

Adaptive Isogeometric Analysis by Local h-Refinement with T-Splines

Michael R. Dörfel^a Bert Jüttler^b Bernd Simeon^a

^a*Technische Universität München, Centre for Mathematical Sciences,
Chair of Numerical Analysis*

^b*Johannes Kepler University, Linz, Faculty of Natural Sciences and Engineering,
Institute of Applied Geometry*

Abstract

Isogeometric analysis based on NURBS (Non-Uniform Rational B-Splines) as basis functions preserves the exact geometry but suffers from the drawback of a rectangular grid of control points in the parameter space, which renders a purely local refinement impossible. This paper demonstrates how this difficulty can be overcome by using T-splines instead. T-splines allow the introduction of so-called T-junctions, which are related to hanging nodes in the standard FEM. Obeying a few straightforward rules, rectangular patches in the parameter space of the T-splines can be subdivided and thus a local refinement becomes feasible while still preserving the exact geometry. Furthermore, it is shown how state-of-the-art a posteriori error estimation techniques can be combined with refinement by T-Splines. Numerical examples underline the potential of isogeometric analysis with T-splines and give hints for further developments.

Key words:

adaptivity, a posteriori error estimation, isogeometric analysis, NURBS, CAD, T-Splines

1 Introduction

Simulation methods that are able to exactly represent free-form shapes and that circumvent the bottleneck of mesh generation hold great promise. Though the idea of bridging the gap between Computer Aided Design (CAD) and the Finite Element Method (FEM) can be traced back a long time, it has only recently reached significant momentum by the introduction of the so-called *Isogeometric Analysis* by Hughes et al.[14].

By extending the isoparametric concept of the standard FEM to more general basis functions such as B-Splines and NURBS (Non-Uniform Rational B-Splines) that are common in CAD approaches, it is possible to fit exact geometries at the coarsest level of discretization and eliminate geometry errors from the very beginning. Moreover, improved convergence and smoothness properties of the numerical solution have been shown [3], and recent results for vibrational analysis [7] and bloodflow simulations [20] underline the potential of isogeometric analysis.

Several key issues, on the other hand, are still open in this new and emerging field, among them the development of competitive codes and adaptive mesh refinement based on a posteriori error estimation techniques. In this paper, we concentrate on the latter issue and present a local refinement method that uses the T-splines introduced by Sederberg et al. [16,17] as appropriate function space.

Due to the research progress of the last two decades, the methodology for Adaptive Finite Element Methods (AFEM) is well-developed today. For a survey see, e.g., Carstensen [6]. Sophisticated algorithms and specific analysis techniques have demonstrated the advantages of locally refined meshes. However, though being well understood and available in various academic codes [2,18], adaptive methods with a posteriori error control have so far not become widespread in commercial software. One of the reasons for this shortcoming is supposedly the complex task of mesh generation, in particular for 3D shapes where the practitioners have little trust in automatic refinement procedures and prefer to rely on their hand-designed meshes.

Within the framework of isogeometric analysis, refined meshes preserve the exact geometry. Using the NURBS as basis functions, however, suffers from the drawback of a rectangular grid of control points in the parameter space, which renders a purely local refinement impossible. In CAD, the idea of using locally refinable tensor-product splines was pioneered by Forsey and Bartels [10], who introduced hierarchical splines as an accumulation of tensor-product splines with different knots and domains. Weller and Hagen [19] discussed tensor-product splines with knot segments. They defined a spline space over a general mesh, but the basis functions are induced by semiregular ones. The work in this paper is based on the new notion of T-splines which was established in [16,17]. In particular, T-splines allow the introduction of so-called T-junctions, which are related to hanging nodes in the standard FEM. Obeying a few straightforward rules, rectangular patches in the parameter space of the T-splines can be subdivided and thus a local refinement becomes feasible.

Based on this observation, we combine here a state-of-the-art error estimation technique by Bank and Smith [1] with refinement by T-Splines. Starting with a coarse rectangular grid and a NURBS basis, the approach successively converts

the mesh into a so-called T-mesh. Numerical results for standard 2D examples illustrate the approach and confirm the expected superior performance of an adaptive isogeometric analysis.

The paper is organized as follows. In Section 2, we give a short introduction to the basic idea of isogeometric analysis and compare the NURBS with standard isoparametric finite elements. Section 3 is devoted to a discussion of NURBS and T-Splines, with particular emphasis on local refinement strategies. The a posteriori error estimator is presented in Section 4, and finally, computational examples are given and analysed in Section 5.

2 Galerkin Projection: Piecewise Polynomials versus NURBS

In this section we define the problem class and summarize the most important features of NURBS as ansatz functions in isogeometric analysis. To keep the presentation simple, we restrict ourselves to linear stationary problems in $d = 2$ dimensions.

Let L be a second order elliptic operator on the Lipschitz domain Ω with boundary $\Gamma = \Gamma_D \cup \Gamma_N$. We want to solve the partial differential equation

$$Lu = f \quad \text{in } \Omega \quad (1)$$

with the boundary conditions

$$u = 0 \quad \text{on } \Gamma_D \quad \text{and} \quad \langle \nabla u, \mathbf{n} \rangle = h \quad \text{on } \Gamma_N$$

for the unknown $u : \Omega \rightarrow \mathbb{R}$ with given data f and h and outer normal vector \mathbf{n} on Γ_N . Without loss of generality, we assume in this section zero Dirichlet boundary conditions on Γ_D .

The weak form of (1) takes the standard form: Find $u \in V$ such that

$$a(u, v) = l(v) \quad \text{for all } v \in V \quad (2)$$

where the appropriate function space is given by

$$V := \{u \in H^1(\Omega), u|_{\Gamma_D} = 0\}.$$

As usual, $a : V \times V \rightarrow \mathbb{R}$ denotes the symmetric bilinear form that corresponds to the operator L and $l : V \rightarrow \mathbb{R}$ a linear functional that contains the right hand side term f and the Neumann term h . The bilinear form a is assumed to be *continuous* and *coercive* such that a unique solution of (2) exists (see [5, Chapt. 2] for definitions and proofs). It defines the so-called energy norm for

all $v \in V$

$$\|v\|_E = \sqrt{a(v, v)}.$$

As indicated in the definition of the space V , the Sobolev space $H^1(\Omega) \subset L_2(\Omega)$ provides the basic setting in the following; it consists of those functions in $L_2(\Omega)$ that possess weak and square-integrable derivatives.

The Galerkin projection replaces the infinite-dimensional space V by the finite dimensional subspace $S \subset V$ that is spanned by n basis functions ϕ_i , $i = 1, \dots, n$. Thus, the approximate solution $u_h \in S$ has to satisfy

$$a(u_h, v_h) = l(v_h) \quad \text{for all } v_h \in S.$$

We write $u_h = \sum_{i=1}^n q_i \phi_i$ with unknown real coefficients q_i and obtain in the usual way the linear system

$$Aq = b \tag{3}$$

where A is the stiffness matrix and b the load vector. They are defined by $A_{ij} = a(\phi_j, \phi_i)$ and by $b_i = l(\phi_i)$ for $i, j = 1, \dots, n$, respectively.

The choice of the subspace S in the Galerkin projection depends on several criteria. Obviously, if the approximate solution u_h is not sufficiently accurate, there should be a straightforward and efficient way to enlarge S . This issue of *refinement* will be enlightened in the next section. For the moment, we discuss the various choices from a more general point of view.

The following properties are desirable in our context:

- (i) **Convergence:** If we refine, the approximative solution should converge to the exact solution. While local refinement is preferred in practice, uniform refinement is the basis for standard convergence proofs.
- (ii) **Regularity:** We aim at conforming methods with basis functions at least in $H^1(\Omega)$; in contrast to conventional FEM wisdom, additional global smoothness is regarded as beneficial.
- (iii) **Support:** The basis functions should have a small and compact support.
- (iv) **Accurate representation of geometry:** complex geometries should be exactly resolved already on coarse grids.

We now briefly discuss these items for the well-established isoparametric FEM and the isogeometric analysis based on NURBS.

2.1 Isoparametric Approach

In standard FEM the basis functions ϕ are chosen as piecewise polynomials, and the concept of *isoparametric* elements is invoked to approximate curved boundaries. Assume we have m shape functions N_j , e.g. polynomials, defined

over a standard geometry $\Omega_0 \subset \mathbb{R}^2$ like a triangle or a square – in $d = 3$ dimensions a tetrahedron or a hexahedron. We call Ω_0 the parameter domain or parameter space. The computational domain Ω is partitioned into a mesh of elements T_k that are subdomains of the same shape as Ω_0 . On each element there are m specific grid points x_j^k , e.g., the corners, that can be used to define the geometry functions $G_k : \Omega_0 \rightarrow T_k$ by

$$G_k(\xi) := x(\xi) = \sum_{j=1}^m N_j(\xi)x_j^k,$$

see also Fig. 1.

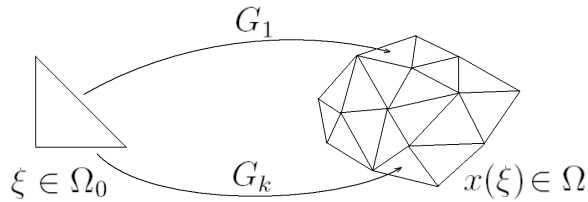


Fig. 1. The parameter space Ω_0 is mapped with different local geometry functions to the triangulation of the computational domain

The basis functions ϕ_i for the Galerkin projection are compositions of the shape functions with the inverse of the geometry function. So for $x \in T_k$ we get the *local* representation

$$u_h(x) = \sum_{j=1}^m N_j \circ G_k^{-1}(x)q_j^k. \quad (4)$$

where q_j^k stands for the unknown coefficients or nodal values. This local representation offers a local evaluation of element stiffness matrices and load vectors, and the linear system (3) is then assembled from these element contributions. For the technical details we refer to [13].

Refinement in isoparametric FEM is either performed by splitting the elements into smaller ones (*h*-refinement) or by using higher order polynomials as shape functions in each element (*p*-refinement). Well-established a posteriori error estimators as well as mesh refinement algorithms are available. Moreover, polynomials as local basis functions can be easily evaluated and integrated. Note that global smoothness is C^0 in general.

The most obvious drawback of isoparametric FEM is the lack of an exact geometry representation for complex engineering shapes. In this case the boundary must be approximated and also the boundary conditions, which may lead to additional errors or even wrong boundary layers, see [14].

2.2 Isogeometric Analysis

Isogeometric analysis based on the NURBS as basis functions allows to exactly map the unit square in the parameter space \mathbb{R}^2 to an arbitrary domain that was designed in a (NURBS-based) CAD-program. The global geometry function $G : \Omega_0 := [0, 1]^2 \rightarrow \Omega$ is element of a NURBS space $\mathcal{R}_{\mathbf{p}}$ that will be described in the next section. With the control points of the NURBS $P_i \in \mathbb{R}^2$ in linear ordering ($i = 1, \dots, n$), the geometry function is defined for all $\xi \in \Omega_0$ by

$$G(\xi) := x(\xi) = \sum_{i=1}^n N_i(\xi) P_i,$$

see also Fig. 2.

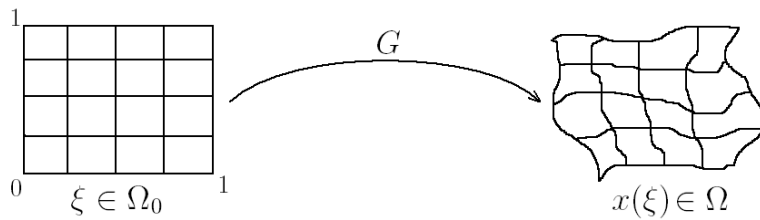


Fig. 2. The parameter space Ω_0 is mapped with one global geometry function to the computational domain

The representation of the approximative solution u_h stays the same as in the isoparametric case, except we are now already in a global setting. So we have for all $x \in \Omega$

$$u_h(x) = \sum_{i=1}^m N_i \circ G^{-1}(x) q_i. \quad (5)$$

It is important to understand that this method does not use elements in the classical sense but *patches* instead. The knot vectors defining a NURBS create two-dimensional boxes in the preimage of G , their image under G is called a patch.

Compared to the piecewise polynomials in the classical FEM, the basis functions are now globally defined and have a larger support, including more patches. Global smoothness can be easily increased to C^1 or even higher.

Isogeometric analysis based on NURBS offers the possibility for h -refinement by inserting new knots in the knot vectors and for p -refinement by increasing the degree of the NURBS. Additionally, the so-called k -refinement is available that combines the h - and p -refinement strategy. However, all three refinements are global in nature, and it is clear that there is a need for more general basis functions that feature local subdivision while still maintaining the favorable properties of NURBS.

3 From NURBS to T-Splines

In this section, we describe the NURBS in more detail and present the generalization to T-splines. Moreover, the issue of local refinement is addressed.

3.1 NURBS as Basis Functions

NURBS were first used as ansatz functions in the isogeometric analysis by Hughes et al. [14]. Referring to the standard reference by Piegl/Tiller [15], we summarize here only the main properties. Since NURBS are defined by means of B-splines, we firstly introduce these functions.

3.1.1 B-Splines

A family of univariate B-splines is defined by a *degree* $p \in \mathbb{N}_0$ and a *knot vector* $\Xi = \{\xi_1, \dots, \xi_m\}$. The latter is a monotonically increasing sequence of real coordinates, the *knots*. We will use non-uniform and open knot vectors where the first and the last knot are repeated $p+1$ times. All other knots have a multiplicity between one and p .

The $n = m - p - 1$ basis functions $B_{i,p}$, which are called B(asis)-splines, can be defined by the Cox-de Boor iteration (fractions with zero denominators are considered zero):

$$B_{i,0}(x) = \begin{cases} 1 & \text{for } \xi_i \leq x < \xi_{i+1} \\ 0 & \text{else} \end{cases}$$

$$B_{i,p}(x) = \frac{x - \xi_i}{\xi_{i+p} - \xi_i} B_{i,p-1}(x) + \frac{\xi_{i+p+1} - x}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1}(x)$$

Bivariate B-splines are defined by a tensor product of two univariate basis functions with possibly different degrees p_1 and p_2 and knot vectors Ξ_1 and Ξ_2 . With the multi-indices $\mathbf{i} = (i_1, i_2)$, $\mathbf{p} = (p_1, p_2)$ and $\mathbf{n} = (n_1, n_2)$ we get for all $\mathbf{x} = (x_1, x_2)$

$$B_{\mathbf{i},\mathbf{p}}(\mathbf{x}) = B_{i_1,p_1}(x_1) \cdot B_{i_2,p_2}(x_2)$$

Finally we introduce B-spline surfaces. Assume that the control mesh is given, which consists of $n_1 \cdot n_2$ control points $P_{\mathbf{i}} \in \mathbb{R}^2$. The B-spline surface C_B is defined as

$$C_B(\mathbf{x}) = \sum_{\mathbf{i}=1}^{\mathbf{n}} B_{\mathbf{i},\mathbf{p}}(\mathbf{x}) P_{\mathbf{i}} \quad \text{for } \mathbf{x} \in [\xi_1^{(1)}, \xi_{m_1}^{(1)}] \times [\xi_1^{(2)}, \xi_{m_2}^{(2)}]. \quad (6)$$

Since the control points are in \mathbb{R}^2 , the B-spline surface describes simply a portion of the plane.

3.1.2 NURBS

Now we have all prerequisites to construct a NURBS surface. In addition to the two knot vectors and the control points $P_i \in \mathbb{R}^2$, we need one weight w_i for each control point, which is often restricted to be positive $w_i > 0$. The weights and control points define the *projective control points*

$$P_i^w = \begin{pmatrix} w_i P_i \\ w_i \end{pmatrix} \in \mathbb{R}^3.$$

Using these points in \mathbb{R}^3 we construct a projective B-spline surface in \mathbb{R}^3 and dehomogenize it. This gives the NURBS surface

$$C_R(\mathbf{x}) = \frac{\sum_{i=1}^n B_{i,\mathbf{p}}(\mathbf{x}) w_i P_i}{\sum_{j=1}^n B_{j,\mathbf{p}}(\mathbf{x}) w_j}.$$

which again parameterizes a segment of the plane \mathbb{R}^2 . Analogously to (6) one may write this formula with the rational basis (NURBS) functions

$$R_{i,\mathbf{p}}(\mathbf{x}) = \frac{B_{i,\mathbf{p}}(\mathbf{x}) w_i}{\sum_{j=1}^n B_{j,\mathbf{p}}(x) w_j}.$$

They span the space

$$\mathcal{R}_{\mathbf{p}}[\Xi] = \text{span}\{R_{i,\mathbf{p}}, \mathbf{i} \in \{1, \dots, n_1\} \times \{1, \dots, n_2\}\}.$$

We mention some properties of these bivariate basis functions since they are relevant for isogeometric analysis:

- (i) Convergence of isogeometric analysis for uniform h -refinement has been proved by Bazilevs et al. [3].
- (ii) The basis functions are continuous. To be more precise, $(p - k_{il})$ directional derivatives in the l th direction at a knot $\xi_i^{(l)}$ exist, where k_{il} is the multiplicity of $\xi_i^{(l)}$ in the knot vector Ξ_l ($l = 1, 2$ and $i = 1, \dots$, number of different knots in Ξ_l).
It follows that $\phi_i := R_{i,\mathbf{p}} \circ G^{-1}$ is also continuous for all \mathbf{i} , thus the span of the ϕ_i is a subset of $H^1(\Omega)$.
- (iii) Their support contains a maximum of $(p_1 + 1)(p_2 + 1)$ two-dimensional boxes, which means that it is compact and relatively small.
- (iv) They are capable of representing the geometry accurately if it was generated by NURBS-based CAD software.

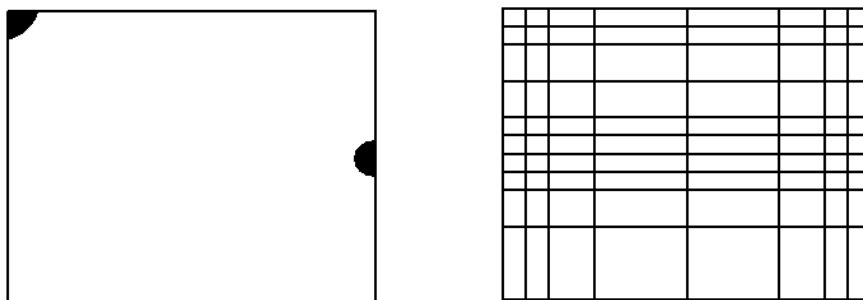
3.1.3 Refinement

The first refinement option for NURBS is the *h-refinement* where we can insert arbitrary new knots in the knot vectors. For example, if we insert one knot in the first knot vector, this operation changes $(p_1 + 1)$ ansatz functions in that direction and adds a new one. Due to the tensor product construction of the bivariate functions, a total of $((p_1 + 2) \cdot n_2)$ NURBS functions are affected by this operation. As we do not want to alter the geometry function, we need to adjust also the corresponding control points. All required formulas and several fast algorithms for this purpose can be found in [15].

The second option is *p-refinement* where we increase the order of the basis functions in one coordinate direction. Because the NURBS surface should again not change, we have to increase the multiplicity of the knots in the knot vector and adjust the control points. Obviously, none of the functions remains unaffected in this case, and if we have s different knots in the first knot vector and increase the first degree to $(p_1 + t)$, this refinement leads to $(st \cdot n_2)$ new basis functions. See again [15] for formulas and algorithms.

Because these two options do not commute, we get various further refinement methods if we connect them in different order. E.g., one can firstly perform all the *p-refinement* steps and then the *h-refinement* steps, which is called *k-refinement*, see [14].

Unfortunately, for dimension $d > 1$ all these refinement methods have a global effect due to the tensor-product mesh structure. We illustrate this issue in Fig. 3a where two areas are marked in black to indicate a need for local refinement caused, e.g., by discontinuous boundary values or reduced regularity of the exact solution. Though local refinement is desirable, the NURBS can only be refined in entire rows or columns, as shown in Fig. 3b.



(a) Areas of large estimated error

(b) Mesh of the knot vectors

Fig. 3. Adaptive refinement with NURBS

3.2 T-splines

In order to overcome the problem of local refinement, we shall now use T-splines instead of NURBS. Initially, T-splines were defined by Sederberg in [16,17] solely for the bivariate case and for degree $p = 3$.

Consider a bivariate NURBS with the same degree $p = 3$. Then each basis function is defined by 5 knots of Ξ_1 and 5 knots of Ξ_2 . If we now associate these functions with grid points defined by the median coordinates of their "small" knot vectors, we can visualize their location in the grid of the knot vectors. To put it the other way round, each intersection of edges in that mesh has thus also at least one corresponding function.

We return to the situation of Fig. 3a to demonstrate the construction of T-splines. Fig. 4 shows the mesh for the NURBS where a filled box \blacksquare marks redundant functions that could be skipped whereas the desired functions are highlighted by an empty box \square .

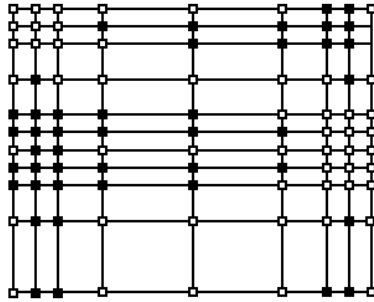


Fig. 4. Mesh of the knot vectors with points corresponding to NURBS basis functions

Now we take only these desired NURBS of the refined mesh and combine them with the NURBS of the original coarse mesh (Fig.5a). This yields a locally refined mesh with the characteristic T-junctions, as shown in Fig. 5b.

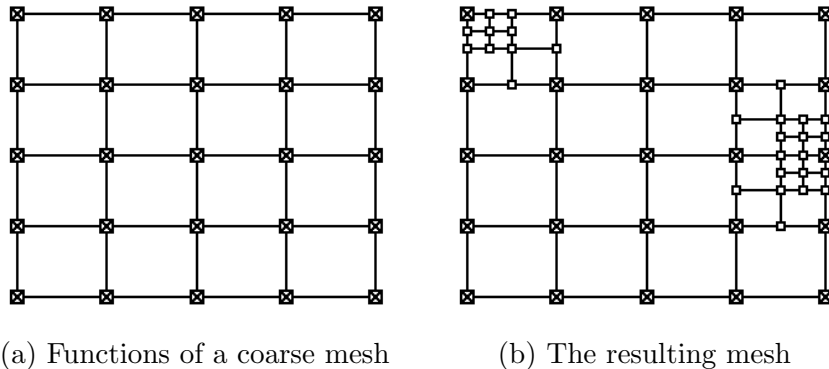


Fig. 5. Introducing the basic idea of T-splines

3.2.1 *T*-mesh and Blending Functions

After this motivation of T-splines, we will next introduce them in a precise way. Because of the different usage, some of our definitions and nomenclature differ slightly from the original work by Sederberg et al. [16,17]. By that, we also overcome the ambiguities in the edge insertion as shown in Fig. 9 in [16].

At first, we give a formal definition of the *T*-mesh as a mesh in an *index space*, not in the parameter space as the mesh in Fig. 5b. With ν_1 and ν_2 denoting integers greater or equal 4, the T-mesh consists of n grid points (or vertices) $p_i \in \{1, \dots, \nu_1\} \times \{1, \dots, \nu_2\}$ connected by horizontal and vertical edges e_j . It has to fulfill the following properties:

- If one edge intersects another, there must be a grid point.
- There is no possibility to connect two grid points with an additional horizontal or vertical edge without intersecting an already existing edge.
- The T-mesh partitions the box $\Omega_I := [1, \nu_1] \times [1, \nu_2]$ into non-degenerated polygons that we call *index patches*. Thus isolated grid points or lines are prohibited.

An example of a T-mesh is shown in Fig. 6.

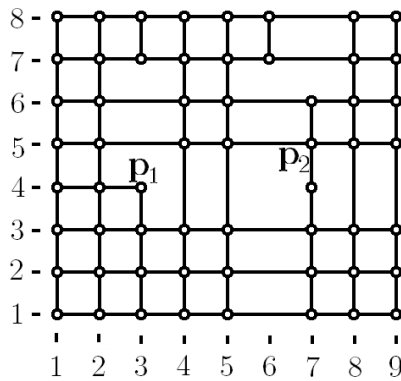


Fig. 6. Example of a T-mesh

In order to associate ansatz functions with the T-mesh, we next define two global knot vectors in the parametric space, $\sigma = \{s_1, \dots, s_{\nu_1}\}$ and $\tau = \{t_1, \dots, t_{\nu_2}\}$ with double first and last knots, i.e. $s_1 = s_2$, $s_{\nu_1-1} = s_{\nu_1}$, $t_1 = t_2$ and $t_{\nu_2-1} = t_{\nu_2}$. All other knots have a multiplicity not higher than 3.

Usually we have a starting T-mesh where the corresponding knot vectors result from the data of bicubic NURBS: We set

$$\sigma = \Xi_1 \setminus \{\xi_1^{(1)}, \xi_2^{(1)}, \xi_{m_1-1}^{(1)}, \xi_{m_1}^{(1)}\} \text{ and}$$

$$\tau = \Xi_2 \setminus \{\xi_1^{(2)}, \xi_2^{(2)}, \xi_{m_2-1}^{(2)}, \xi_{m_2}^{(2)}\}.$$

Note that in this case $\nu_1 = m_1 - 4$ and $\nu_2 = m_2 - 4$.

Furthermore we set on every intersection of grid lines a grid point and everywhere edges connecting adjacent grid points.

In this regular mesh, the index-patches are two dimensional boxes. It can then either be refined or coarsened by inserting or deleting grid points with the only restriction that the resulting mesh preserves the previously mentioned properties. The corresponding algorithms are discussed in [17].

Now we return to the construction of the ansatz functions and define for each point p_i a so-called *blending function* Bl_i . With the following two rules we extract for every grid point five-element knot vectors $\sigma_i = \{\sigma_{i1}, \sigma_{i2}, \sigma_{i3}, \sigma_{i4}, \sigma_{i5}\}$ and $\tau_i = \{\tau_{i1}, \tau_{i2}, \tau_{i3}, \tau_{i4}, \tau_{i5}\}$ from the T-mesh:

Rule 1 The coordinates of the point $p_i = (p1, p2)$ are the indices of the knots in the global knot vectors for the medians of the small knot vectors. Thus $\sigma_{i3} = s_{p1}$ and $\tau_{i3} = t_{p2}$.

Rule 2 The indices for the knots σ_{i4} and σ_{i5} are the first coordinates of the first two points or vertical edges that intersect with the ray $R(\alpha) = (p1 + \alpha, p2)$, $\alpha > 0$ in the T-mesh. If the ray leaves the T-mesh before the coordinates are found, the knot vector is filled up with the last knot s_{ν_1} . The remaining knots of σ_i and τ_i are found by analogous rays.

In order to make these rules clear, we construct the knot vectors for the functions corresponding to p_1 and p_2 of Fig. 6. We get $\sigma_1 = \{1, 2, 3, 4, 5\}$ and $\tau_1 = \{2, 3, 4, 5, 6\}$ as well as $\sigma_2 = \{4, 5, 7, 8, 9\}$ and $\tau_2 = \{3, 4, 5, 6, 7\}$. They are shown in Fig. 7.

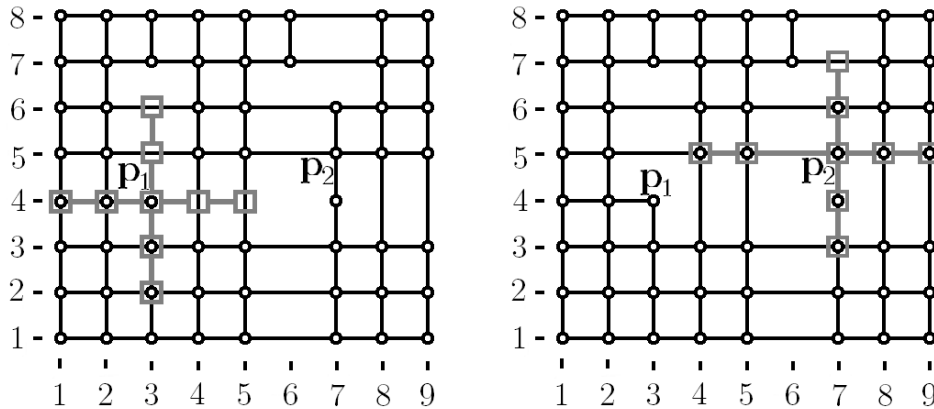


Fig. 7. The knot vectors for p_1 and p_2 in the T-mesh

The blending functions are now defined as a product of the two univariate, cubic B-splines B_{σ_i} and B_{τ_i} , where B_{σ_i} uses the five knots of the knot vector σ_i and B_{τ_i} the ones of the knot vector τ_i . We get for $s, t \in \mathbb{R}$

$$Bl_i(s, t) = B_{\sigma_i}(s) \cdot B_{\tau_i}(t).$$

3.2.2 Normalized Blending Functions and T-Spline Surfaces

Analogously to isogeometric analysis, we do not use the polynomial blending functions as ansatz functions but instead the rational, normalized blending functions N_i , that we will also call *T-spline functions*. They are constructed in the same way as the bivariate NURBS by dehomogenizing a projective surface which is defined by the blending functions surface. For that purpose we need again n control points $P_i \in \mathbb{R}^d$ and weights $w_i > 0$ in linear order and get for $(s, t) \in \Omega_0 := [s_1, s_{\nu_1}] \times [t_1, t_{\nu_2}]$

$$N_i(s, t) := \frac{Bl_i(s, t)w_i}{\sum_{j=1}^n Bl_j(s, t)w_j}.$$

These functions are well defined and span the function space

$$\mathcal{N} := \text{span}\{N_i, i = 1 \dots n\}.$$

They define the T-spline C in the usual way,

$$C(s, t) = \sum_{i=1}^n N_i(s, t)P_i, \quad \text{for } (s, t) \in \Omega_0.$$

Due to the analogous derivation, the bicubic NURBS are a special case of T-spline functions. Every "full" T-mesh like the one used as starting mesh creates NURBS. Before we discuss the adaptive refinement, we list some properties of the T-spline functions.

- (i) A uniform insertion of the grid points into the starting T-mesh results in NURBS and does not create additional blending functions. This also means that the convergence analysis of [3] applies in this case.
- (ii) The T-spline functions are continuous. Furthermore, the $(3 - k_j)$ directional derivatives of N_i in the s or t direction at an edge s_j or t_j exist, where k_j is the multiplicity of s_j in σ_i , respectively of t_j in τ_i . It follows that $\phi_i := N_i \circ G^{-1}$ is also continuous for all i , thus the FEM ansatz functions are in $H^1(\Omega)$.
- (iii) They have a compact support with $\text{supp}(N_i) = [\sigma_{i1}, \sigma_{i5}] \times [\tau_{i1}, \tau_{i5}]$. This is possibly larger than the support of a NURBS built by the global knot vectors σ and τ and leads to the effect that normally more than 16 ansatz functions are non zero in a box as it is using bicubic NURBS. The upper bound depends on the connectivity of the T-mesh – in our experience the number lies between 16 and 20 ansatz functions.
- (iv) They accurately represent the geometry. Like the NURBS the T-mesh inherits the geometry.

3.2.3 General Refinement Rules

So far, T-splines admit solely h -refinement, which is simply accomplished by inserting grid points into the T-mesh. We just want to give a short overview over this process - for a deeper understanding the reader should consult [16] and [17].

If we insert a new grid point into the T-mesh, we have to refine the blending functions, because after the insertion the old blending functions do not fit to the new T-mesh. The following violations of the rules 1 and 2 may occur:

- **Violation 1:** A blending function skips a grid point in the T-mesh.
- **Violation 2:** In a knot vector of a blending function there is a knot that is not induced by a grid point in the T-mesh.
- **Violation 3:** A grid point in the T-mesh has no blending function.

By inserting the new grid point we have to ensure that at least one of the first two violation occurs - else we can not fix the inevitable violation 3. If we insert the grid point on an edge, this problem does not arise.

For fixing the first violation, we have to refine the affected blending function using the standard B-Spline refinement for the corresponding factor of the blending function B_{σ_i} or B_{τ_i} . If the second violation arises, we have to insert an additional (unwanted) grid point in the T-mesh at the relevant coordinates. Both solutions may lead to new violations and thus to a recursive insertion of new grid points (see Section 3.2.5 for an example).

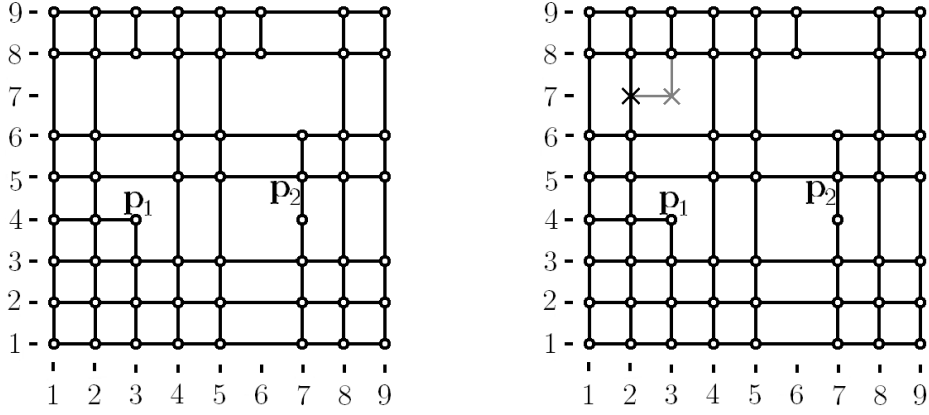
But if there are none of the first two violations left, there is also no violation of type 3 remaining (see [17]).

The last step in the refinement is the adjustment of the control points, again analogously to the B-spline refinement, in order to preserve the representation of the geometry (for more details see again [17]).

Note that in this context it is also possible to enlarge the global knot vectors σ or τ and shift the T-mesh at the indices of the new knot before grid point insertion is performed (see Fig. 8a). In this example, continuing the grid of Fig. 6 and 7, we also want to show a case where a single additional knot occurs (Fig. 8b). Due to the insertion the small knot vector τ_2 of p_2 changes to $\{3, 4, 5, 6, 8\}$, but because of the simultaneous change of the global knot vector τ the blending function belonging to p_2 remains the same.

3.2.4 Adaptive Refinement

Suppose an error estimator or some other kind of automatic or manual refinement indicator is available. There are various possibilities to select the points for insertion based on these data. The following strategy turned out to give good results for most cases, as we will demonstrate later in Section 5.1 and 5.2. However, in some special cases it leads to a too extensive refinement as



(a) Shift of the indices in t -direction because of an insertion of ξ_{new} (b) The newly inserted point is at $(2,7)$, the additional one at $(3,7)$

Fig. 8. Continuing the example of Fig. 6: Enlargement of Ξ_2 with $\xi_{\text{new}} \in [\xi_6^{(2)}, \xi_7^{(2)})$ and grid point insertion leading to an unwanted additional grid point

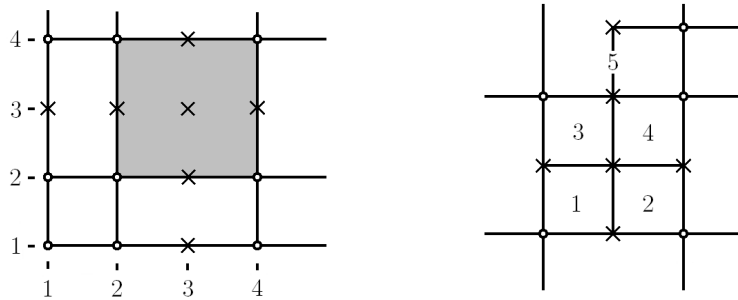
we show in Section 3.2.5 and 5.3.

Refinement Strategy:

We insert grid points on every adjacent edge and at the centres of the index patches marked by the error estimator. If one of the edges refers to a multiple knot (e.g. at the beginning or the end of the knot vector), we also insert a knot on the corresponding edge.

The second part ensures that it is possible to insert points on edges bordering only on index patches where the two s - or (exclusive or) t -coordinates index to the same knot value. This strategy is illustrated in Fig. 9a (remember that the first two indices of a global knot vector must index to the same knot value).

The corresponding index patches are then subdivided into four parts. If the algorithm inserts additional grid points in other index patches, these are handled as if not subdivided to ensure that all of them remain two-dimensional boxes. This rule is shown in Fig. 9b.



(a) Selection of grid points if the grey box has been marked (b) Subdividing the index patch

Fig. 9. Refinement strategy of the T-splines

3.2.5 Cascadic Insertion of Grid Points in the Refinement Process

In this section we discuss a worst case scenario of the refinement process. We start with a simple domain $\Omega = [0, 1]^2$ and choose as parametrization the identity. Although we mark in every step only those patches containing the diagonal from $(0, 0)$ to $(1, 1)$ (marked in grey in Fig. 10), substantially more grid points are inserted due to the refinement algorithm.

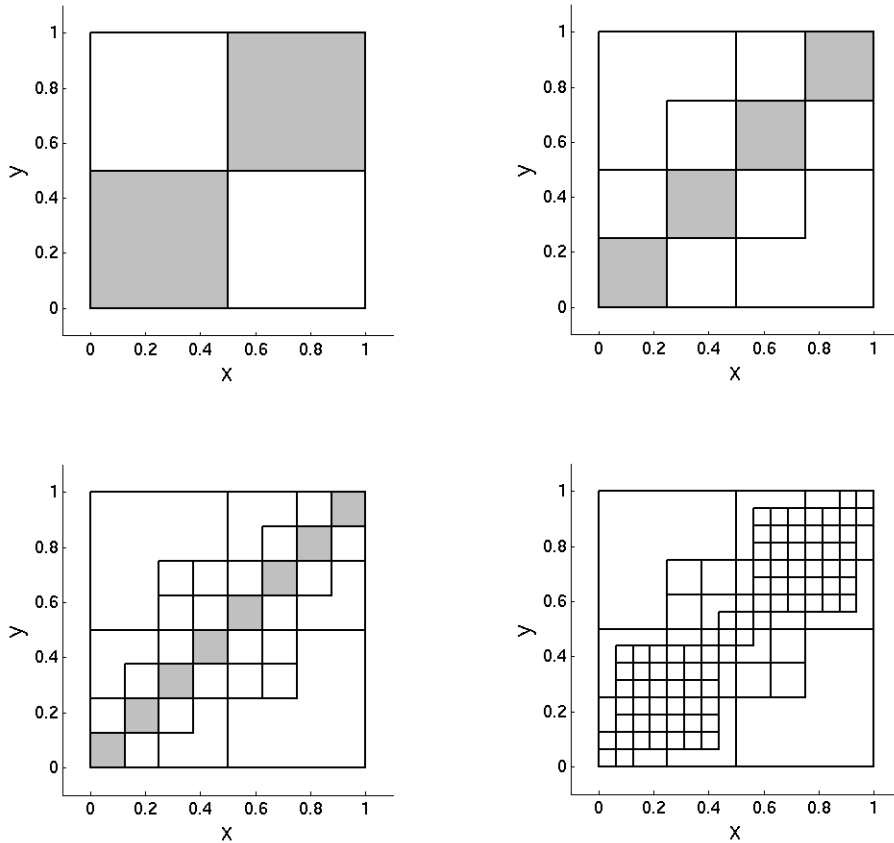


Fig. 10. The T-mesh mapped on Ω : Refinement leads partially to nearly full refinement in this worst case scenario

This is clearly a drawback of the present approach and calls for modifications of the T-spline concept.

With the actual state of the art, the problem can only be fixed if we use some knowledge on the problem at hand. E.g., if we know that there is a boundary layer in diagonal direction, we can try to choose an appropriate parametrization such that one parameter direction is parallel to the layer and the other one normal to it.

4 A Posteriori Error Estimator

In this section we extend a state-of-the-art a posteriori error estimation technique to isogeometric analysis. Let $n_{\mathcal{T}}$ denote the cardinality of the index patches in Ω_I . Due to the construction and refinement of the T-mesh, all index patches are boxes and we can define patches in the computational domain by

$$T_k = G\left([s_{k1}, s_{k2}] \times [t_{k3}, t_{k4}]\right) \subset \Omega \text{ for the index-patch } [k1, k2] \times [k3, k4].$$

Moreover, by \mathcal{T} we denote the set of all patches T_k .

We next aim at refining only those patches T_k which contribute significantly to the error $u - u_h$. However, as the exact solution u of our problem is unknown, we have to compute this information by an *error estimator* $\eta : \Omega \rightarrow \mathbb{R}$ which satisfies

$$\eta(T_k) \approx c \| |u|_{T_k} - u_h|_{T_k} \|_{\star}$$

with a suitable norm $\|\cdot\|_{\star}$, typically the energy norm, and a real constant $c > 0$. As the constant c indicates, we do not require an approximation of the error itself – for practical reasons it is sufficient that the error estimator behaves like the error. For that purpose it has to be *reliable* and *efficient* (see Carstensen [6] for detailed definitions). The first property guarantees that the error estimator does not systematically underestimate the true error while the second property ensures that the estimator behaves asymptotically like the true error if the true error tends to zero.

Assume we have an estimator with these properties available. We then mark each patch T_k to be refined, if $\eta(T_k)$ exceeds a certain threshold θ . Often this threshold depends on the results of the estimator, e.g.,

$$\theta = \begin{cases} \alpha \cdot \max_k \{ \eta(T_k) \} & \text{with } \alpha \approx 0.5 \\ \alpha\text{-quantile}_k \{ \eta(T_k) \} & \text{with } \alpha < 1 \end{cases}$$

Among the different approaches available, our choice for an error estimator to be used in combination with T-splines takes up the idea of hierarchical bases and bubble functions, introduced by Deuffhard/Leinen/Yserentant in [9]. In this method we enlarge our Galerkin subspace S by another, disjoint subspace $\mathcal{W} \subset V$. This leads to a new subspace

$$\bar{S} = S \oplus \mathcal{W}.$$

For a simple and efficient implementation, we choose \mathcal{W} as the function space spanned by the so-called *bubble functions*. The univariate bubble functions are

defined with two real parameters $a < b$ as

$$w[a, b](x) = \begin{cases} \frac{x-a}{b-a} \cdot \frac{b-x}{b-a}, & x \in (a, b) \\ 0 & , \text{ else} \end{cases}$$

They have a small and compact support, the intervall $[a, b]$. For each patch T_k they are now used to define a bivariate bubble function $W_k : \Omega \rightarrow \mathbb{R}$,

$$W_k(x, y) = w[s_{k1}, s_{k2}](s) \cdot w[t_{k3}, t_{k4}](t) \quad \text{with } (s, t) = G^{-1}(x, y).$$

Making use of these bubble functions, we can write the subspace \mathcal{W} as

$$\mathcal{W} = \text{span} \{W_k, k = 1, \dots, n_{\mathcal{T}}\}.$$

In order to estimate the error of u_h , we have the straightforward choice of creating a large system of equations for the Galerkin projection of $\bar{u}_h \in \bar{S}$, similar to (3), and evaluating the difference $u_h - \bar{u}_h$. This approach, however, has two drawbacks. One is the increased size of the linear system, and the other one is compatibility of the basis functions in S and \mathcal{W} . Though the bubble functions also preserve the geometry by construction, it is not clear how they match with the T-splines in terms of smoothness and approximation properties.

A way out is to use the subspace \mathcal{W} simply for error estimation purposes and not for constructing a better approximation in an enlarged space \bar{S} . As suggested by Bank and Smith in [1], we are thus looking for an approximation e_h of the difference $u - u_h$ in \mathcal{W} . This leads to the following weak form for the error estimator: Find $e_h \in \mathcal{W}$ such that

$$a(e_h, v) = l(v) - a(u_h, v) \quad \text{for all } v \in \mathcal{W}$$

Writing $e_h = \sum_{k=1}^{n_{\mathcal{T}}} \epsilon_k W_k$, we get

$$A^{\mathcal{W}, \mathcal{W}} \epsilon = b^{\mathcal{W}} - A^{\mathcal{W}, S} q$$

where q is the old coefficient vector of u_h from the linear system (3) and $A^{\mathcal{W}, \mathcal{W}}$ the stiffness matrix defined by $A_{k,l}^{\mathcal{W}, \mathcal{W}} = a(W_l, W_k)$ for $k, l = 1, \dots, n_{\mathcal{T}}$. Moreover, $b^{\mathcal{W}}$ is the right hand side vector and $A^{\mathcal{W}, S}$ a rectangular matrix, which are given by $b_k^{\mathcal{W}} = l(W_k)$ and $A_{k,i}^{\mathcal{W}, S} = a(\phi_i, W_k)$ for $k = 1, \dots, n_{\mathcal{T}}$ and $i = 1, \dots, n$. Because the bubble functions have disjoint supports, the corresponding stiffness matrix $A^{\mathcal{W}, \mathcal{W}}$ is a diagonal matrix and therefore almost for free to invert.

The error estimator η is now defined for $T_k \in \mathcal{T}$ by

$$\eta(T_k) := \|e_h|_{T_k}\|_E = \|\epsilon_k W_k\|_E = \epsilon_k \sqrt{a(W_k, W_k)} = \epsilon_k \sqrt{A_{k,k}^{\mathcal{W}, \mathcal{W}}}.$$

It is reliable and efficient, if the saturation assumption on \bar{S} and the strict Cauchy inequality for $a|_{S \times \mathcal{W}}$ hold with fixed constants in the whole refinement algorithm (for definitions and proofs see [1]). Numerical tests revealed that the first assumption is the critical one – its validity depends on the example.

5 Computational Examples

In this section we present results of our adaptive refinement algorithm for three two-dimensional examples. We compare the adaptive T-splines solution with the solution achieved by bicubic NURBS (degree $p = 3$) and uniform h -refinement. We start with the stationary heat conduction on an L-shaped domain that leads to a local decrease of regularity. The second example is from linear elasticity and features stress peaks that should be detected. Finally, the last example is an advection dominated flow with discontinuous boundary conditions that introduce sharp layers in the interior and near the boundary of the domain.

In all examples we also compute the condition number κ of the stiffness matrix. It turns out that this number is only moderately increasing due to the adaptive refinement, which indicates that the T-spline functions are linearly independent. However, no formal proof is available yet.

5.1 Stationary Heat Conduction: L-Domain

We consider the L-shaped domain $\Omega = [-1, 1]^2 \setminus [0, 1]^2$ and use the function $f : \mathbb{R}^+ \times (0, 2\pi]$,

$$f(r, \varphi) = r^{\frac{2}{3}} \sin\left(\frac{2\varphi - \pi}{3}\right)$$

as exact solution. After transformation to Cartesian coordinates, f satisfies the Laplace equation $\Delta f = 0$. Accordingly, we solve $\Delta u = 0$ on Ω with zero Dirichlet conditions at the part of the boundary that lies on the axes $0 = f\left(r, \frac{\pi}{2}\right) = f\left(r, \frac{3\pi}{2}\right)$ and Neumann conditions determined by the exact solution f elsewhere. See also Fig. 11 for the problem definition.

We do not model the geometry with double control points and use instead ansatz functions that are continuous but not differentiable at the straight line connecting the corners $(-1, -1)$ and $(0, 0)$. Global smoothness is thus C^0 in the isogeometric analysis, which leads here to better performance of the Gaussian quadrature and the error estimation. The corner singularity implies that the solution is not in $H^2(\Omega)$, and thus classical convergence theory [4,5] does

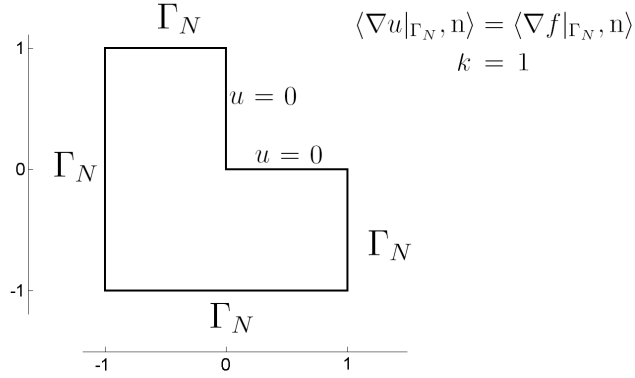
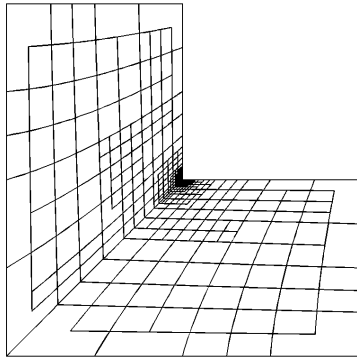


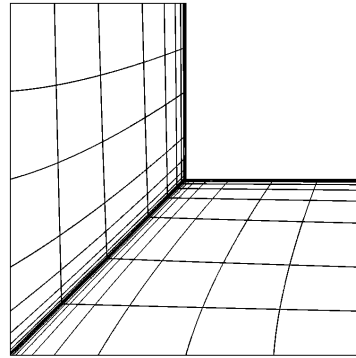
Fig. 11. Setup for the stationary heat conduction - \mathbf{n} is the outer normal vector

not hold. On the other hand, a good error estimator should detect the singularity and initiate appropriate local refinement steps. In Fig. 12a we can see the resulting T-mesh where we used the error estimator and the refinement strategy of the last section. The area of interest is detected and resolved locally. The condition of the stiffness matrix increases from $\kappa_0 = 110.0$ from the initial NURBS mesh to $\kappa_{T8} = 949.9$.

Because we know where the refinement needs to be done, we compare the adaptive results not only to isogeometric analysis with uniform refining strategy, but also with a 'rule of thumb' refinement strategy. The preimage of the reentrant corner under G is $(0.5, 1)$, and thus we only insert new knots in Ξ_1 in the intervals with start or end knot 0.5 and in Ξ_2 in the last two intervals. The resulting mesh is shown in Fig. 12b, the condition of the stiffness matrix increases to $\kappa_{rot7} > 3 \cdot 10^5$.



(a) The T-mesh after 8 refinement steps, mapped on Ω - 652 DOF



(b) The 'rule of thumb' refined NURBS mesh after 7 refinement steps, mapped on Ω - 475 DOF

Fig. 12. Refinements for the heat example

In Fig. 13 we measure the convergence of the numerical methods in the L^2 - and energy norm. As foreseen, the uniform refinement decreases the error only slightly, while the adaptive refinement with the T-splines performs best. The manual 'rule of thumb' refinement is suitable for the very first refinement

steps where it can compete with the adaptive strategy. After this initial phase, however, no further error reduction can be observed for this manual approach.

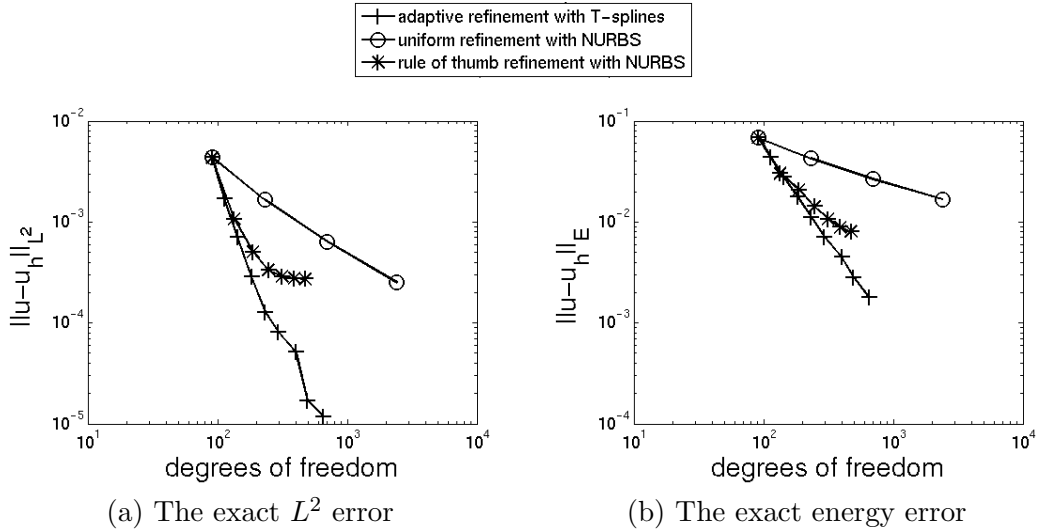


Fig. 13. Comparison of the convergence of the adaptive algorithm versus the two non-adaptive ones

5.2 Linear Elasticity: Plate with a Circular Hole

This example is taken from the initial paper on isogeometric analysis [14]. We consider the plate with circular hole in its center and study the linear elastic behavior of the displacement field $u : \Omega \rightarrow \mathbb{R}^2$ described by

$$\operatorname{div} \sigma(u) = 0.$$

The boundary conditions model an in-plane force in x -direction applied at both sides of the plate. Due to this force we get a peak stress in x -direction at the upper side of the hole. For a homogeneous and isotropic material this problem features an exact solution that can be found in [12]. Due to symmetry, it suffices to simulate solely one quarter of the plate. We have Dirichlet boundary conditions along the coordinate axes, a free boundary at the hole and the exact solution applied as a Neumann condition at the rest of the boundary. The setup is displayed in Fig. 14.

As in the first example, we model the geometry without a double control point to improve the Gaussian quadrature. Hence the ansatz functions are not differentiable at the straight line connecting the upper lefthand corner with the middle of the quarter hole, and global smoothness is again C^0 . In the two subdomains divided by the straight line, however, we have locally C^1 ansatz functions.

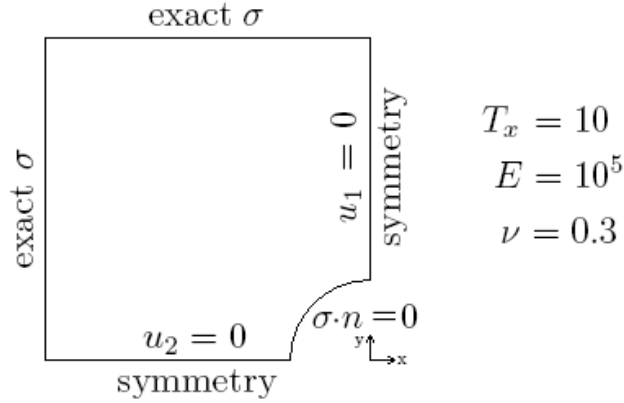


Fig. 14. Setup for the plate with a circular hole

In this example we study the maximal stress in x -direction σ_x within the plate. The exact solution predicts this value at the upmost point of the quarter hole to be $\sigma_x^{\max} = 3T_x = 30$. Fig. 15a shows the error $\sigma_x^h - \sigma_x^{\max}$ where σ_x^h is the actual result achieved by the adaptive respectively the non-adaptive isogeometric analysis. We see again that the adaptive algorithm leads to a faster error reduction.

However if we compare the errors in the energy norm, we get only a slight improvement.

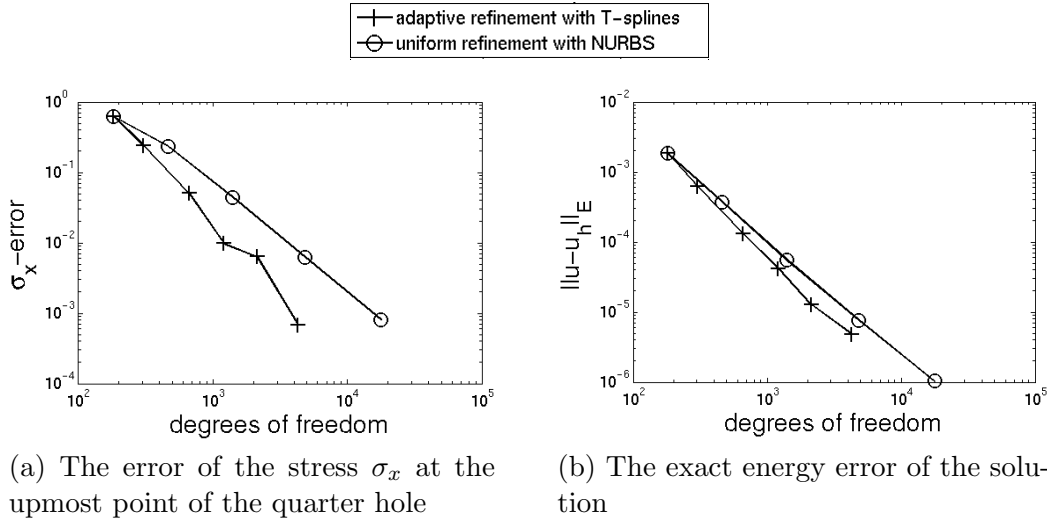


Fig. 15. Comparison of adaptive and non-adaptive solution

The resulting T-mesh is shown in Fig. 16. Here we can see that the algorithm detects the critical areas nicely where the stress in x -direction is maximal (at $(0, 1)$) and minimal (near $(-1, 0)$). The condition of the stiffness matrix grows from $\kappa_0 = 9.1 \cdot 10^5$ to $\kappa_{T5} = 1.9 \cdot 10^6$.

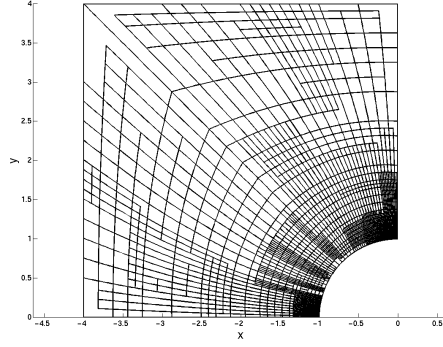


Fig. 16. The T-mesh after 5 refinement steps, mapped on Ω – 4302 DOF

5.3 Fluid Analysis: Advection Dominated Advection-Diffusion Problem

This example is also adopted from [14]. We solve the advection-diffusion equation

$$\kappa \Delta u + \mathbf{a} \cdot \nabla u = 0$$

in the unit square with discontinuous Dirichlet boundary conditions as shown in Fig. 17. In order to optimally resolve this discontinuity we use ansatz functions that are not differentiable at the horizontal line starting from $(0, 0.2)$. The diffusion coefficient is set to $\kappa = 10^{-6}$, and the advection velocity is $\mathbf{a} = (\sin \theta, \cos \theta)$, which yields a high Peclet number. Thus the problem is advection-dominated and sharp layers arise that start at the discontinuity of the boundary condition.

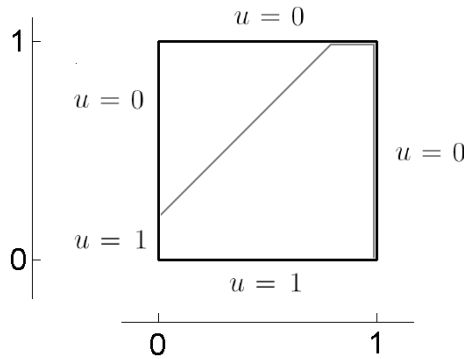


Fig. 17. Setup for the advection dominated advection-diffusion problem - the estimated position of the sharp layers is marked in grey for $\theta = 45^\circ$

The strong advection requires the use of SUPG stabilization (see [11]) with stabilization factor $\tau_T = h_T/2a$ and $h_T = \text{diam}(T)/(\sqrt{2} \cdot \max(\cos \theta, \sin \theta))$.

Since there is no exact solution, we skip a comparison between the adaptive and the non-adaptive algorithms and give in Fig. 18 solely data from the adaptive runs. On the left, the T-meshes mapped on Ω are displayed, in

the middle the numerical solutions at different refinement levels, and on the right the patches that were marked by the a posteriori error estimator at the corresponding level.

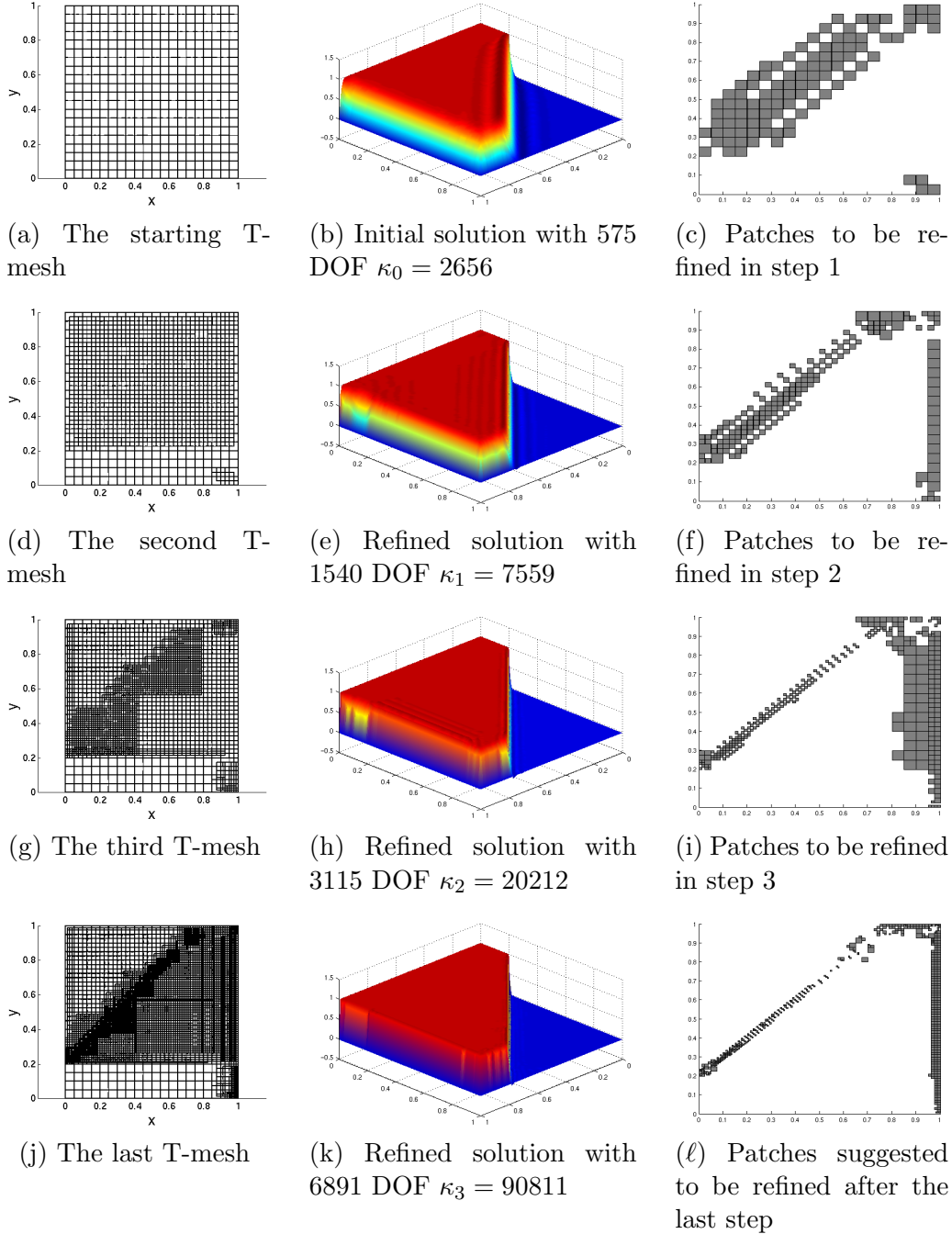


Fig. 18. Solving the advection diffusion problem with T-splines

We can see that the error estimator detects the internal layer as well as the boundary layer. However, the refined patches above and below the diagonal layer do not stem from the refinement strategy but from the insertion of additional grid points in the T-mesh grid point insertion algorithm (as shown in Section 3.2.5). Note that we refine our blending functions firstly in the hori-

zontal direction and then in the vertical one, which leads to more refinement in the s -direction than in the t -direction.

6 Conclusion

In this paper, we have shown that T-splines have the potential to overcome the tensor-product mesh structure of NURBS in isogeometric analysis. Combined with a straightforward a posteriori error estimator, we obtain an adaptive algorithm that generates locally refined meshes and leads to substantial gains in efficiency.

However, as one observes by comparing the first and the last column in Fig. 18, in some cases the refinement of T-splines is not as local as one would hope, since the insertion of grid points may trigger a whole chain of additional refinement steps. We anticipate that this problem might become more severe for the three-dimensional case.

Also, we remark that several theoretical questions concerning T-splines are open, among them criteria for the linear independence of the blending functions and the precise characterization of the corresponding function spaces.

Recently, the dimension of certain polynomial spline spaces over T-meshes has been analyzed in [8]. If the degree is higher than $(2s + 1, 2s + 1)$ for splines of smoothness s , this dimension can be computed and a basis of the space can be constructed. As a potential advantage of this approach, local refinement without triggering additional knot insertions is supported. We feel that this latter notion of spline spaces over T-meshes is promising for isogeometric analysis though important issues such as the choice of a suitable (non-negative) basis still need to be addressed.

Acknowledgement

The authors would like to thank T.J.R. Hughes, Y. Bazilevs, and J.A. Cottrell for inspiring discussions on the topic. Valuable hints and suggestions of three unknown referees, which improved the first version of this contribution, are also acknowledged.

The first author was supported by the International Graduate School of Science and Engineering (IGSSE) of the Technische Universität München within Project 2-11, and the second author by the Austrian Science Fund through the Research Network on Industrial Geometry, Subproject S9202.

A Data for the examples

In this section we briefly summarize the data that is necessary for constructing the basic geometry of the examples presented in Section 3.2.5 and 5. The starting T-mesh is then obtained by p -refinement until $p_1 = p_2 = 3$ and , for the examples in Section 5, additional knot insertions to obtain an already finer mesh for the starting computations.

A.1 Worst Case Scenario: Unity Square

For the worst case scenario in Section 3.2.5 we used $p_1 = p_2 = 1$,

$$\Xi_1 = \{0, 0, 0.5, 1, 1\} \text{ and}$$

$$\Xi_2 = \{0, 0, 0.5, 1, 1\}.$$

Because of the simple shape of the domain $\Omega = [0, 1]^2$ the geometry function is the identity – but to work here analogously to the other examples we construct the identity as a T-spline using the control points in Table A.1. The weights w_i are all 1.

i	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$
1	(0, 0)	(0, 0.5)	(0, 1)
2	(0.5, 0)	(0.5, 0.5)	(0.5, 1)
3	(1, 0)	(1, 0.5)	(1, 1)

Table A.1

Mesh of the control points for the unity square

A.2 Stationary Heat Conduction: L-Domain

For the first example of chapter 5 we used $p_1 = p_2 = 2$,

$$\Xi_1 = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\} \text{ and}$$

$$\Xi_2 = \{0, 0, 0, 1, 1, 1\}.$$

Note that the double knot in Ξ_1 at 0.5 leads to a reduction of the regularity: The basis functions with 0.5 twice in their support (and with them the geometry function) are only continuous at the line $\xi_1 = 0.5$ but not differentiable with respect to the first coordinate. This statement holds also after p - or h -refinement.

The L-domain is then constructed from the control points in table A.2 and again constant weights $w_i = 1$, for all $i = 1, \dots, 15$. Hence the denominator of the NURBS is also constant and we have polynomial functions – B-splines. Because the denominator remains unchanged through refinement, also all T-spline functions are polynomials in this example.

i	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$
1	(-1, 1)	(-0.6, 1)	(0, 1)
2	(-1, 0)	(-0.55, 0)	(0, 0.5)
3	(-1, -1)	(-0.5, -0.5)	(0, 0)
4	(0, -1)	(0, -0.55)	(0.5, 0)
5	(1, -1)	(1, -0.6)	(1, 0)

Table A.2

Mesh of the control points for the L-domain

A.3 Linear Elasticity: Plate with a Circular Hole

Although this example is from [14], we do not use the data given therein because the modelling with a double knot in the upper lefthand corner leads to a singularity and problems for the numerical quadrature. Thus we use $p_1 = p_2 = 2$,

$$\Xi_1 = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\} \text{ and}$$

$$\Xi_2 = \{0, 0, 0, 1, 1, 1\}.$$

Again, the corresponding basis functions and the geometry function are only continuous at $\xi_1 = 0.5$. The mesh of the control points as well as the weights are displayed in Table A.3.

i	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$	$w_{i,1}$	$w_{i,2}$	$w_{i,3}$
1	(-1, 0)	(-2.5, 0)	(-4, 0)	1	1	1
2	(-1, $\sqrt{2} - 1$)	(-2.5, 0.75)	(-4, 2)	$(1 + 1/\sqrt{2})/2$	1	1
3	$(1/\sqrt{2}, 1/\sqrt{2})$	(-1.5, 1.5)	(-4, 4)	$(1 + 1/\sqrt{2})/2$	1	1
4	$(1 - \sqrt{2}, 1)$	(-0.75, 2.5)	(-2, 4)	$(1 + 1/\sqrt{2})/2$	1	1
5	(0, 1)	(0, 2.5)	(0, 4)	1	1	1

Table A.3

Mesh of the control points and weights for the plate with circular hole

A.4 Fluid Analysis: Advection Dominated Advection-Diffusion Problem

The data used here also differs from [14] in order to cope better with the discontinuous Dirichlet boundary condition at $(0, 0.2)$. More precisely, we have $p_1 = p_2 = 1$ and

$$\Xi_1 = \{0, 0, 1, 1\} \text{ and} \\ \Xi_2 = \{0, 0, 0.2, 1, 1\}.$$

Here the functions are not differentiable at $\xi_2 = 0.2$.

This example has again the simple domain of the unit square and we choose again the identity as geometry function. The control points for this example are shown in Table A.4, the weights are again all 1.

i	$P_{i,1}$	$P_{i,2}$	$P_{i,3}$
1	(0, 0)	(0, 0.2)	(0, 1)
2	(1, 0)	(1, 0.2)	(1, 1)

Table A.4

Mesh of the control points and weights for the advection-diffusion problem

References

- [1] R.E. Bank and R.K. Smith: A Posteriori Error Estimates Based on Hierarchical Bases (1993), SIAM Journal on Numerical Analysis (30, No. 4), pp. 921-935
- [2] P. Bastian and G. Wittum. Adaptive multigrid methods: The UG concept (1993), in Adaptive methods - algorithms, theory and applications, Proceedings of the 9th GAMM Seminar, Notes Numer. Fluid Mech. (46), pp. 7-37, Vieweg., Braunschweig
- [3] Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes and G. Sangalli: Isogeometric Analysis: Approximation, stability and error estimates for h -refined meshes (2006), Mathematical Models and Methods in Applied Sciences (M³AS, 16 No. 7), pp. 1031-1090
- [4] D. Braess: Finite Elemente (1997) Springer-Verlag Berlin Heidelberg New York, 2nd edition
- [5] S.C. Brenner and L.R. Scott: The Mathematical Theory of Finite Element Methods (1994), Springer-Verlag New York Inc., 2nd edition
- [6] C. Carstensen: Some remarks on the history and future of averaging techniques in a posteriori finite element error analysis (2004), ZAMM (84, No.4), pp. 3-21

- [7] J.A. Cottrell, A. Reali, Y. Bazilevs and T.J.R. Hughes: Isogeometric analysis of structural vibrations (2006), *Computer Methods in Applied Mechanics and Engineering* (195), pp. 5257-5296
- [8] J. Deng, F. Chen and Y. Feng: Dimension of spline spaces over T-meshes (2006), *J. Comput. Appl. Math.* (194), pp. 267–283.
- [9] P. Deuffhard, P. Leinen and H. Yserentant: Concepts of an Adaptive Hierarchical Finite Element Code (1989), *Impact of Computing in science and engineering* (1), pp. 3-35
- [10] D. Forsey and R.H. Bartels: Hierarchical B-spline refinement(1988), *Comput. Graphics* (22, No. 4), pp. 205-212.
- [11] L.P. Franca, G. Hauke and A. Masurd: Stabilized Finite Element Methods (2003), in *Finite Element Methods: 1970's and Beyond*, Chapter 3, CIMNE, Barcelona
- [12] P.L. Gould: Introduction to linear elasticity (1983), Springer-Verlag New York, 1st edition
- [13] T.J.R. Hughes: The Finite Element Method - Linear Static and Dynamic Finite Element Analysis, Reprint, Dover Publications, Inc. Mineola, New York
- [14] T.J.R. Hughes, J.A. Cottrell and Y. Bazilevs: Isogeometric Analysis: CAD, Finite Elements, NURBS, Exact Geometry and Mesh Refinement (2005), *Computer Methods in Applied Mechanics and Engineering* (194), pp. 4135-4195
- [15] L. Piegl and W. Tiller: The NURBS Book (1995), Springer-Verlag Berlin Heidelberg, 2nd edition
- [16] T.W. Sederberg, J. Zheng, A. Bakenov and A. Nasri: T-splines and T-NURCCS (2003), *ACM Transactions on Graphics* (22, No. 3), pp. 477-484
- [17] T.W. Sederberg, D.L. Cardon, G.T. Finnigan, N.S. North, J. Zheng and T. Lyche: T-spline Simplification and Local Refinement (2004), *ACM Transactions on Graphics* (23, No. 3), pp. 276-283
- [18] A. Schmidt and K.G. Siebert: Design of Adaptive Finite Element Software - The Finite Element Toolbox ALBERTA (2005), *Lecture Notes in Computational Science and Engineering* (42), Springer
- [19] F. Weller and H. Hagen: Tensor-product spline spaces with knot segments (1995), in: M. Dæhlen et al. (eds.), *Mathematical Methods for Curves and Surfaces*, Vanderbilt University Press, Nashville, pp. 563–572.
- [20] Y. Zhang, Y. Bazilevs, S. Goswami, C.L. Bajaj and T.J.R. Hughes: Patient-Specific Vascular NURBS Modeling for Isogeometric Analysis of Blood Flow (2006), in *Proceedings of the 15th International Meshing Roundtable*, pp. 73-92, Springer Berlin Heidelberg