniqueness of a solution.

We use Galerkin's principle for discretization: We choose an appropriate finite-dimensional subspace $V_h \subset V$ and corresponding spaces V_{gh} and V_{0h}, and compute an approximate solution u_h as the solution of the finite-dimensional variational problem

Find $u_h \in V_{gh}$ such that
$a(u_h, v_h) = \langle F, v_h \rangle \quad$ for all $v_h \in V_{0h}$.

We refer to u_h as the *discrete solution*. Existence and uniqueness of the discrete solution are again guaranteed by the theorem of Lax-Milgram under the standard assumptions (see, e.g. [8, 12]).

Choosing a basis $\{\varphi^{(i)},\ i = 1, \ldots, n\}$ of V_h, we can express the discrete solution u_h as a linear combination of these basis functions with constant real coefficients $u^{(i)}$, i.e. in the following, general form:

$$u_h(x) = \sum_{i=1}^{n} u^{(i)}\ \varphi^{(i)}(x). \tag{6}$$

Starting point of *isogeometric analysis* proposed by Hughes et al. [21] is the NURBS geometry mapping as in (4). As discussed in [21], the NURBS basis functions R_{ij} have properties that make them suitable as finite element basis functions. Hence, their span is used as the finite-dimensional space V_h. The basis functions are globally defined over the parameter domain Q and transformed to the physical domain by the global geometry mapping G, which is illustrated in Fig. 4(a). The representation of the discrete solution (6) is re-formulated as follows:

$$u_h(x) = \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} u^{(ij)}\ (R_{ij} \circ G^{-1})(x).$$

The advantages of this approach regarding the exact geometry representation, as well as its weakness with regard to local refinement, even when using T-splines, have already been mentioned in Section 1.

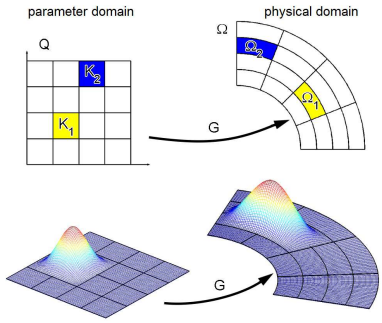
In the *classic FEM*, the domain Ω is partitioned into disjoint cells Ω_j. Each cell Ω_j is constructed as the image of the same reference element $\widehat{\Omega}$ under the local geometry mapping G_j, which is illustrated in Fig. 4(b). Here, the finite element basis functions $\varphi^{(i)}$ are represented as follows: On a single reference element $\widehat{\Omega}$, m basis functions $\widehat{p}_i$ are defined (e.g., for linear elements on a triagular element, $m = 3$). For each cell Ω_j, the local degrees of freedom have a local numbering, which we indicate with the index i. A local index is linked to the global index of the corresponding control variable, which we indicate by writing $k(i, j)$. The local representation of the discrete solution can thus be written as follows:

$$u_h|_{\Omega_j}(x) = \sum_{i=1}^{m} u^{(k(i,j))}\ \left(\widehat{p}_i \circ G_j^{-1}\right)(x).$$

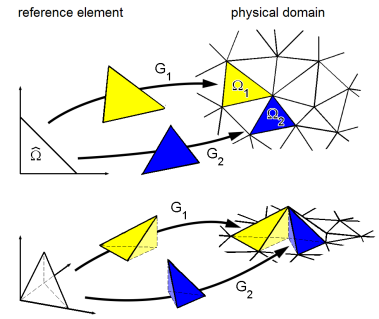
In *isoparametric FEM*, it is required that the geometry mappings G_j are in the space of shape functions, i.e. that they can be represented as linear combinations of the shape functions $\widehat{p}_k$.

While higher-order isoparametric finite elements may be capable of representing a piecewise curved approximation of boundaries, they still suffer the disadvantages mentioned in Section 1, such as inexact geometry representation, the need for data transformation, and the need for costly communication with the CAD-programme in the course of refinement.

In order to combine the advantages of these approaches, we suggest a hybrid method, which we discuss in the following section.



(a) Isogeometric approach:
One globally defined mapping.



(b) Classic FEM:
Element-wise defined mappings.

Figure 4: Illustration of mapping of geometry and basis functions in IgA and in classic FEM.

3. Hybrid Isogeometric FE Method

At first, we define the finite element space over a hierarchical T-mesh $\mathcal{T}$ of Q in Section 3.1. Its transformation to the physical domain in the isogeometric framework will then be discussed in Section 3.2.

3.1. FE Basis Functions over Q

Let the space of piecewise bicubic functions over $\mathcal{T}$ be denoted by

$$\mathcal{Q}_3(\mathcal{T}) = \Big\{ u : Q \to \mathbb{R} : \\ u|_K(\xi_1, \xi_2) = \sum_{i,j=0}^{3} c_K^{(ij)} \xi_1^i \xi_2^j, c_K^{(ij)} \in \mathbb{R}, \text{ for all } K \in \mathcal{T} \Big\}.$$

We choose the set of piecewise bicubic and globally C^1-continuous functions as finite element space X_h:

$$X_h = \mathcal{Q}_3(\mathcal{T}) \cap C^1(\overline{Q}). \tag{7}$$

We now discuss some properties and the construction of the used nodal basis for X_h.

3.1.1. C^1-Continuity on T-Meshes

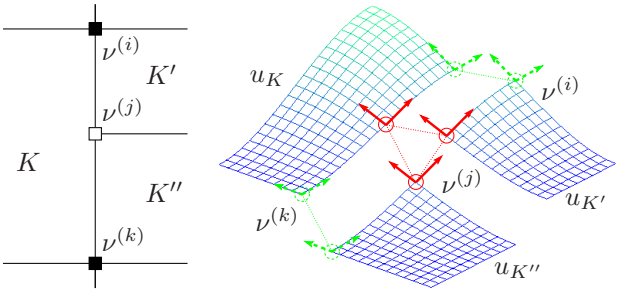
Let the restriction of a function $u \in \mathcal{Q}_3(\mathcal{T})$ to a cell $K \in \mathcal{T}$ be denoted by $u_K = u|_K$. We fix the 16 local degrees of freedom of u_K by prescribing the Hermite data u_K, $\partial_1 u_K$, $\partial_2 u_K$, and $\partial_{12} u_K$ at the four corners of K. Here, ∂_1 and ∂_2 denote the derivatives with respect to ξ_1 and ξ_2, respectively, and ∂_{12} denotes the

mixed derivative. For brevity, we will also write $\partial_0 u = u$ and $\partial_3 u = \partial_{12} u$. Over a rectangular mesh without hanging nodes, this finite element is known as the *Bogner-Fox-Schmit rectangle* [8, 12].

Let $\mathcal{I}_\alpha = \{0, 1, 2, 3\}$, and let $\mathcal{T}^{(i)}$ denote the set of all cells $K \in \mathcal{T}$ that contain $\nu^{(i)}$ in their closure. The following condition given in [34] is sufficient and necessary for global C^1-continuity of the extension of the Bogner-Fox-Schmit rectangle to T-meshes:

$$\left.\begin{aligned} &\partial_\alpha u_K(\nu^{(i)}) \;=\; \partial_\alpha u_{K'}(\nu^{(i)}), \\ &\text{for all } \alpha \in \mathcal{I}_\alpha, \;\text{ for all } K, K' \in \mathcal{T}^{(i)}, \\ &\text{for all nodes } \nu^{(i)}. \end{aligned}\right\} \tag{8}$$

For an illustration of (8) refer to Fig. 5: If we prescribe Hermite data of our choice at the primary nodes $\nu^{(i)}$ and $\nu^{(k)}$, condition (8) is automatically fulfilled by u_K and $u_{K'}$ at $\nu^{(i)}$ due to the way we define u. The same holds for u_K and $u_{K''}$ at $\nu^{(k)}$. This is indicated by the green, dashed circles and arrows in Fig. 5(b). In order to ensure C^1-continuity of u across the edges $(\nu^{(i)}\nu^{(j)})$ and $(\nu^{(j)}\nu^{(k)})$, we also have to ensure that (8) is fulfilled at $\nu^{(j)}$. Here, we can not choose the Hermite data freely. We have to prescribe the data determined by $u_K(\nu^{(j)})$, which is indicated by the red circles and arrows in Fig. 5(b).



(a) Part of mesh with hanging node $\nu^{(j)}$.

(b) Function values, first and mixed derivatives must be equal at nodes.

Figure 5: Illustration of condition (8) for global C^1-continuity.

3.1.2. Functions on a Hierarchical T-Mesh

We utilize the proof of the C^1-continuity in [12] to derive a procedure for determining the local representation of a function $u \in X_h$. We give a sketch of the procedure based on the situation illustrated in Fig. 5. The procedure for cases with more additional hanging nodes on the edge $(\nu^{(i)}\nu^{(k)})$ is the same; the procedure for p-edges which are parallel to the ξ_1-axis is analogous.

We note that the restriction of u_K to the edge $(\nu^{(i)}\nu^{(k)})$ is a univariate cubic polynomial in ξ_2. We denote this polynomial by v_{ik}:

$$v_{ik} = u_K|_{(\nu^{(i)}\nu^{(k)})}$$

The four degrees of freedom associated with $\partial_0 u_K$ and $\partial_2 u_K$ (i.e. function value and first derivative with respect to ξ_2) at the nodes $\nu^{(i)}$ and $\nu^{(k)}$ uniquely determine v_{ik}. In the same manner, the four degrees of freedom associated with $\partial_1 u_K$ and $\partial_3 u_K$ at $\nu^{(i)}$ and $\nu^{(k)}$ uniquely determine another univariate cubic polynomial which we denote by w_{ij}. This polynomial describes the derivative of u_K with respect to ξ_1 along the edge $(\nu^{(i)}\nu^{(k)})$, which here is also the outer normal derivative of u_K :

$$w_{ik} = (\partial_1 u_K)|_{(\nu^{(i)}\nu^{(k)})}$$

In order to derive the local representation $u_{K''}$, we need to determine the Hermite data at the four corners of K'', in particular at $\nu^{(j)}$ and $\nu^{(k)}$. As mentioned above, we can prescribe Hermite data of our choice at $\nu^{(k)}$, because it is a primary node and (8) is fulfilled automatically. To obtain the Hermiate data at $\nu^{(j)}$, we proceed as follows:

1. Identify the parent nodes $\nu^{(i)}$ and $\nu^{(k)}$ of $\nu^{(j)}$.
2. Determine v_{ik} and w_{ik} from the Hermite data at these parent nodes.
3. Evaluate v_{ik} and w_{ik} and prescribe the following Hermite data at $\nu^{(j)}$:
   $\partial_0 u_{K''}(\nu^{(j)}) = v_{ik}(\nu^{(j)})$
   $\partial_2 u_{K''}(\nu^{(j)}) = v'_{ik}(\nu^{(j)})$
   $\partial_1 u_{K''}(\nu^{(j)}) = w_{ik}(\nu^{(j)})$
   $\partial_3 u_{K''}(\nu^{(j)}) = w'_{ik}(\nu^{(j)})$

This choice guarantees that (8) is fulfilled at $\nu^{(j)}$. Whenever one of the parent nodes is a hanging node itself, the this procedure is applied recursively. This is where the rules (T1) and (T2) imposed on hierarchical T-meshes in Section 2.1 come into play, as they guarantee that the recursion terminates at a primary node, where we can choose the Hermite data.

By this, we have shown the following theorem:

Theorem 3.1. *Let $\mathcal{T}$ be a hierarchical T-mesh. Then any function in the finite element space X_h (see (7)) is uniquely determined by its Hermite data at the primary nodes of the T-mesh.*

For later reference, we define the *ancestors of a node* $\nu^{(i)}$ as the set of all primary nodes that are reached in the recursive procedure above. For example, in Fig. 1, the ancestors of $\nu^{(9)}$ are $\{\nu^{(6)}, \nu^{(11)}, \nu^{(14)}\}$, while the set of ancestors of $\nu^{(6)}$ is $\{\nu^{(6)}\}$. The set of *ancestors of a cell* K is defined as the union of the ancestors of the corner nodes of K. Hence, if all four corners of a cell K are primary nodes, these four primary nodes form the set of ancestors of K.

While steps 2. and 3. of the procedure above might seem costly, they are, in fact, combined into one calculation. The formula used to calculate the Hermite data at a hanging node from the data at its parent nodes is straightforward to derive by direct calculation.

From [12, 34] and the procedure given above, one can see that the Hermite data at a hanging node only depends on the data at its parent nodes (and the coordinates of these three nodes), but not on any other information. Hence, the only additional information that needs to be stored besides the coordinates of a node are the indices of its two parent nodes (see also Section 3.3).

3.1.3. Definition of Basis Functions

Since the Hermite data at a hanging node is already uniquely determined by the Hermite data at its parent nodes, the degrees of freedom at hanging nodes are restricted. Hence, the dimension of the space X_h corresponds to the number of degrees of freedom associated with primary nodes:

$$\dim X_h \quad = \quad 4N_X. \tag{9}$$

This corresponds to the dimension formula provided in [15]. Recall that $\mathcal{I}_X$ denotes the set of indices of primary nodes, and that $\mathcal{I}_\alpha = \{0, 1, 2, 3\}$. For a fixed $(i, \alpha) \in \mathcal{I}_X \times \mathcal{I}_\alpha$, the basis function $\widehat{\varphi}_\alpha^{(i)} : Q \to \mathbb{R}$ of X_h is uniquely determined by the $4N_X$ conditions:

$$\partial_\beta \widehat{\varphi}_\alpha^{(i)}(\nu^{(j)}) = \delta_{ij}\delta_{\alpha\beta}, \text{ for all } (j, \beta) \in \mathcal{I}_X \times \mathcal{I}_\alpha, \tag{10}$$

where δ_{ij} denotes Kronecker's delta. The set of functions $\widehat{\Phi} = \{\widehat{\varphi}_\alpha^{(i)} | (i, \alpha) \in \mathcal{I}_X \times \mathcal{I}_\alpha\}$ defined by (10) forms our basis of X_h. Linear independence of the functions in $\widehat{\Phi}$ follows directly from their definition and holds for any hierarchical T-mesh. Hence, linear independence is preserved in any refinement procedure that can be realized by consecutive edge insertions. This basis $\widehat{\Phi}$ does not form a partition of unity, which is not a problem in the finite element-context. If desired, the partition of unity property can be achieved easily by adapting the nodal values on the right-hand-side of the definition (10).

On a rectangular mesh without hanging nodes, the support of a basis function $\widehat{\varphi}_\alpha^{(i)}$ is contained in the four cells that have the node $\nu^{(i)}$ as a corner, i.e. to $\mathcal{T}^{(i)}$. When any of these cells are refined, the support of $\widehat{\varphi}_\alpha^{(i)}$ is reduced. For the process of cell-wise assembling of the stiffness matrix, one needs to know which basis functions have support on one specific cell K. The local representation a basis function $\widehat{\varphi}_\alpha^{(i)}$ is determined through the recursive procedure given in Section 3.1.2, and when a primary node is reached in this recursion, (10) is evaluated. Hence, only those basis function which are associated with the ancestors of a cell K have support on K. Identifying these basis functions is done by following the same recursion from hanging nodes to their parent nodes. Note that it is possible to construct hierarchichal T-meshes where the number of ancestors of a specific cell can increase with the number of levels. In practice, however, this number of ancestors is usually limited to 5 or, at the most, 6 primary nodes.

While Nguyen-Thanh et al. [29] use the same space of piecewise bicubic, globally C^1-continuous test functions, the nodal representation used herein does not need the computation of Bezier segments for refinement.

3.2. Hybrid Isogeometric-FE Basis Functions

Instead of transforming the basis functions $\widehat{\varphi}_\alpha^{(i)}$ from Q to Ω by the simple push-forward $\widehat{\varphi}_\alpha^{(i)} \circ G^{-1}$, we incorporate the denominator R_D (3) of the NURBS mapping in the transformation as follows: At first, we define the rational basis functions $\widetilde{\varphi}_\alpha^{(i)}$ over Q:

$$\begin{aligned} \widetilde{\varphi}_\alpha^{(i)} &= \frac{\widehat{\varphi}_\alpha^{(i)}}{R_D}, \\ \widetilde{X}_h &= \operatorname{span}\{\widetilde{\varphi}_\alpha^{(i)}\}. \end{aligned} \tag{11}$$

Then, we transform these rational functions to the physical domain Ω:

$$\begin{aligned} \varphi_\alpha^{(i)} &= \widetilde{\varphi}_\alpha^{(i)} \circ G^{-1} = \frac{\widehat{\varphi}_\alpha^{(i)} \circ G^{-1}}{R_D \circ G^{-1}}, \\ V_h &= \operatorname{span}\{\varphi_\alpha^{(i)}\}. \end{aligned} \tag{12}$$

The discrete solution $u_h \in V_h$ is now represented analogously to (6) in the following form:

$$u_h(x) = \sum_{i \in \mathcal{I}_X} \sum_{\alpha \in \mathcal{I}_\alpha} u_\alpha^{(i)} \left(\frac{\widehat{\varphi}_\alpha^{(i)} \circ G^{-1}}{R_D \circ G^{-1}} \right)(x).$$

We can define (11) and (12) for any NURBS mapping, but the most interesting case is the following:

(i) Let the initial mesh $\mathcal{T}_0$ be the grid defined by the knot vectors s and t of the geometry mapping G in (4) (or a refinement thereof),
(ii) let the degrees p and q be less or equal to 3, and
(iii) let the NURBS mapping be C^1-continuous.

Then, the bivariate B-spline basis functions $B_{i,p}^s(\xi_1) B_{j,q}^t(\xi_2)$ can be represented *exactly* as a linear combination of the basis functions $\widehat{\varphi}_\alpha^{(i)}(\xi_1, \xi_2)$. Hence, the numerator R_{ij}^N (2) of the NURBS mapping is in X_h. With (11), this means that

$$R_{ij} \in \widetilde{X}_h \qquad \text{for all } i, j.$$

To summarize, the use of these hybrid isogeometric-FE basis functions has the following consequences:

(a) The setting is *isoparametric*, because the NURBS basis functions can be represented exactly as linear combinations of the finite element basis functions $\widetilde{\varphi}_\alpha^{(i)}$. The element passes the standard patch test.
(b) The *geometry is represented exactly* on the coarsest mesh.
(c) The *exact geometry representation is preserved* in the refinement process.

(d) The *possibility for local refinement is gained* by the change of the basis: The finite element basis functions are defined over hierarchical T-meshes.
(e) By construction, the refinement of a cell effects only the cell itself and nodes on its boundary. In this sense, the refinement is *locally restricted*.
(f) *Linear independence* of the basis functions follows directly from their definition in Section 3.1, and linear independence is *preserved in the refinement process*.

Furthermore, from [6], the following approximation property of the finite element space immediately follows:

Theorem 3.2. *There exists a projector* $\Pi_h : V \to V_h$ *such that*

$$\|v - \Pi_h v\|_{H^1(\Omega)} \leq C\ h^3\ \|v\|_{H^4(\Omega)} \quad \text{for all } v \in H^4,$$

where h *denotes the mesh size, i.e. the diameter of the largest cell in the physical domain* Ω*, and where* C *is a constant which depends on the specific geometry mapping* G *but is independent of the mesh size.*

A corresponding approximation result is also available for functions v with $v|_{G(K)} \in H^4(G(K))$ for all cells $K \in \mathcal{T}$, see [6].

One can always choose the initial mesh $\mathcal{T}_0$ as a refinement of the grid determined by s and t, and in practical applications, the degrees of the NURBS geometry mappings usually are less or equal to 3. Hence, assumptions (i) and (ii) are easily fulfilled. The C^1-continuity of the geometry mapping (iii) is a more restrictive requirement. If the geometry mapping is only C^0-continuous, the setting is no longer isoparametric and we have to expect slower convergence rates due to the reduced smoothness of the transformed basis functions. However, note that, even in this case, statements (b)-(f) regarding the exact geometry representation, local refinement capabilities and linear independence of the basis functions remain valid.

Depending on the parametrization of a NURBS surface, an equally-spaced mesh of the parameter domain may be mapped to a distorted mesh in the physical domain. Hughes et al. [21] describe an algorithm that provides a more balanced mesh of the physical domain. This algorithm can also be applied to the T-meshes used herein.

If all weights w_{ij} of the geometry mapping G are equal, then the NURBS geometry mapping is reduced to a B-spline surface S of the form

$$S(\xi_1, \xi_2) = \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} B^s_{i,p}(\xi_1)\ B^t_{j,q}(\xi_2).$$

In this case, it is sufficient to transform the polynomial basis functions without incorporating the denominator R_D, i.e. $\varphi^{(i)}_\alpha = \widehat{\varphi}^{(i)}_\alpha \circ G^{-1}$. The isoparametric setting is then preserved under the same assumptions (i)-(iii) mentioned above.

3.3. Implementational Issues

As discussed in Section 2.2, the NURBS geometry mapping has an underlying grid-structure. This does not only motivate the use of rectangular finite elements, but also automatically provides an initial, coarse mesh.

To fully describe the hierarchical T-mesh, we store the following information:

- For each cell:
  - The indices of the four nodes at the cell's corners.
- For each node:
  - Its coordinates.
  - A flag indicating if the node is primary or hanging.
  - The indices of the two parent nodes, if any.

As we have elaborated in Section 3.1.2, identifying the parent nodes of a hanging node is necessary for obtaining the local representation of a function in X_h. Obviously, the primary/hanging-flag listed above can be omitted, if, for example, a primary node is marked by setting its parent node indices to zero. Hence, when compared to classical FEM mesh structures, the only additional information that needs to be stored are the two indices of the parent nodes.

As mentioned in Section 2.1, changing a hanging node $\nu^{(i)}$ into a primary node might split one p-edge into two p-edges. Since this might require that the parent nodes of some hanging nodes are re-assigned, one needs to find those hanging nodes which have $\nu^{(i)}$ as a parent. This information can be retrieved directly from the stored nodal data, if a fast method is available. If not, it would be convenient to store the following additional information:

- For each primary node $\nu^{(i)}$:
  - The indices of the hanging nodes that have $\nu^{(i)}$ as parent node.

Re-assigning parent nodes in the course of splitting a p-edge is simple: Due to the axis-aligned structure, we only have to check one coordinate in order to determine which one of the new p-edges contains a certain hanging node.

Note that, even though, in priciple, each node has an assigned level, it is neither necessary to store nor to update these levels. Rule (T2) states that the levels of the parent nodes have to be smaller, but the actual number of the level is irrelevant. As long as the re-assignment of parent nodes is done properly in the refinement process, the hierarchical structure is preserved.

With this general approach to cell-refinement, *any* refinement procedure can be applied as long as it can be realized by consecutive edge insertions.

4. Adaptive Refinement

In the previous section, we have discussed the local refinement concept. We now discuss the ingredients needed for the implemented adaptive refinement procedure: The cell refinement procedure, the a posteriori error estimator, and the definition of refinement criteria.

4.1. Cell Refinement Procedure

In the refinement process illustrated in Fig. 2, several edge insertions result in the creation of only one new interior primary node. However, as discussed in Section 3.1, the number of degrees of freedom is directly related to the number of primary nodes by (9). Naturally, one desires a refinement procedure that guarantees the creation of new degrees of freedom, i.e. of at least one new primary node, when a cell is refined. The simplest such procedure is *cross insertion*, which was applied to the numerical test described in Section 5: We split a cell into four smaller, axis-aligned cells, such that one new primary node is created in the interior of the original cell. In addition, we align the new nodes in such a way that, when two neighbouring cells are refined, a primary node is created on their common edge. Note that a cross insertion can be interpreted as three consecutive edge insertions, hence all statements from above remain valid.

4.2. A Posteriori Error-Estimator

We apply the a posteriori error estimator discussed by Dörfel et al. [17] to our situation. It is based on an concept presented by Bank and Smith [4]: We enlarge the function space V_h by additional basis functions which are in V. Let B_h denote the space that is spanned by these additional functions, and let $\bar{V}_h = V_h \oplus B_h \subset V$. Let $\bar{u}_h$ denote the solution of the partial differential equation in $\bar{V}_h$. Assuming that $\bar{V}_h$ is a better approximation of V than V_h, we can expect $\bar{u}_h$ to be a better solution than u_h. Hence, we may use $e_h = \bar{u}_h - u_h \in B_h$ as an indicator to identify areas with large local error.

One could obtain e_h by setting up and solving the variational formulation for $\bar{u}_h$ in $\bar{V}_h$, which would be very costly. Instead, as discussed in [17], we determine e_h from known data without having to calculate $\bar{u}_h$ explicitly, which is done with a special choice of B_h. We define the univariate *bubble function* as follows:

$$\widehat{\beta}_{a,b}(\xi) \;=\; \begin{cases} \dfrac{\xi-a}{b-a}\,\dfrac{b-\xi}{b-a}, & \xi \in (a,b) \\ 0 \qquad\qquad\quad , & \text{else} \end{cases}$$

For a cell $K \in \mathcal{T}$, $K = (a_1, b_1) \times (a_2, b_2)$, we define the bivariate bubble function $\widehat{\beta}^{(K)}$:

$$\widehat{\beta}^{(K)}(\xi_1, \xi_2) \;=\; \widehat{\beta}_{a_1,b_1}(\xi_1)\,\widehat{\beta}_{a_2,b_2}(\xi_2).$$

Note that the support of $\widehat{\beta}^{(K)}$ is the cell K. We define their transformations to the physical domain and B_h as their span:

$$\begin{aligned} \beta^{(K)}(x) &= \widehat{\beta}^{(K)} \circ G^{-1}(x) \\ B_h &= \operatorname{span}\{\beta^{(K)}, K \in \mathcal{T}\} \end{aligned}$$

We can now re-formulate the variational formulation for $\bar{u}_h$ in $\bar{V}_h$ to obtain the following variational form for e_h in B_h:

> Find $e_h \in B_h$ such that
> $a(e_h, v) = \langle F, v\rangle - a(u_h, v) \quad$ for all $v \in B_h$.

Expressing $e_h = \sum_{K\in\mathcal{T}} e^{(K)}\beta^{(K)}$, inserting all bubble functions $\beta^{(K)}$ as test functions, and noting that $a(\beta^{(K)}, \beta^{K'}) = 0$ for $K \neq K'$, we obtain the following formula for the coefficients $e^{(K)}$ of e_h:

$$e^{(K)} = \frac{\left\langle F, \beta^{(K)}\right\rangle - a(u_h, \beta^{(K)})}{\|\beta^{(K)}\|_E^2},$$

where the *energy norm* $\|\cdot\|_E$ is defined by $\|u\|_E = \sqrt{a(u,u)}$ for bounded and coercive bilinear forms $a(\cdot,\cdot)$.

We obtain the following a posteriori error estimator η:

$$\begin{aligned} \eta^2 &= \|e_h\|_E^2 \\ &= \sum_{K\in\mathcal{T}} \|e^{(K)}\beta^{(K)}\|_E^2 \;=\; \sum_{K\in\mathcal{T}} \eta_K^2, \end{aligned} \tag{13}$$

where

$$\eta_K \;=\; \frac{\left\langle F, \beta^{(K)}\right\rangle - a(u_h, \beta^{(K)})}{\|\beta^{(K)}\|_E}. \tag{14}$$

In the case of two-dimensional, vector-valued solutions, we define the vector-valued bubble-functions

$$\beta_1^{(K)} = (\beta^{(K)}, 0)^T, \qquad \beta_2^{(K)} = (0, \beta^{(K)})^T,$$

and we express $e_h = \sum_{K\in\mathcal{T}} \sum_{d=1,2} e_d^{(K)}\beta_d^{(K)}$. Analogously to above, we obtain the following formula for the coefficients $e_d^{(K)}$:

$$\begin{pmatrix} e_1^{(K)} \\ e_2^{(K)} \end{pmatrix} \;=\; \left(A_{\beta\beta}^{(K)}\right)^{-T} \begin{pmatrix} R_1^{(K)} \\ R_2^{(K)} \end{pmatrix}$$

where

$$\begin{aligned} \left(A_{\beta\beta}^{(K)}\right)_{ij} &= a(\beta_i^{(K)}, \beta_j^{(K)}), \quad i,j = 1,2, \\ R_d^{(K)} &= \left\langle F, \beta_d^{(K)}\right\rangle - a(u_h, \beta_d^{(K)}), \quad d = 1,2. \end{aligned}$$

The 2×2-matrix $A_{\beta\beta}^{(K)}$ is inverted easily. The error estimator is then again defined as

$$\begin{aligned} \eta^2 &= \|e_h\|_E^2 \\ &= \sum_{K \in \mathcal{T}} \underbrace{\sum_{i,j=1,2} e_i^{(K)} e_j^{(K)} a(\beta_i^{(K)}, \beta_j^{(K)})}_{=\eta_K^2} . \end{aligned} \tag{15}$$

For an error estimator to be useful, it must be both *reliable* and *efficient* (see [11] for detailed definitions). The efficiency of this error estimator defined in (14) and (15) follows directly from the boundedness of the bilinear form $a(\cdot, \cdot)$, while its reliability depends on the specific problem. For a detailed discussion of the reliability and efficiency of this error estimator, refer to [4, 17].

Once we have calculated the local errors η_K for each cell $K \in \mathcal{T}$, we can mark those cells for refinement, for which the criterion

$$\eta_K > \Theta \tag{16}$$

is fulfilled with a threshold Θ. In the examples in Section 5, two methods for determining Θ are applied: Choosing a parameter $\alpha \in (0, 1)$, we calculate $\Theta = \mathsf{Val}[\alpha]$ or $\Theta = \mathsf{Qua}[\alpha]$, where

$$\mathsf{Val}[\alpha] = \alpha \cdot \max_{K \in \mathcal{T}} \{\eta_K\}, \tag{17}$$

$$\mathsf{Qua}[\alpha] = \alpha - \text{quantile}_{K \in \mathcal{T}} \{\eta_K\}, \tag{18}$$

In the first case (17), we mark all cells K for refinement for which η_K is larger than α-times the largest appearing local error. In the second case (18), Θ is chosen such that the $(1 - \alpha) \cdot 100$ percent of all cells with the largest local errors are refined (e.g. if $\alpha = 0.85$, then 15% of the cells are refined). In both cases, the smaller the value of α, the more cells will be refined; $\alpha = 0$ corresponds to global refinement.

When $\Theta = \mathsf{Val}[\alpha]$ in (16), the number of marked cells varies, depending on the distribution of the error. This way, on the one hand, it is possible to achieve a concentrated refinement in areas with large local errors, even if these areas are very small. On the other hand, if the local error in a certain area is much larger than in the rest of the domain, and if α is chosen too large, this criterion might fail to identify other areas that need refinement. When $\Theta = \mathsf{Qua}[\alpha]$ in (16), a specific percentage of all cells is marked for refinement and the resulting refinement is more balanced. However, if α is chosen too small, an "over-refinement" might occur in areas in which the discrete solution is already accurate.

The adaptive refinement procedure can be summarized as follows:

1. Create the initial mesh $\mathcal{T}_0$ from knot vectors s and t of the geometry mapping (or let it be a refinement thereof).
2. Define the basis functions on the T-mesh $\mathcal{T}_k$ over Q, incorporate the denominator R_D in the transformation to Ω.
3. Solve the discretized system and obtain the discrete solution u_h^k.
4. If the desired accuracy is not reached, refine adaptively:
   (a) Calculate the local error estimator η_K for each cell K, determine threshold Θ for the chosen parameter α.
   (b) Refine all cells for which $\eta_K > \Theta$ by cross insertion, thereby obtaining the refined T-mesh $\mathcal{T}_{k+1}$. Return to step 2.

5. Numerical Examples

We apply the described method to advection-dominated advection-diffusion problems, to the Laplace equation and to linear elasticity problems.

For better readability, we will express Cartesian coordinates as (x, y) instead of (x_1, x_2) throughout this section.

5.1. Advection-Dominated Advection-Diffusion Problems

We solve the advection-diffusion equation

$$-\kappa\Delta u + b^T \nabla u = 0 \quad \text{in } \Omega$$

with

$$\begin{aligned} \kappa &= 10^{-6}, \\ b &= (\cos\phi, \sin\phi)^T, \end{aligned}$$

and Dirichlet boundary conditions, i.e. $\partial\Omega = \Gamma_D$.

We consider two cases: In Case 1, we set $\phi = 45°$ and $\Omega = (0,1)^2$, and we apply the following boundary conditions (see also Fig. 6(a)):

$$g_D(x,y) = \begin{cases} 0, & \text{if } (x=1) \vee (y=1) \\ & \vee (x=0 \wedge y > 0.2), \\ 1, & \text{else.} \end{cases}$$

In Case 2, we set $\phi = 15°$, and Ω is the curved domain illustrated in Fig. 7(a). In polar coordinates, this domain is given by $\Omega = \{(r,\theta) \in (1,2)\times(0,\pi/2)\}$; the complete geometry data is given in Appendix Appendix A.1. The boundary conditions in this case are as follows:

$$g_D(x,y) = \begin{cases} 1, & \text{if } (x < 0.8) \wedge (y < 1.2), \\ 0, & \text{else.} \end{cases}$$

The Peclet number Pe is defined by $Pe = L|b|/\kappa$, where L is the side length of the domain. If $Pe \gg 1$, the advection dominates the diffusion, which is clearly the case in our examples with $Pe \approx 10^6$. This requires the use of a stabilization method, and as in [17, 21], the Streamline Upwinding Petrov Galerkin (SUPG) stabilization method is used (see, e.g., [30]). The stabilization parameter τ is set to

$$\tau(K) = \frac{h_b(K)}{2\,|b|},$$

where $h_b(K)$ is the length of the cell K in direction of the flow b.

The strong advection and the discontinuous boundary conditions result in sharp layers. The numerical results in Fig. 6 and Fig. 7 show that the a posteriori error estimator detects the interior and boundary layers in both cases. The refined areas correspond to the expected positions of the layers, which are indicated by thin blue lines in Fig. 6(a) and Fig. 7(a).

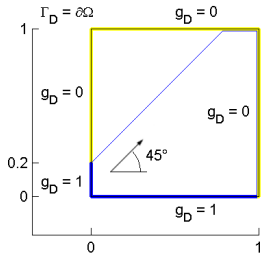
Note that Case 1 is used by Dörfel et al. [17] to illustrate how the refinement with T-splines requires an unwanted insertion of additional knots in the process of mesh refinement. This triggers a snowball-effect where more additional knots are inserted consecutively, leading to almost global refinement. Clearly, the refinement by cross-insertion in hierarchical T-meshes is locally restricted and additional nodes are created only in the area of interest.

5.2. Laplace Equation on L-Shaped Domain

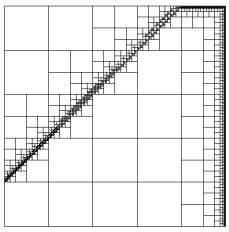
This example was also considered by Dörfel et al. [17]: We solve the Laplace equation

$$\Delta u = 0 \tag{19}$$

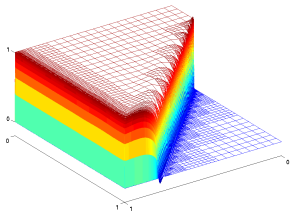
on the L-shaped domain $\Omega = (-1,1)^2 \setminus [0,1]^2$ that is illustrated in Fig. 8(a). We use double control points to model the corners at $(0,0)$ and $(-1,-1)$; the complete geometry data is given in Appendix Appendix A.2.



(a) Problem setting.



(b) T-mesh.



(c) Numerical solution.

Figure 6: Problem setting, T-mesh and solution for Case 1 after 6 refinements with $\Theta = \mathsf{Val}[0.2]$.

The following function u solves (19) and is used as our exact solution:

$$\begin{aligned} u: \quad \mathbb{R}^+ \times (0, 2\pi] &\rightarrow \mathbb{R} \\ (r, \theta) &\mapsto r^{\frac{2}{3}} \sin\left(\frac{2\theta - \pi}{3}\right) \end{aligned}$$

We define the following boundaries (see also Fig. 8(a)):

$$\begin{aligned} \Gamma_D &= \{0\} \times [0,1] \cup [0,1] \times \{0\} \\ \Gamma_N &= \partial\Omega \backslash \Gamma_D \end{aligned}$$

The corresponding boundary conditions are $u = g_D$ on Γ_D and $\nabla u \cdot \vec{n} = g_N$ on Γ_N, where $\vec{n}$ denotes the outer unit normal vector. g_D and g_N are determined by the exact solution u.

The exact solution has a singularity at the corner $(0, 0)$, and we test the ability of the error estimator to detect this. We calculate the refinement threshold as $\Theta = \mathsf{Val}[\alpha]$. Since the local errors near the singularity are very large in comparison with other areas, we use the small parameter $\alpha = 0.1$ in order to achieve a proper refinement.

The T-mesh after 5 adaptive refinement steps with $\Theta = \mathsf{Val}[0.1]$ is depicted in Fig. 8(b) and illustrates that the singularity is detected correctly. The error in the energy norm $\|u - u_h\|_E$ is plotted in Fig. 8(c) versus the degrees of freedom (DoF) and shows the faster convergence of the adaptive method (solid line) in comparison to uniform refinement (dashed line).

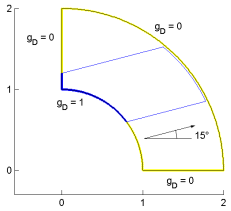
When comparing the results for uniform refinement to those of Dörfel et al. [17], we see that we do not only reach the same convergence rate, but surprisingly, a better accuracy as well.

5.3. Linear Elasticity Problems

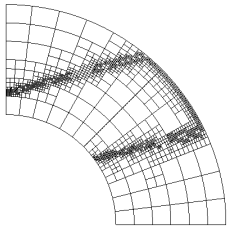
We study the linear elastic behaviour of the displacement field $u : \Omega \rightarrow \mathbb{R}^2$ described by

$$-\mathrm{div}\, \sigma(u) = 0 \quad \text{in } \Omega$$

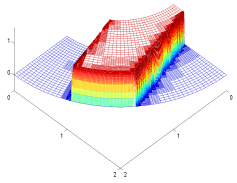
with the boundary conditions $u = u_D$ on Γ_D and $\sigma(u) \cdot \vec{n} = t_N$ on Γ_N.



(a) Problem setting.



(b) T-mesh.



(c) Numerical solution.

Figure 7: Problem setting, T-mesh and solution for Case 2 after 4 refinements with $\Theta = \mathsf{Qua}[0.6]$.

The Lamé's parameters λ and μ are given by

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)},$$

where E denotes the Young's modulus and ν the Poisson's ratio.

5.3.1. Bending of a Cantilever Loaded at the End

We consider a cantilever of length L and thickness D. It is fixed at $x = 0$ and subject to a parabolic traction at $x = L$ with resultant P as illustrated in Fig. 9(a). We choose the parameters as follows: $L = 48$m, $D = 12$m, $E = 3 \cdot 10^7$kPa, $\nu = 0.3$, and $P = 1000$N.

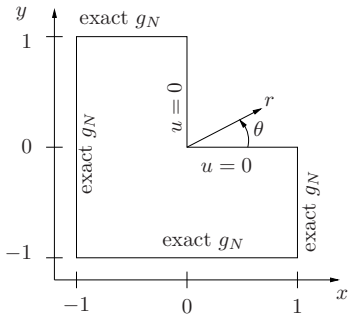
An analytical solution for the displacement field $u = (u_1, u_2)^T$ can be found in [29, 35]:

$$\begin{aligned} u_1 &= \frac{Py}{6E_0 I}\left((6L-3x)x + (2+\nu_0)\left(y^2 - \frac{D^2}{4}\right)\right), \\ u_2 &= -\frac{P}{6E_0 I}\left(3\nu_0 y^2(L-x) + (4+5\nu_0)x\frac{D^2}{4} \right. \\ &\quad \left. +(3L-x)x^2\right), \end{aligned}$$

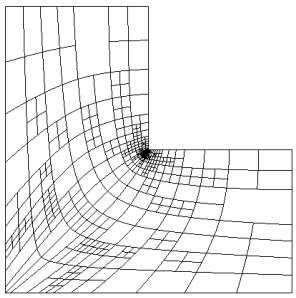
where $I = D^3/12$ is the moment of inertia of the cross section of the cantilever.

When we consider the plane stress problem, we set $E_0 = E$ and $\nu_0 = \nu$. For the plane strain problem, we set $E_0 = E/(1-\nu^2)$ and $\nu_0 = \nu/(1-\nu)$. Then, in both cases, the resulting exact stress components are as follows:

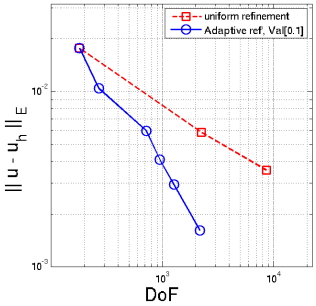
$$\begin{aligned} \sigma_{11} &= \frac{P(L-x)y}{I}, \\ \sigma_{12} &= -\frac{P}{2I}\left(\frac{D^2}{4} - y^2\right), \\ \sigma_{22} &= 0. \end{aligned}$$



(a) Problem setting.



(b) T-mesh after 5 refinements.



(c) Errors $\|u - u_h\|_E$.

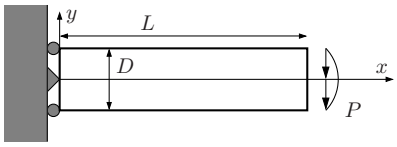
Figure 8: Laplace equation on L-shaped domain. Refinement with $\Theta = \mathsf{Val}[0.1]$.

We apply exact Dirichlet boundary conditions at the boundary $x = 0$, and the exact traction t_N as Neumann boundary condition at $x = L$. The remaining boundaries at $y = \pm D/2$ are free of traction.

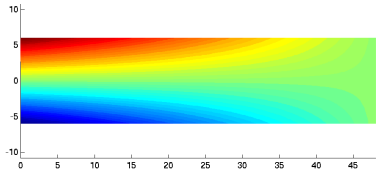
Due to the simple geometry mapping, the denominator R^D of the NURBS mapping (1) cancels out, and since the exact solution is a bicubic polynomial, the exact solution u is in V_h. By using a 4-point Gaussian quadrature rule, we can exactly calculate the integrals necessary for assembling the stiffness matrix and the right-hand-side of the discrete variational form. Hence, both in the plane stress and the plane strain case, our numerical solution is exact up to machine precision. The stress components σ_{11} and σ_{12} of the calculated solution are illustrated in Fig. 9(b) and Fig. 9(c), respectively. The same problem was considered in [29], where a mesh-dependent discretization error was observed. This may be due to numerical errors introduced by the computation of Bezier segments, which is not necessary in our approach.

5.3.2. Bending of a Curved Bar by a Force at the End

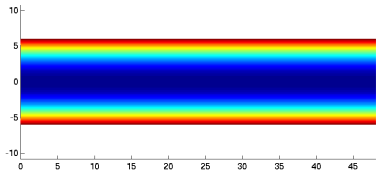
We now consider the following plane stress problem: A curved bar is fixed on one end and bent by a traction with resultant P acting at the other end, as illustrated in Fig. 10(a). This example can also be found in [2]. Note that, like in the previous example, the traction is not distributed uniformly, but determined by the exact solution presented below. We set the inner radius of the bar $a = 13$m, and the outer radius $b = 17$m. The other parameters are chosen as $E = 10^5$Pa, $\nu = 0.3$, and $P = 1$N. The exact solution is again



(a) Problem setting.



(b) Stress component σ_{11}.



(c) Stress component σ_{12}.

Figure 9: Bending of a cantilever loaded at the end.

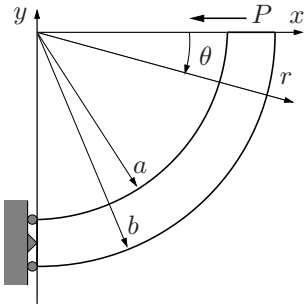
presented in [35]:

$$
\begin{aligned}
u_r &= \frac{\sin\theta}{E}\left(D(1-\nu)\log r + A(1-3\nu)r^2 + \frac{B(1+\nu)}{r^2}\right) \\
&\quad + \frac{1}{E}\left(-2D\theta\cos\theta + K\sin\theta + L\cos\theta\right), \\
u_\theta &= -\frac{\cos\theta}{E}\Big(-D(1-\nu)\log r + A(5+\nu)r^2 \\
&\quad + \frac{B(1+\nu)}{r^2}\Big) + \frac{1}{E}(2D\theta\sin\theta + D(1+\nu)\cos\theta \\
&\quad + K\cos\theta - L\sin\theta),
\end{aligned}
$$

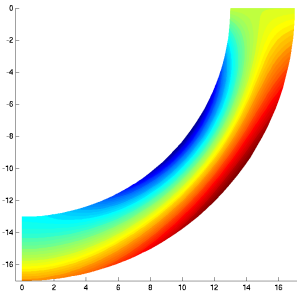
where the constants are set as follows:

$$
\begin{aligned}
N &= a^2 - b^2 + (a^2+b^2)\log(b/a) \\
A &= P/(2N) \\
B &= -Pa^2b^2/(2N) \\
D &= -P(a^2+b^2)/N \\
L &= D\pi \\
r_0 &= (a+b)/2 \\
K &= -D(1-\nu)\log r_0 - A(1-3\nu)r_0^2 \\
&\quad - B(1+\nu)/r_0^2
\end{aligned}
$$

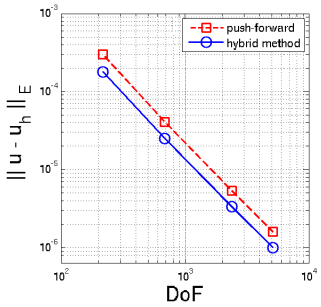
We apply exact Dirichlet boundary conditions at the boundary $x = 0$ (i.e. $\theta = \pi/2$), and the exact traction t_N as Neumann boundary condition at $y = 0$ (i.e. $\theta = 0$). The curved boundaries (i.e. $r = a$ and $r = b$) are free of traction. The calculated stress component σ_{22} is depicted in Fig. 10(b).



(a) Problem setting.



(b) Stress component σ_{22}.



(c) Errors $\|u - u_h\|_E$, uniform refinement.

Figure 10: Bending of a curved bar by a force at the end.

The circular boundary of the domain is represented exactly by the NURBS mapping, and different to the previous example 5.3.1, we now have a mapping with unequal weights and a denominator R^D that does not cancel out. For the geometry data, see Appendix Appendix A.1.

In this example, we investigate the effect of the rational character of the geometry mapping: We solve the problem by transforming the basis functions to the physical domain as described in Section 3.2. This method will be referred to as hybrid method. We also solve the problem by using basis functions $\bar{\varphi}_\alpha^{(i)}$ that result from the simple push-forward to the physical domain $\bar{\varphi}_\alpha^{(i)} = \widehat{\varphi}_\alpha^{(i)} \circ G^{-1}$. This will be referred to as push-forward method. The corresponding errors are plotted in Fig. 10(c) and show that the hybrid method (solid line), in which the isoparametric setting is preserved, leads to more accurate results than just using the push-forward method (dashed line).

5.3.3. Infinite Plate with Circular Hole

The following plane strain problem has also been discussed in, e.g., [17, 21, 27, 29]: We consider an infinite plate with a circular hole with radius $r = 1$m in its center. We choose $E = 10^5$Pa and $\nu = 0.3$. The plate is subject to an in-plane force of magnitude $T_1 = 10$N in x-direction, as illustrated in Fig. 11. The exact solution for the displacement field $u = (u_1, u_2)^T$ can be found, for example, in [27, 28]:

$$u_1(x,y) = \frac{xT_1A_1}{4\mu(\lambda+\mu)(x^2+y^2)^3}$$
$$u_2(x,y) = \frac{yT_1A_2}{4\mu(\lambda+\mu)(x^2+y^2)^3}$$

where

$$
\begin{aligned}
A_1 &= -(\lambda+\mu)x^2 + (3\lambda+5\mu)x^4 + (\lambda+2\mu)x^6 \\
&\quad +(3\lambda+3\mu)y^2 + (2\lambda+6\mu)x^2y^2 + (3\lambda+6\mu)x^4y^2 \\
&\quad -(\lambda-\mu)y^4 + (3\lambda+6\mu)x^2y^4 + (\lambda+2\mu)y^6, \\
A_2 &= -(3\lambda+3\mu)x^2 + (3\lambda+\mu)x^4 - \lambda x^6 \\
&\quad +(\lambda+\mu)y^2 + (2\lambda-2\mu)x^2y^2 - 3\lambda x^4y^2 \\
&\quad -(\lambda+3\mu)y^4 - 3\lambda x^2y^4 - \lambda y^6.
\end{aligned}
$$

Due to the symmetry of the problem, we solve the problem on a quarter of the domain. As in [27], we do

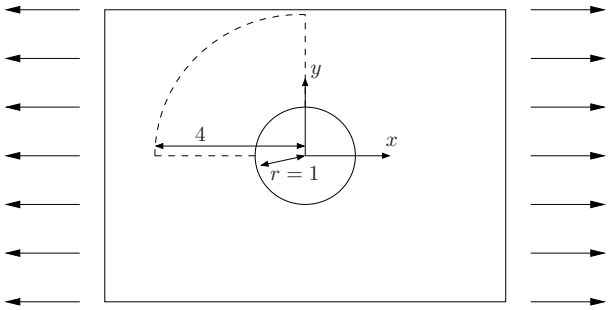
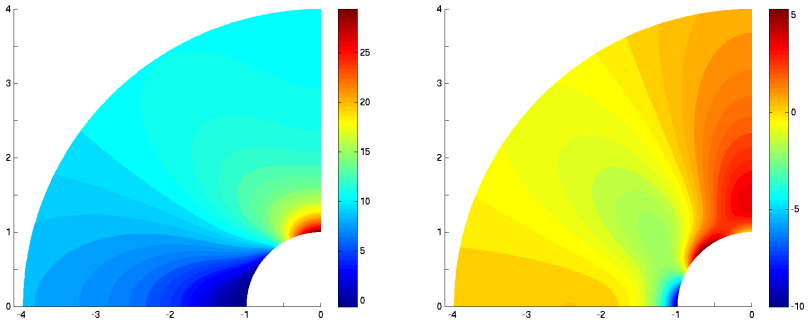


Figure 11: Infinite plate with circular hole: Problem setting.

this on the circular domain indicated by the dashed lines in Fig. 11 and depicted in Fig. 13(a). For the geometry data see Appendix Appendix A.1.



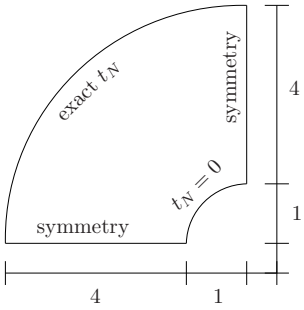
(a) Stress component σ_{11}. (b) Stress component σ_{22}.

Figure 12: Stress components of u_h.

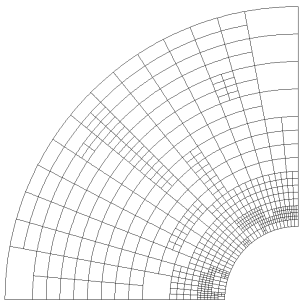
We apply symmetry conditions at the boundaries $x = 0$ and $y = 0$, and traction free Neumann conditions at the circular hole. At the remaining boundary, the exact traction is applied as a Neumann boundary condition (see Fig. 13(a)).

In this example, large stresses appear in the area near the circular hole, which can be seen in Fig. 12, where the stress components σ_{11} and σ_{22} are plotted. The maximum of the stress component σ_{11} is $3T_1$ and appears at the top end of the circular hole at $(0, 1)$. The a posteriori error estimator identifies and marks the area near the circular hole for refinement. The adaptively refined T-mesh after 4 refinements with $\Theta = \mathsf{Qua}[0.5]$ is depicted in Fig. 13(b). Note that, even though the stresses are greatest near the circular hole, the exact solution of the displacement field in the domain is smooth. Hence, the adaptive

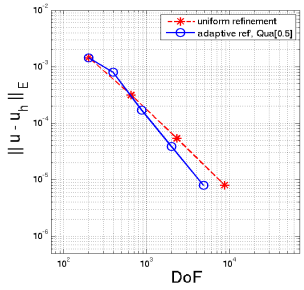
refinement is not only concentrated near the hole, but gradually results in an over-all refinement of the whole domain. The error $\|u - u_h\|_E$ in Fig. 13(c) also shows that, while the convergence rate achieved by the adaptive refinement procedure (solid line) is indeed higher than with uniform refinement (dashed line), the quantitative differences are not very dramatic due to the absence of a peak or singularity in the exact solution. This result corresponds to the findings in [17, 29].



(a) Boundary conditions.



(b) T-mesh after 4 refinements.



(c) Errors $\|u - u_h\|_E$.

Figure 13: Infinite plate with circular hole. Refinement with $\Theta = \mathsf{Qua}[0.5]$.

6. Conclusion

We have extended the piecewise bicubic, globally C^1-continuous Bogner-Fox-Schmit rectangle to hierarchical T-meshes in the isogeometric framework. The findings and advantages gained thereby are manifold:

Coupling bicubic elements with NURBS geometry mappings provides a simple approach to isogeometric analysis. The chosen representation over hierarchical T-meshes allows a simple implementation for local refinement. By embedding the well-known finite element in the isogeometric framework, the advantage of exact geometry representation on the first, coarsest mesh is inherited. The exact geometry representation is preserved throughout the refinement process without the need for additional communication with the CAD programme.

We can combine this approach with the discussed a posteriori error estimator, which results in an adaptive refinement procedure. The presented numerical tests illustrate that this procedure is capable of identifying and adaptively refining areas with large local errors. The results show the desired, faster convergence.

The incorporation of the NURBS-denominator in the transformation of the basis functions is essential to preserve the isoparametric setting. The numerical test carried out on a domain with an exactly represented, circular boundary shows that a naive push-forward method leads to a less accurate solution.

Since the used basis functions are only C^1-continuous, we have to expect a higher error compared to smoother B-spline basis functions. However, this was not confirmed in the numerical test on the L-shaped

domain.

Appendix A. Geometry Data

Appendix A.1. Curved Domains

We give the geometry data for the curved domain in Case 2 of the advection-dominated advection-diffusion problem, Section 5.1. The curved geometries in Section 5.3.2 (bending of a curved bar by a force at the end) and Section 5.3.3 (plate with circular hole) are variations of this mapping.

Degrees: $p = q = 2$
Knot vectors: $s = (0, 0, 0, 1, 1, 1)$
$t = (0, 0, 0, 0.5, 1, 1, 1)$

Coordinates of control points and weights ($i = 1, 2, 3$):

j	P_{ij}	w_{ij}
1	$k_i(1, 0)$	1
2	$k_i(1, \sqrt{2} - 1)$	$(1 + \frac{1}{\sqrt{2}})/2$
3	$k_i(\sqrt{2} - 1, 1)$	$(1 + \frac{1}{\sqrt{2}})/2$
4	$k_i(0, 1)$	1

with $k_1 = 1$, $k_2 = 1.5$, and $k_3 = 2$.

Appendix A.2. L-Shaped Domain

Geometry data for the L-shaped domain in Section 5.2.

Degrees: $p = q = 2$
Knot vectors: $s = (0, 0, 0, 0.25, 0.5, 0.75, 1, 1, 1)$
$t = (0, 0, 0, 0.5, 1, 1, 1)$
Weights: $w_{ij} = 1$ for all i, j

Coordinates of control points:

i	P_{i1}	P_{i2}	P_{i3}	P_{i4}
1	$(-1, 1)$	$(-0.75, 1)$	$(-0.3, 1)$	$(0, 1)$
2	$(-1, 0.2)$	$(-0.7, 0.35)$	$(-0.3, 0.5)$	$(0, 0.5)$
3	$(-1, -1)$	$(-0.6, -0.3)$	$(-0.2, -0.05)$	$(0, 0)$
4	$(-1, -1)$	$(-0.3, -0.6)$	$(-0.05, -0.2)$	$(0, 0)$
5	$(0.2, -1)$	$(0.35, -0.7)$	$(0.5, -0.3)$	$(0.5, 0)$
6	$(1, -1)$	$(1, -0.75)$	$(1, -0.3)$	$(1, 0)$

[1] I. Akkerman, Y. Bazilevs, V.M. Calo, T.J.R. Hughes, S. Hulshoff, The role of continuity in residual-based variational multiscale modeling of turbulence, Comput. Mech. (2008) 41: 371-378.

[2] S.N. Atluri, Z.D. Han and A.M. Rajendran, A New Implementation of the Meshless Finite Volume Method, Through the MLPG "Mixed" Approach, CMES, vol.6, no.6 (2004): 491-513.

[3] F. Auricchio, L. Beirão da Veiga, A. Buffa, C. Lovadina, A. Reali, G. Sangalli, A fully "locking-free" isogeometric approach for plane linear elasticity problems: A stream function formulation, Comput. Methods Appl. Mech. Engrg. 197 (2007): 160-172.

[4] R.E. Bank, R.K. Smith, A Posteriori Error Estimates Based on Hierarchical Bases, SIAM Journal on Numerical Analysis, Vol. 30, No. 4 (1993): 921-935.

[5] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, T.W. Sederberg, Isogeometric analysis using T-splines, Comput. Methods Appl. Mech. Engrg. 199 (2010): 229-263.

[6] Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes, G. Sangalli, Isogeometric analysis: approximation, stability and error estimates for h-refined meshes, Math. Mpdels Methods Appl. Sci. 16 (2006): 1031-1090.

[7] Y. Bazilevs, V.M. Calo, Y. Zhang, T.J.R. Hughes, Isogeometric fluid-structure interaction analysis with applications to arterial blood flow, Comput. Mech. 38 (2006): 310-322.

[8] D. Braess: Finite elements, 2nd edition, Cambridge University Press, Cambridge, 2001.
[9] A. Buffa, D. Cho, G. Sangalli, Linear independence of the T-spline blending functions associated with some particular T-meshes, Comput. Methods Appl. Mech. Engrg. 199 (2010): 1437-1445.
[10] A. Buffa, G. Sangalli, R. Vázquez, Isogeometric analysis in electromagnetics: B-splines approximation, Comput. Methods Appl. Mech. Engrg. 199 (2010): 1143-1152.
[11] C. Carstensen, Some remarks on the history and future of averaging technique in a posteriori finite element error analysis, Z. Angew. Math. Mech. 84 (2004) No. 1: 3-21.
[12] P.G. Ciarlet, The Finite Element Method for Elliptic Problems, North-Holland Publishing Co., Amsterdam, 1978.
[13] J.A. Cottrell, T.J.R. Hughes, A. Reali, Studies of refinement and continuity in isogeometric structural analysis, Comput. Methods Appl. Mech. Engrg. 196 (2007): 4160-4183.
[14] J.A. Cottrell, A. Reali, Y. Bazilevs, T.J.R. Hughes, Isogeometric analysis of structural vibrations, Comput. Methods Appl. Mech. Engrg. 195 (2006): 5257-5296.
[15] J. Deng, F. Chen, Y. Feng, Dimensions of spline spaces over T-meshes, Journal of Computational and Applied Mathematics, 194 (2006): 267-283.
[16] J. Deng, F. Chen, X. Li, C. Hu, W. Tong, Z. Yang, Y. Feng, Polynomial splines over hierarchical T-meshes, Graphical Models 70 (2008): 76-86.
[17] M.R. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis by local h-refinement with T-splines, Comput. Methods Appl. Mech. Engrg. 199 (2010): 264-275.
[18] R. Echter, M. Bischoff, Numerical efficiency, locking and unlocking of NURBS finite elements Comput. Methods Appl. Mech. Engrg. 199 (2010): 374-382.
[19] P. Fischer, M. Klassen, J. Mergheim, P. Steinmann, R. Müller, Isogeometric analysis of 2D gradient elasticity, Comput. Mech. (2010), doi 10.1007/s00466-010-0543-8.
[20] D.R. Forsey, R.H. Bartels, Hierarchical B-Spline Refinement, Computer Graphics, Vol. 20, No. 4 (1988): 205-212.
[21] T.J.R. Hughes, J.A. Cottrell, Y.Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Engrg. 194 (2005): 4135-4195.
[22] T.J.R. Hughes, A. Reali, G. Sangalli, Efficient quadrature for NURBS-based isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 199 (2010) 301-313.
[23] H.J. Kim, Y.D. Seo, S.K. Youn, Isogeometric analysis for trimmed CAD surfaces Comput. Methods Appl. Mech. Engrg. 198 (2009) 2982-2995
[24] X. Li, M.A. Scott, On the Nesting Behavior of T-splines, ICES REPORT 11-13, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, May 2011.
[25] X. Li, J. Zheng, T.W. Sederberg, T.J.R. Hughes, M.A. Scott, On Linear Independence of T-splines, ICES REPORT 10-40, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, October 2010.
[26] S. Lipton, J.A. Evans, Y. Bazilevs, T. Elguedj, T.J.R. Hughes, Robustness of isogeometric structural discretizations under severe mesh distortion, Comput. Methods Appl. Mech. Engrg. 199 (2010) 357-373
[27] C. Manni, F. Pelosi, M.L. Sampoli, Generalized B-splines as a tool in isogeometric analysis, Comput. Methods Appl. Mech. Engrg. 200 (2010): 867-881.
[28] N.I. Mußchelischwili, Einige Grundaufgaben zur mathematischen Elastizitätstheorie, Fachbuchverlag, Leipzig, 1971.
[29] N. Nguyen-Thanh, H. Nguyen-Xuan, S.P.A. Bordas, T. Rabczuk, Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids, Comput. Methods Appl. Mech. Engrg. (2011), doi: 10.1016/j.cma.2011.01.018.
[30] O. Pironneau, Finite Elements for Fluids, Masson, Paris, 1989.
[31] M.A. Scott, X. Li, T.W. Sederberg, T.J.R. Hughes, Local Refinement of Analysis-Suitable T-splines, ICES REPORT 11-06, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, March 2011.
[32] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, T-Spline Simplification and Local Refinement, ACM Transactions on Graphics 23, No. 3 (2004): 276-283.
[33] T.W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-Splines and T-NURCCs, ACM Transactions on Graphics 22, No. 3 (2003): 161-172.
[34] R.H. Stogner, G.F. Carey, B.T. Murray, Approximation of Cahn-Hilliard diffuse interface models using parallel adaptive mesh refinement and coarsening with C^1 elements, Int. J. Numer. Meth. Engng No.76 (2008): 636-661.
[35] S.P. Timoshenko, J.N. Goodier, Theory of Elasticity, McGraw-Hill Book Company, New York, 1970.
[36] T.K. Uhm, S.K. Youn, T-spline finite element method for the analysis of shell structures Int. J. Numer. Meth. Engng 80 (2009): 507-536.
[37] Y. Zhang, Y. Bazilevs, S. Goswami, C.L. Bajaj, T.J.R. Hughes, Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow, Comput. Methods Appl. Mech. Engrg. 196 (2007): 2943-2959.