Article:
Chen, L. and de Borst, R. orcid.org/0000-0002-3457-3574 (2018) Adaptive refinement of hierarchical T-splines. Computer Methods in Applied Mechanics and Engineering, 337. pp. 220-245. ISSN 0045-7825

https://doi.org/10.1016/j.cma.2018.03.032

Adaptive Refinement of Hierarchical T-splines

L. Chen[a], R. de Borst[a,*]

[a] University of Sheffield, Department of Civil and Structural Engineering, Sir Frederick Mappin Building, Mappin Street, Sheffield S1 3JD, UK

Abstract

We present an adaptive local refinement technique for isogeometric analysis based on hierarchical T-splines. An element-wise point of view is adopted, which exploits Bézier extraction, and allows adaptive refinement of standard hierarchical T-splines and truncated hierarchical T-splines in a straightforward and unified manner. No explicit basis function operations are required to build the hierarchical basis function space, as only matrix manipulations are involved. This makes the efficiency superior to that of existing implementations. In particular, the implementation of truncated hierarchical T-splines requires no explicit truncation of the basis functions. In the analysis, a multi-level T-mesh is constructed by successive cell subdivisions of an initial, coarse T-mesh. An important feature is that Bézier extraction is employed to compute the refinement operator between two successive hierarchical levels, and that, at each level, Bézier extraction is applied to obtain the stiffness matrix without, initially, considering multi-level interaction. This interaction is recovered through a subdivision operator. Numerical examples are presented for validation purposes, and to assess the convergence properties.

Keywords: Hierarchical T-splines; Bézier extraction; isogeometric analysis; adaptive refinement

1. Introduction

A main advantage of isogeometric analysis is that NURBS functions commonly employed in the Computer Aided Geometric Design model, can directly be employed in the analysis model [1, 2], thus reducing the effort expended in (re)meshing, and improving, or even eliminating the error committed in the geometry description. However, the tensor product structure of NURBS prevents local mesh refinement. To obviate this drawback, various local refinement strategies have been proposed, including T-splines [3–5] and hierarchical and truncated hierarchical T-splines as further developments [6–8], LR-splines [9–12], hierarchical and truncated hierarchical B-splines [13–18], and PHT-splines [19–21]. It is further noted that adaptive splines also behold promise as an effective tool for local refinement in isogeometric analysis [6, 7, 13, 17, 22].

T-splines were introduced by Sederberg [3, 4]. They remove the rigidity of the tensor product structure of NURBS by allowing extra vertices to be inserted. Their first use in (isogeometric) analysis is in Reference [23], and the possibility to use them in existing finite element datastructures through Bézier extraction has been described in [24]. Mathematical properties of T-splines, for instance linear independence and partition-of-unity property of basis functions, are given in [25–27]. The local refinement of T-splines has been investigated in [5–7, 28, 29]. Of particular relevance for the work reported here is Reference [6, 7], in which the concepts of hierarchical and truncated hierarchical T-splines were proposed. They enable to combine the ability to locally refine hierarchical B-splines with the geometrical representation capability of T-splines.

In this contribution, we will develop the adaptive hierarchical refinement of T-splines. An element-wise point of view, enabled through Bézier extraction, will be employed for implementation purposes. A multi-level, hierarchical T-spline mesh is generated by successive cell subdivisions of an initial, coarse T-spline mesh. At each hierarchy level the element stiffness matrices are obtained by applying Bézier extraction without, initially, considering multi-level interaction. This interaction is enforced through the introduction of a subdivision operator. Two cases are considered:

[*] Corresponding author
Email address: r.deborst@sheffield.ac.uk (R. de Borst)

standard hierarchical T-spline bases and truncated hierarchical T-spline bases. It is noted that the hierarchical bases are not implemented explicitly in the formulation, but implicitly through matrix multiplications.

The paper is structured as follows. Section 2 gives a succinct summary of T-splines and Bézier extraction, including the construction of the Bézier extraction operator. Some basic notions of hierarchical T-spline basis functions are discussed in Section 3. Section 4 illustrates the use of Bézier extraction for hierarchical T-splines, and Section 5 provides information on adaptive hierarchical refinement. The paper is completed with numerical examples in Section 6 to validate the approach, and with some concluding remarks.

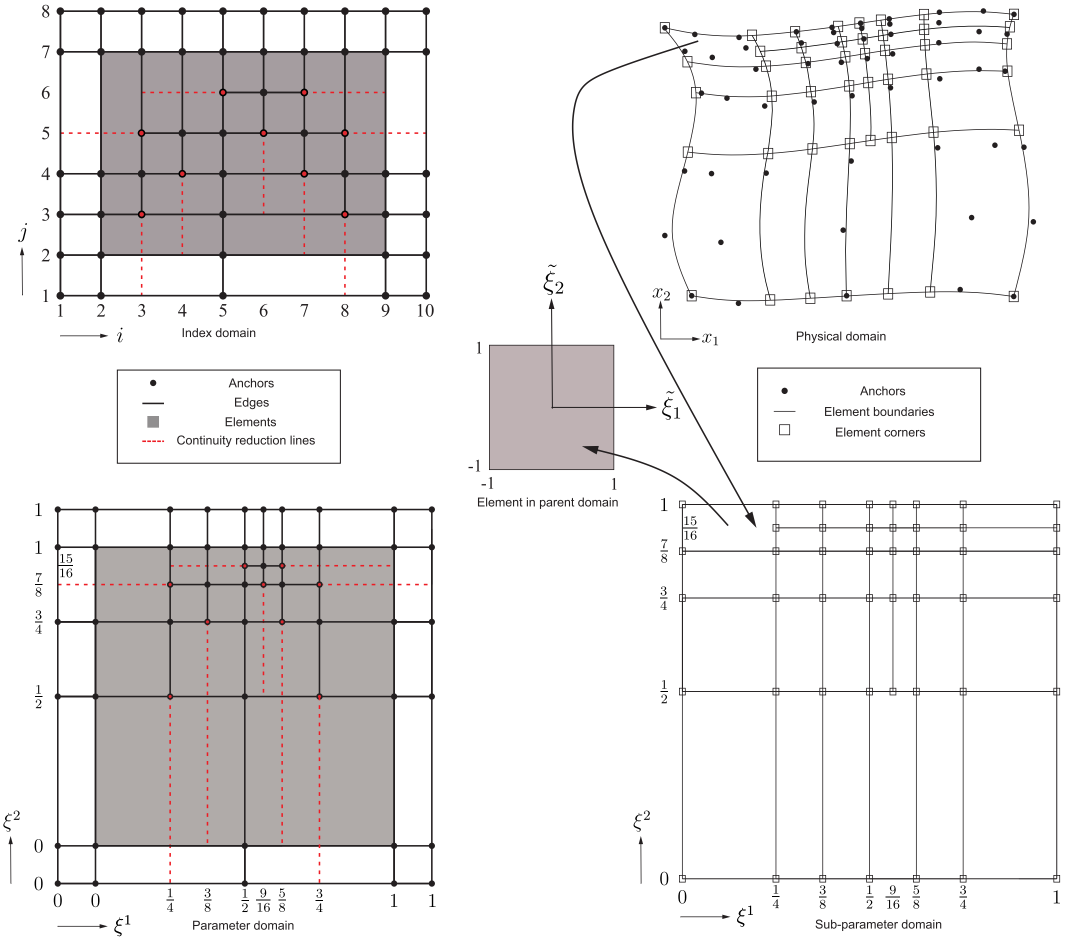


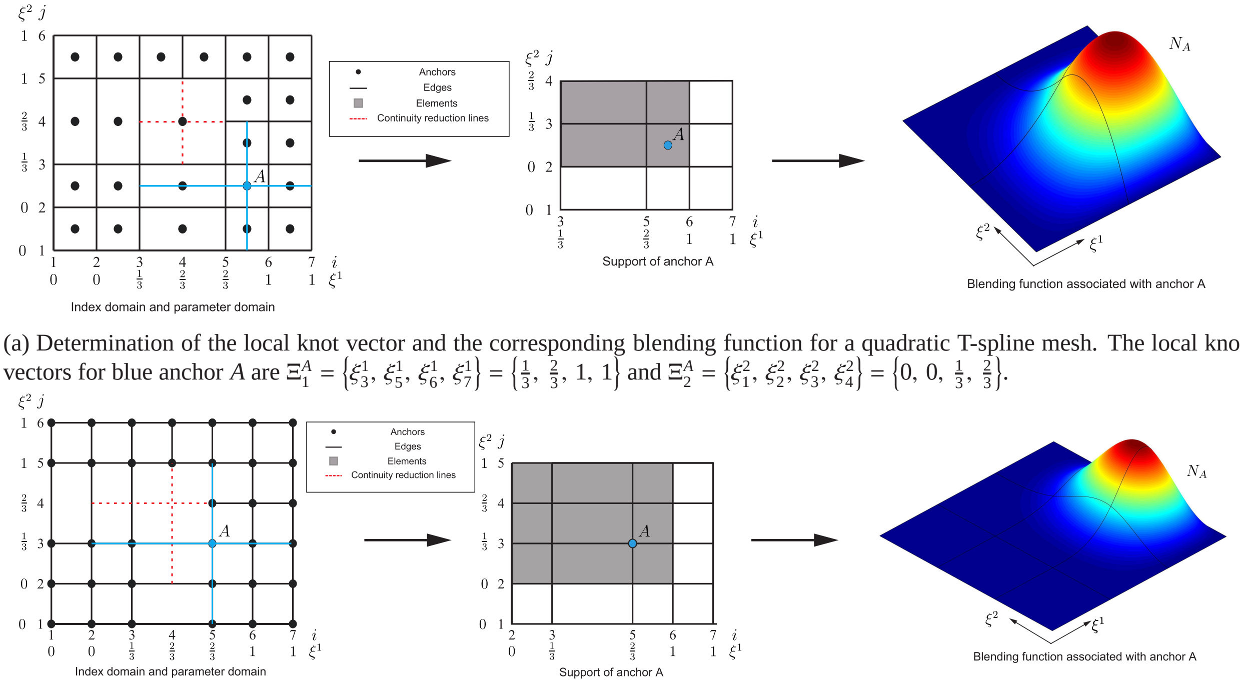
Figure 1: Example of a cubic T-spline mesh. The object is given in the index domain $(i,\ j)$, in the physical domain $(x_1,\ x_2)$, and in the parameter domain $\left(\xi^1,\ \xi^2\right)$. The element is also shown in the parent domain.

2. T-spline and Bézier extraction

In this Section, we will first give a concise overview of T-splines and Bézier extraction, please see [23, 24] for a more in-depth discussion.

2.1. Fundamentals of T-splines

T-splines are constructed on a T-mesh. For two dimensional objects, the T-mesh is a mesh of quadrilateral elements, which allows T-junctions. An example of a cubic T-spline mesh is given in Figure 1. In the figure, the index domain $(i,\ j)$, the physical domain $(x_1,\ x_2)$, the parameter and sub-parameter domain $\left(\xi^1,\ \xi^2\right)$ of T-splines are



ξ^2 j
i ξ^1
A
Anchors
Edges
Elements
Continuity reduction lines
Index domain and parameter domain
Support of anchor A
N_A
Blending function associated with anchor A

(a) Determination of the local knot vector and the corresponding blending function for a quadratic T-spline mesh. The local knot vectors for blue anchor A are $\Xi_1^A = \left\{\xi_3^1,\, \xi_5^1,\, \xi_6^1,\, \xi_7^1\right\} = \left\{\frac{1}{3},\, \frac{2}{3},\, 1,\, 1\right\}$ and $\Xi_2^A = \left\{\xi_1^2,\, \xi_2^2,\, \xi_3^2,\, \xi_4^2\right\} = \left\{0,\, 0,\, \frac{1}{3},\, \frac{2}{3}\right\}$.

(b) Determination of the local knot vector and the corresponding blending function for a cubic T-spline mesh. The local knot vectors for blue anchor A are $\Xi_1^A = \left\{\xi_2^1,\, \xi_3^1,\, \xi_5^1,\, \xi_6^1,\, \xi_7^1\right\} = \left\{0,\, \frac{1}{3},\, \frac{2}{3},\, 1,\, 1\right\}$ and $\Xi_2^A = \left\{\xi_1^2,\, \xi_2^2,\, \xi_3^2,\, \xi_4^2,\, \xi_5^2\right\} = \left\{0,\, 0,\, \frac{1}{3},\, \frac{2}{3},\, 1\right\}$.

Figure 2: Determination of the local knot vector and the construction of blending functions for quadratic and cubic T-spline meshes.

shown. To carry out Gaussian integration, the element e must be mapped from the physical domain onto the parameter domain, and then onto the parent domain $\tilde{\xi}^l \in [-1,\, 1]$, see Figure 1.

In a T-spline mesh, elements are defined by the edges of T-spline mesh and continuity reduction lines [26], see the areas that are shaded gray in Figures 1 and 2. Anchors are prescribed in the index domain and in the parameter domain [23], and a multivariate blending function is attached to each of them. In Figure 1, the anchors are placed at each vertex of a cubic T-mesh. To obtain the blending function associated with an anchor, a local knot vector $\left(\boldsymbol{\Xi}^i\right)_{i=0,\cdots,n-1}$ has to be defined, with n denoting the number of anchors. In Reference [26] the construction of local knot vectors for T-meshes of even and of odd degrees has been described in detail. Figure 2 gives an example of the determination of local knot vectors. On the basis of local knot vectors, we obtain the blending functions associated with an anchor [24], see Figure 2.

2.2. *Bézier extraction fundamentals*

The blending function N_A is defined over entire support of anchor A, see Figures 1 and 2. It is cumbersome to directly incorporate a blending function in a standard finite element code. However, Bézier extraction overcomes this by representing blending functions as element-wise Bernstein shape functions [24], and this approach will be adopted herein. We suppose that the domain can be decomposed into E elements with n anchors. The local knot vectors of anchor i are Ξ_1^i and Ξ_2^i. Then, the blending function N_i of anchor i over element e can be expressed as

$$N_i^e\left(\xi\right) = \left[\mathbf{C}_i^e\right]^{\mathrm{T}} \mathbf{B}^e\left(\xi\right) \tag{1}$$

where $\mathbf{B}^e\left(\xi\right)$ contains (element-local) Bernstein polynomials with dimension $(p+1)^2 \times 1$ [26]. Here, we consider T-splines with same polynomial degree, p, in the ξ^1 and the ξ^2 parametric direction.

$\mathbf{C}_i^e$ is the Bézier extraction operator of anchor i over element e, which is determined by the tensor product [26]:

$$\mathbf{C}_i^e = \mathbf{C}_i^{e2} \otimes \mathbf{C}_i^{e1} \tag{2}$$

where $\mathbf{C}_i^{e1}$ and $\mathbf{C}_i^{e2}$ are univariate Bézier extraction operators of anchor i over element e in the ξ^1 and the ξ^2 parametric direction, respectively.

We take anchor A in Figure 2(a) as an example to illustrate the calculation of Bézier extraction operator $\mathbf{C}_i^e$. The local knot vectors of anchor A are $\Xi_1^A = \left\{\xi_3^1, \xi_5^1, \xi_6^1, \xi_7^1\right\} = \left\{\frac{1}{3}, \frac{2}{3}, 1, 1\right\}$ and $\Xi_2^A = \left\{\xi_1^2, \xi_2^2, \xi_3^2, \xi_4^2\right\} = \left\{0, 0, \frac{1}{3}, \frac{2}{3}\right\}$, respectively. We evaluate the Bézier extraction operator of anchor A over element b, which is bounded in the parameter domain by $\left[\xi_5^1, \xi_6^1\right] \times \left[\xi_2^2, \xi_3^2\right]$ and in the sub-parameter domain by $\left[\frac{2}{3}, 1\right] \times \left[0, \frac{1}{3}\right]$. Figure 3 shows the blending function N_A^b for each parametric direction. The part of N_A^b in the ξ^l parametric direction, i.e. N_A^{bl}, is plotted as a solid line over element b. Now, we express the blending function N_A^{bl} as a linear combination of Bernstein polynomials $\mathbf{B}^{bl}$ over element b:

$$N_A^{b1} = [C_A^b]^1 \mathbf{B}^{b1} = \begin{bmatrix} C_{A1}^{b1} & C_{A2}^{b1} & C_{A3}^{b1} \end{bmatrix} \begin{pmatrix} B_1^{b1} \\ B_2^{b1} \\ B_3^{b1} \end{pmatrix} = \begin{bmatrix} \frac{1}{2} & 1 & 0 \end{bmatrix} \begin{pmatrix} B_1^{b1} \\ B_2^{b1} \\ B_3^{b1} \end{pmatrix} \tag{3}$$

$$N_A^{b2} = [C_A^b]^2 \mathbf{B}^{b2} = \begin{bmatrix} C_{A1}^{b2} & C_{A2}^{b2} & C_{A3}^{b2} \end{bmatrix} \begin{pmatrix} B_1^{b2} \\ B_2^{b2} \\ B_3^{b2} \end{pmatrix} = \begin{bmatrix} 0 & 1 & \frac{1}{2} \end{bmatrix} \begin{pmatrix} B_1^{b2} \\ B_2^{b2} \\ B_3^{b2} \end{pmatrix} \tag{4}$$

Employing Equation (2) and considering Equations (3) and (4), we obtain the Bézier extraction operator of anchor A over element b, as follows:

$$\mathbf{C}_A^b = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{2} & 1 & 0 & \frac{1}{4} & \frac{1}{2} & 0 \end{bmatrix} \tag{5}$$

When we subsequently apply Equation (2) to all elements, the Bézier extraction operator of anchor i is obtained:

$$\mathbf{C}_i = \begin{bmatrix} \mathbf{C}_i^1 \\ \vdots \\ \mathbf{C}_i^E \end{bmatrix} \tag{6}$$

The dimension of $\mathbf{C}_i$ is $E\,(p+1)^2 \times 1$, where E is the total number of elements.

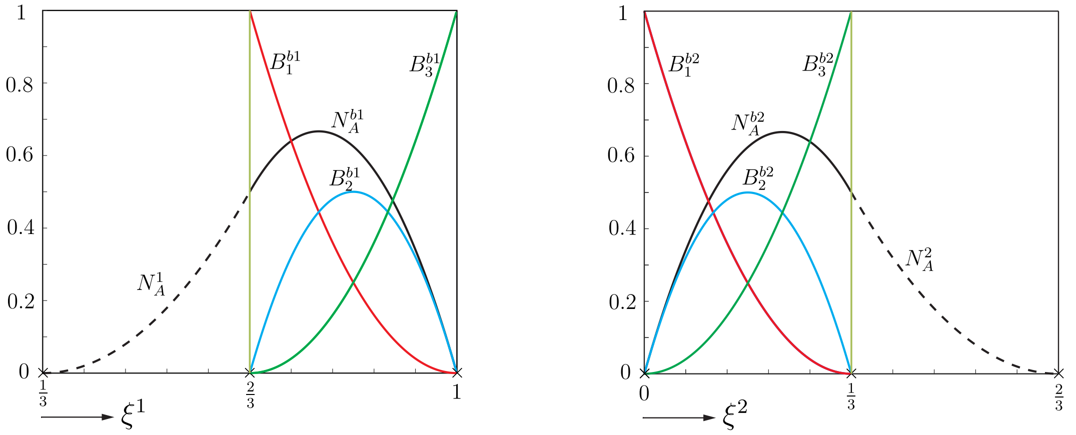


Figure 3: The blending function N_A^{bl} of anchor A and the Bernstein polynomials over element b in the ξ^l parametric direction.

When we consider Equation (6) for all anchors, we can express the n blending functions as a linear combination of Bernstein basis functions:

$$\mathbf{N}(\xi) = \mathbf{C}\mathbf{B}(\xi) = \begin{bmatrix} N_1(\xi) \\ \vdots \\ N_n(\xi) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{C}_n^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{B}^1(\xi) \\ \vdots \\ \mathbf{B}^E(\xi) \end{bmatrix} \tag{7}$$

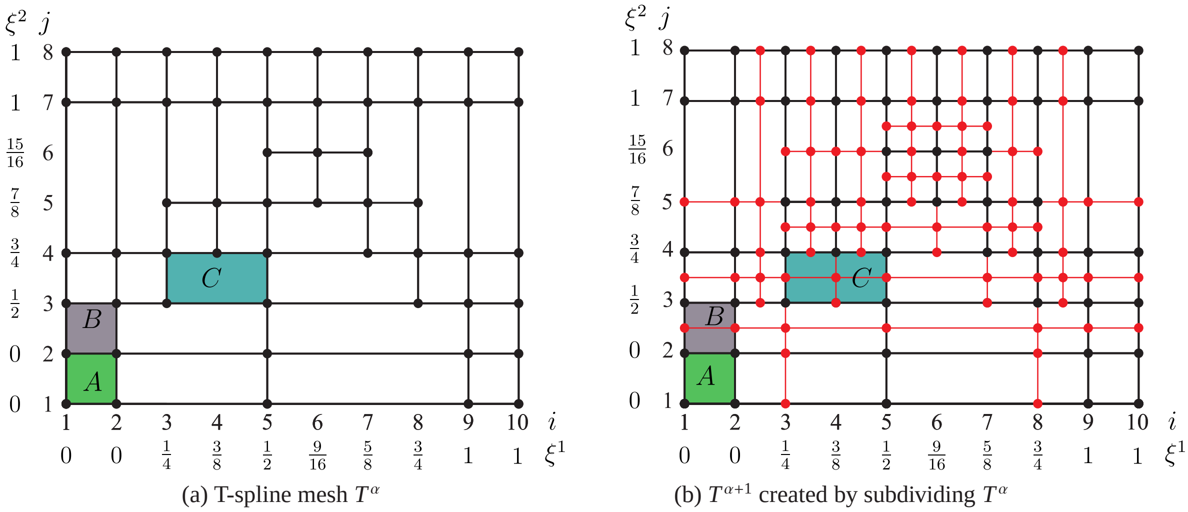
where $\mathbf{C}$ has the dimension $n \times E(p+1)^2$ and $\mathbf{B}(\xi)$ is of $E(p+1)^2 \times 1$. The blending functions with support over element e are then expressed as:

$$\mathbf{N}_e(\xi) = \mathbf{C}_e \mathbf{B}_e(\xi) \tag{8}$$

with $\mathbf{C}_e$ element Bézier extraction operator.

3. Hierarchical T-splines

Hierarchical T-splines were introduced in Reference [6] and allow for local refinement of a given, normally coarse T-spline mesh. The basic idea of hierarchical T-splines is to locally enrich the approximation space by replacing selected coarse grid T-splines by fine grid T-splines. We now give a concise outline of the mechanism how to construct hierarchical T-splines.



(a) T-spline mesh T^α

(b) $T^{\alpha+1}$ created by subdividing T^α

Figure 4: Construction of T-spline mesh $T^{\alpha+1}$ from T^α. The anchors are indicated by circular dots. Black denotes the mesh and the anchors of T-spline mesh T^α, and red stands for the mesh and the anchors generated for $T^{\alpha+1}$.

3.1. *Nested spaces and domains*

A hierarchical T-spline space is constructed from a finite sequence of P nested T-spline spaces $\left(\mathcal{T}^l\right)_{l=0,\cdots,P-1}$ bounded by P open sets $\left(\Omega^l\right)_{l=0,\cdots,P-1}$. The nested nature of T-spline space defines the nested domains for hierarchy:

$$\mathcal{T}^0 \subset \mathcal{T}^1 \subset \cdots \subset \mathcal{T}^{P-1} \qquad\qquad \Omega^{P-1} \subseteq \Omega^{P-2} \subseteq \cdots \subseteq \Omega^0 \tag{9}$$

The sequence of P T-spline meshes is built by subdividing each effective rectangular cell in T^α into two or four congruent cells such that $\mathcal{T}^\alpha \subset \mathcal{T}^{\alpha+1}$, $\alpha = 0, \cdots, P-2$, where the term 'effective rectangular cell' refers to a cell with non-zero parametric length in at least one parametric direction. Figure 4 illustrates the algorithm to generate the T-spline mesh $T^{\alpha+1}$ from T^α. Cell A has a zero parametric length in both directions, which results in no subdivision in A. Cell B only has a non-zero length in the ξ^2 parametric direction. It is divided into two congruent cells in $T^{\alpha+1}$. For cell C, the parametric length is non-zero in both directions, which leads to four congruent cells in $T^{\alpha+1}$.

3.2. *Hierarchical T-spline bases*

The algorithm of [6] has been adopted for the construction of hierarchical T-spline bases, but we rephrase the algorithm below. Defining τ as a T-spline basis function and by $\mathcal{T}$ we denote the T-spline basis function space [13]. The hierarchical T-spline bases $\mathcal{H}$ is recursively built as follows:

(1) Initialize: $\mathcal{H}^0 = \left\{\tau \in \mathcal{T}^0 : \operatorname{supp} \tau \neq \varnothing\right\}$.

(2) Construct: $\mathcal{H}^{\alpha+1}$ from $\mathcal{H}^{\alpha}$ in a recursive manner: $\mathcal{H}^{\alpha+1} = \mathcal{H}^{\alpha+1}_{\text{coarse}} \cup \mathcal{H}^{\alpha+1}_{\text{fine}}$, $\alpha = 0, \cdots, P-2$ where $\mathcal{H}^{\alpha+1}_{\text{coarse}} = \left\{\tau \in \mathcal{H}^{\alpha} : \operatorname{supp} \tau \nsubseteq \Omega^{\alpha+1}\right\}$; $\quad \mathcal{H}^{\alpha+1}_{\text{fine}} = \left\{\tau \in \mathcal{T}^{\alpha+1} : \operatorname{supp} \tau \subseteq \Omega^{\alpha+1}\right\}$.

(3) Set $\mathcal{H} = \mathcal{H}^{P-1}$.

Using linear combinations between basis functions on hierarchy level α and $\alpha + 1$, we can obtain a truncated hierarchical basis function space [7, 14]:

(1) Initialize: $\mathcal{H}^0 = \left\{\tau \in \mathcal{T}^0 : \operatorname{supp} \tau \neq \varnothing\right\}$.

(2) Construct: $\mathcal{H}^{\alpha+1}$ from $\mathcal{H}^{\alpha}$ in a recursive manner: $\mathcal{H}^{\alpha+1} = \mathcal{H}^{\alpha+1}_{\text{coarse}} \cup \mathcal{H}^{\alpha+1}_{\text{fine}}$, $\alpha = 0, \cdots, P-2$ where $\mathcal{H}^{\alpha+1}_{\text{coarse}} = \left\{\tau \in \mathcal{H}^{\alpha} \wedge \operatorname{supp} \tau \nsubseteq \Omega^{\alpha+1}\right\}$; $\quad \mathcal{H}^{\alpha+1}_{\text{fine}} = \left\{\tau \in \mathcal{T}^{\alpha+1} : \operatorname{supp} \tau \subseteq \Omega^{\alpha+1}\right\}$.

(3) Set $\mathcal{H} = \mathcal{H}^{P-1}$.

Figure 5 illustrates the hierarchical and truncated hierarchical basis function spaces for the univariate case, while Figures 6 and 7 do so for the bivariate case. The mathematical properties of hierarchical and truncated hierarchical basis functions, e.g. nested nature, linear independence and partition of unity property have been discussed in [6, 7].

4. Bézier extraction of hierarchical T-splines

We will now extend the Bézier extraction framework to allow for the implementation of hierarchical T-splines. This procedure is similar to that used for hierarchical B-splines [17]. As stated, an element-wise point of view is adopted, which conforms ideally to Bézier extraction framework. The hierarchical T-spline basis functions are defined over multiple hierarchy levels. Strong boundary condition is imposed over different hierarchy levels [13]. This results in a nested hierarchical element structure.

4.1. Data structure

The element-wise implementation of hierarchical T-splines is a natural choice in adaptive finite element analysis. The hierarchical T-spline bases consist of T-splines over multiple hierarchical levels with same polynomial degree. Below, we will outline the data structure of multi-level hierarchical bases.

4.1.1. Multi-level mesh and Bézier extraction operator

We first construct a hierarchy of P levels. The T-spline basis functions at each hierarchy level are defined over local knot vector set Ξ_i ($i = 0, 1, ... P-1$). Ξ_i is defined as $\Xi_i = \left\{\Xi_i^j\right\}_{j=0}^{n_i-1}$, in which n_i denotes the number of anchors at level i. Ξ_i is obtained by successive cell subdivision within Ω_d, starting from initial local knot vector Ξ_0, where Ω_d denotes the parameter domain. The algorithm is visualised in Figure 4. In this process, the nested parameter domains $\Omega_d^{i+1} \subset \Omega_d^i$ and the nested local knot vectors $\Xi_i \subset \Xi_{i+1}$ are obtained, see Figure 4. Each knot vector Ξ_i defines a set of T-spline basis functions $\mathbf{N}^i = \left\{N_j^i\right\}_{j=1}^{n_i}$, which in turn forms a nested T-spline approximation space $\mathcal{T}^i$, see Figure 6. With Ξ_i and Ω_d^i, the Bézier extraction operator $\mathbf{C}^i$ can be obtained for each anchor at level i, see Section 2.2. Furthermore, T-spline basis functions at level i can be defined in terms of elements at level $i+1$ by the Bernstein polynomials $\mathbf{B}^{i+1}(\xi)$:

$$\mathbf{N}^i = \mathbf{C}_R^i \mathbf{B}^{i+1}(\xi) \tag{10}$$

where $\mathbf{C}_R^i$ denotes Bézier extraction operator of each anchor at level i over elements at level $i+1$.

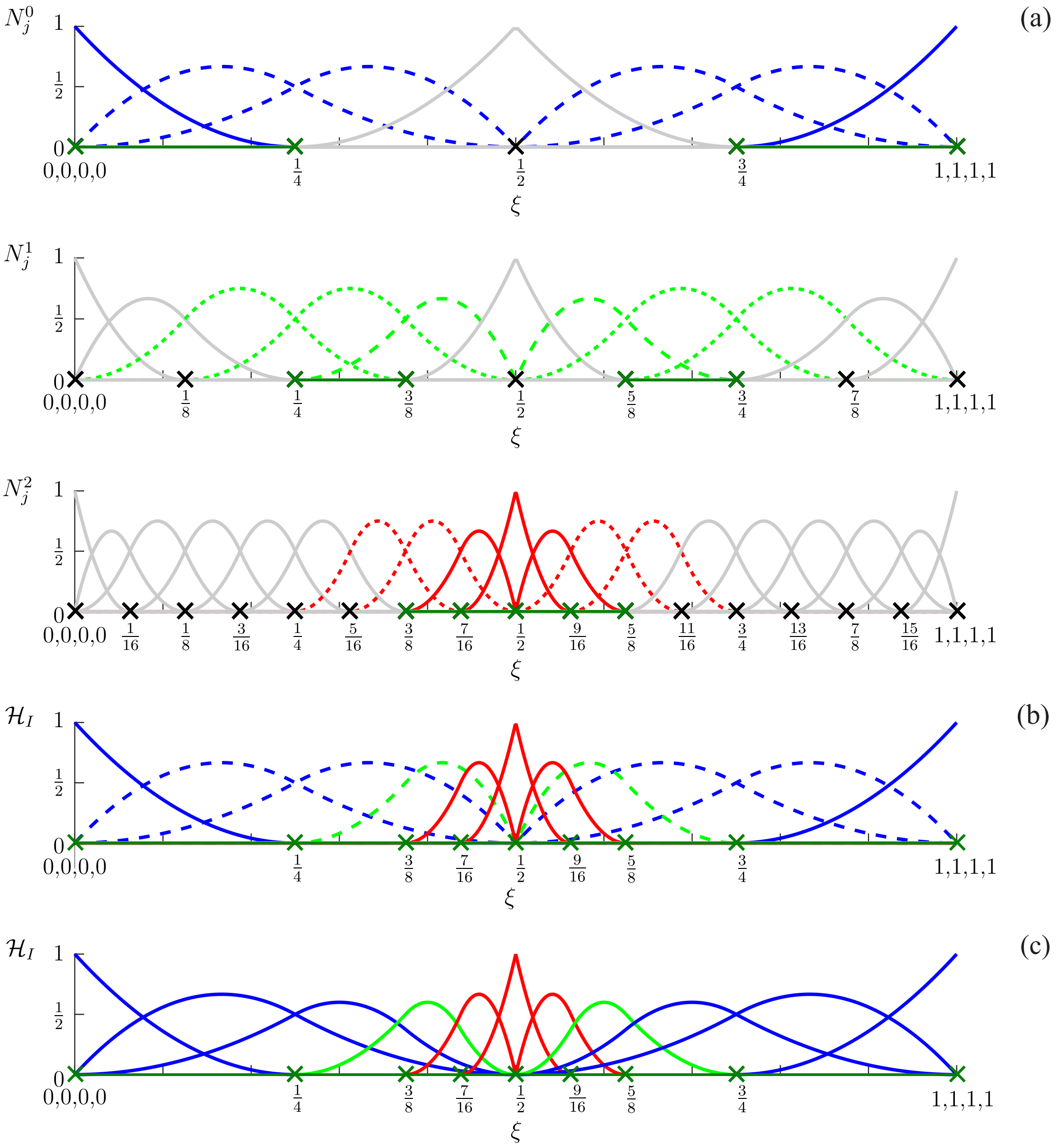


Figure 5: Definition of basis function space; (a) illustration of basis function sets $\mathcal{A}^l$, $\mathcal{A}^l_-$ and $\mathcal{A}^l_+$; (b) final hierarchical basis functions; (c) final truncated hierarchical basis functions.

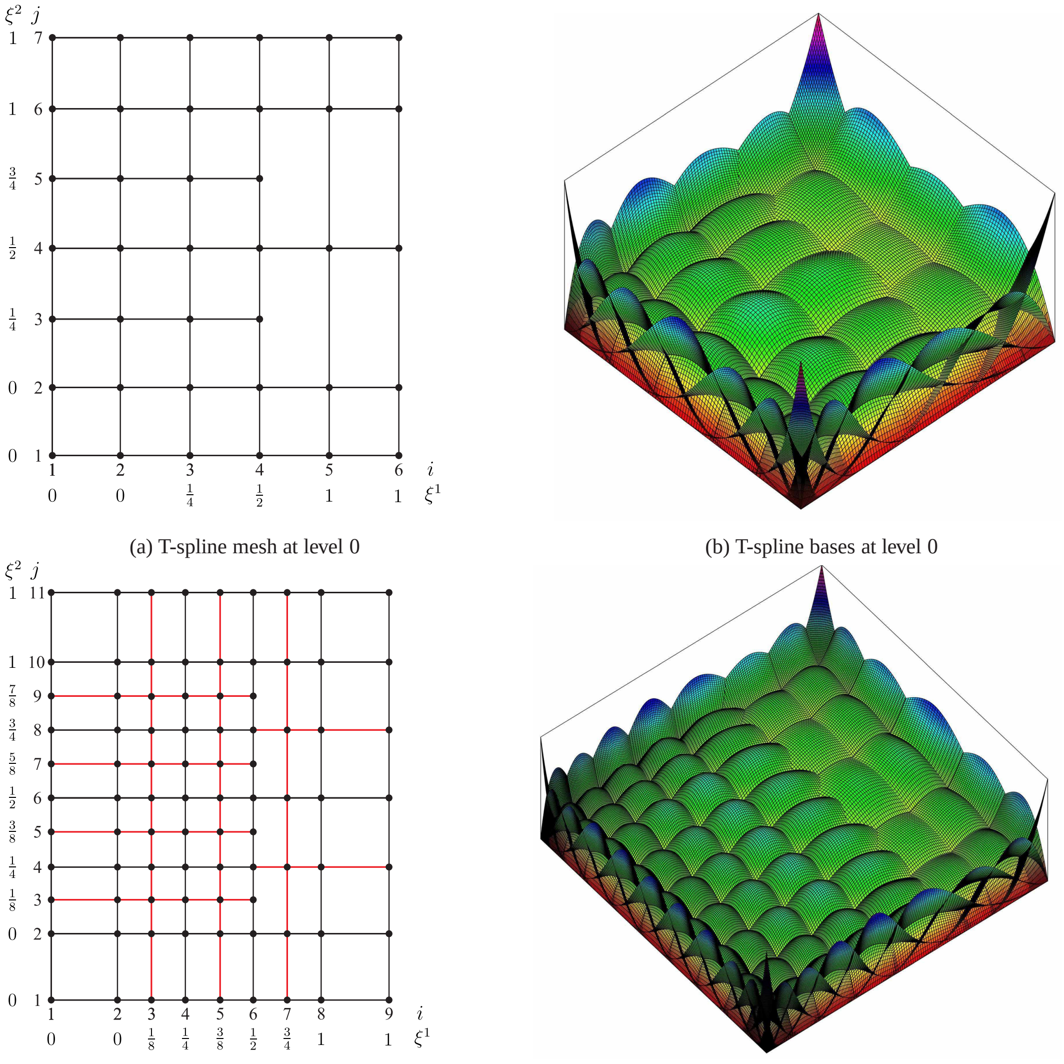
4.1.2. Subdivision operator and control point

Due to the nested nature of $\mathcal{T}^i$, the T-spline basis functions at hierarchy level i can be described by the T-spline basis functions at hierarchy level j:

$$\mathbf{N}^i = \mathbf{S}^{i,j}\mathbf{N}^j = \prod_{l=i}^{j-1} \mathbf{S}^{l,l+1}\mathbf{N}^{l+1} \tag{11}$$

where $\mathbf{S}^{l,l+1}$ is the subdivision or refinement operator [26]. For the NURBS basis functions, the subdivision operator $\mathbf{S}^{l,l+1}$ can be directly obtained from the linear relation between basis functions from two hierarchy levels [15, 18]. It is noted that $\mathbf{S}^{l,l+1}$ is a matrix with a high degree of sparsity. Using Equations (7) and (10), $\mathbf{S}^{l,l+1}$ can now be computed:

$$\mathbf{N}^l = \mathbf{C}_R^l \mathbf{B}^{l+1}(\xi) = \mathbf{S}^{l,l+1}\mathbf{C}^{l+1}\mathbf{B}^{l+1}(\xi) \tag{12}$$



(a) T-spline mesh at level 0

(b) T-spline bases at level 0

(c) T-spline mesh at level 1

(d) T-spline bases at level 1

Figure 6: T-spline meshes and bases at hierarchy level 0 and 1

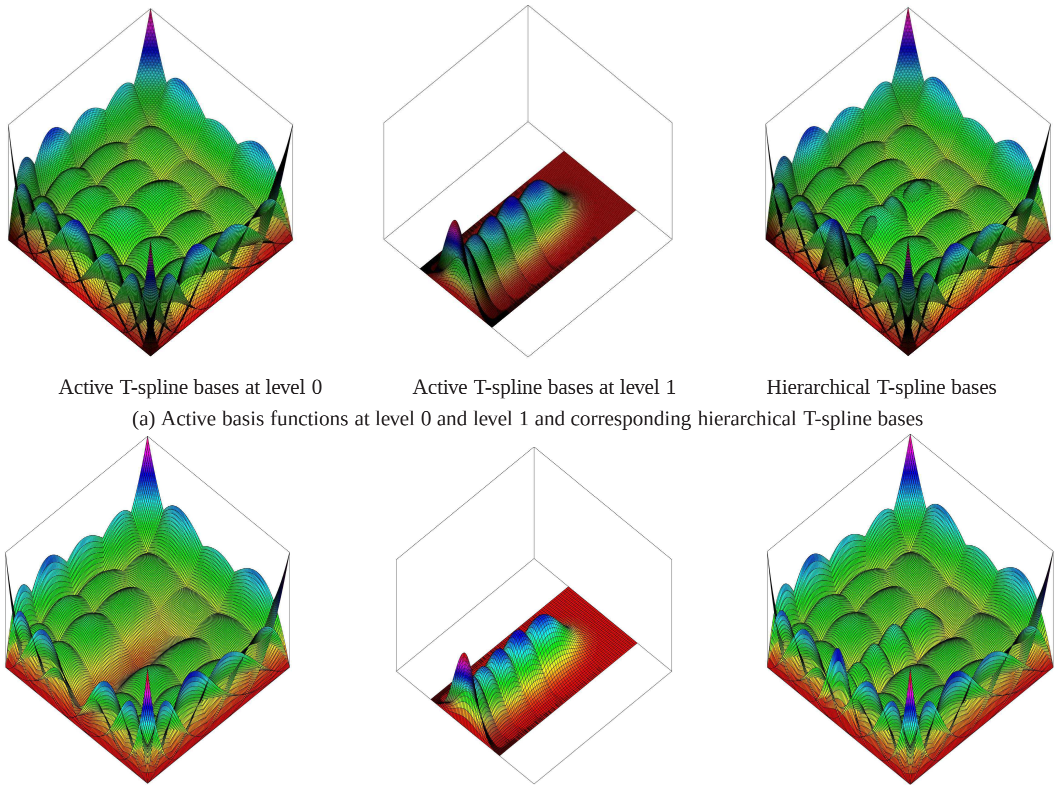
where $\mathbf{C}_R^l$ denotes Bézier extraction operator of each anchor at level l over elements at level $l+1$, and $\mathbf{C}^{l+1}$ represents Bézier extraction operator of each anchor at level $l+1$.

Expanding Equation (12) in a vector form, it obtains

$$\begin{bmatrix} [\mathbf{C}_{R0}^l]^{\mathrm{T}} \\ \vdots \\ [\mathbf{C}_{Rn_l}^l]^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} [\mathbf{S}_1^{l,l+1}]^{\mathrm{T}} \\ \vdots \\ [\mathbf{S}_{n_l-1}^{l,l+1}]^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} [\mathbf{C}_1^{l+1}]^{\mathrm{T}} \\ \vdots \\ [\mathbf{C}_{n_{l+1}-1}^{l+1}]^{\mathrm{T}} \end{bmatrix} \tag{13}$$

with n_l and n_{l+1} the number of anchors at level l and $l+1$, respectively. $\mathbf{C}_{Ri}^l$ denotes Bézier extraction operator of anchor i at level l over elements at level $l+1$, whose dimension is $E_{l+1}\,(p+1)^2 \times 1$. $\mathbf{C}_i^{l+1}$ represents Bézier extraction operator of anchor i at level $l+1$ with the dimension of $E_{l+1}\,(p+1)^2 \times 1$. E_{l+1} is the number of elements at level $l+1$. The row values of $\mathbf{S}^{l,l+1}$ are obtained from:

$$\mathbf{C}_{Ri}^l = [\mathbf{C}^{l+1}]^{\mathrm{T}}\mathbf{S}_i^{l,l+1} \qquad \text{for} \qquad i = 0, \cdots, n_l - 1 \tag{14}$$


Active T-spline bases at level 0 Active T-spline bases at level 1 Hierarchical T-spline bases

(a) Active basis functions at level 0 and level 1 and corresponding hierarchical T-spline bases

Active T-spline bases at level 0 Active T-spline bases at level 1 Truncated hierarchical T-spline bases

(b) Active basis functions at level 0 (already truncated) and level 1 and corresponding truncated hierarchical T-spline bases

Figure 7: Hierarchical and truncated hierarchical T-spline bases

With the subdivision operator, the coordinates and weights of anchors in a refined T-spline mesh at hierarchy level i can be determined recursively [26]:

$$\mathbf{P}_w^i = [\mathbf{S}^{i,0}]^{\mathrm{T}}\mathbf{P}_w^0 = \left(\prod_{l=0}^{i-1}\mathbf{S}^{l,l+1}\right)^{\mathrm{T}}\mathbf{P}_w^0 \tag{15}$$

where $\mathbf{P}_w^i$ are the weighted control points at level i. Each weighted control point is defined as $\mathbf{P}_{w,j}^i = \left(w_j^i x_{1j}^i,\ w_j^i x_{2j}^i,\ w_j^i\right)$.

With the subdivision operator defined in Equation (12), and the weight w from Equation (15), the T-spline bases can also be written in a rational form [17]:

$$R_I^l(\xi) = \frac{w_I^l N_I^l(\xi)}{W^l(\xi)} = \frac{w_I^l \sum_J S_{IJ}^{l,l+1} N_J^{l+1}(\xi)}{W^l(\xi)} = w_I^l \sum_J \frac{S_{IJ}^{l,l+1}}{w_J^{l+1}} R_J^{l+1}(\xi) \tag{16}$$

Using the rational T-spline bases, the subdivision operator must be modified as below:

$$\bar{S}_{IJ}^{l,l+1} = \frac{w_I^l}{w_J^{l+1}} S_{IJ}^{l,l+1} \tag{17}$$

to replace the standard subdivision operator $\mathbf{S}^{l,l+1}$.

4.2. Multi-level implementation of hierarchical T-splines

We next take univariate bases to illustrate the construction of hierarchical basis functions. Due to the nested structure of T-spline basis function space, the multi-variate case can be derived straightforwardly. To construct the hierarchical basis function space $\mathcal{H}$, the active elements and basis functions in the multi-level hierarchy must be identified. The active element is chosen by a certain marking criterion, for example a posteriori error estimator [16]. No overlap or gap may exist between the active elements from different hierarchy levels, Figure 5(a). In the figure, the parameter domain of active elements is plotted in green, where the parameter domain of active elements is defined as the union of three parameter domains associated with hierarchy level l:

$$\Omega_d = \Omega_d^{l-} + E_A^l + \Omega_d^{l+} = \bigcup_{i=0}^{l-1} E_A^i + E_A^l + \bigcup_{i=l+1}^{P-1} E_A^i \qquad \text{with} \qquad E_A^i = \bigcup_e \Omega_d^{e,i} \tag{18}$$

where P is the number of hierarchy levels: E_A^i represents the parameter domain of active elements at hierarchy level i, $\Omega_d^{e,i}$ denotes the parameter domain of element e at hierarchy level i; Ω_d^{l-} is the parameter domain of active elements at coarser hierarchy levels; and Ω_d^{l+} represents the parameter domain of active elements at finer hierarchy levels.

Herein, an element-based selection approach is employed to define the hierarchical basis function space $\mathcal{H}$ [17]. On the basis of Ω_d, Ω_d^{l+} and Ω_d^{l-}, one defines three sets of basis function spaces:

$$\begin{cases} \mathcal{A}^l = \left\{ N_j^l \in \mathcal{T}^l : \operatorname{supp} N_j^l \bigcap E_A^l \neq \emptyset \right\} \\ \mathcal{A}_+ = \displaystyle\bigcup_{l=0}^{P-1} \mathcal{A}_+^l \quad \text{with} \quad \mathcal{A}_+^l = \left\{ N_j^l \in \mathcal{A}^l : \operatorname{supp} N_j^l \bigcap \Omega_d^{l+} \neq \emptyset \right\} \\ \mathcal{A}_- = \displaystyle\bigcup_{l=0}^{P-1} \mathcal{A}_-^l \quad \text{with} \quad \mathcal{A}_-^l = \left\{ N_j^l \in \mathcal{A}^l : \operatorname{supp} N_j^l \bigcap \Omega_d^{l-} \neq \emptyset \right\} \end{cases} \tag{19}$$

$\mathcal{A}^l$ denotes the union of basis functions with support over active elements at hierarchy level l, Figure 5(a). $\mathcal{A}_+^l$ denotes the basis function set in $\mathcal{A}^l$ with support over active elements at finer hierarchy levels, plotted as dashed lines, see Figure 5(a). $\mathcal{A}_-^l$ represents basis functions in $\mathcal{A}^l$ which have a support over active elements at coarser hierarchy levels, as indicated by dotted lines in Figure 5(a). Finally, the space of hierarchical basis functions $\mathcal{H}$ is defined as:

$$\mathcal{H} = \bigcup_{l=0}^{P-1} \mathcal{A}_a^l \qquad \text{with} \qquad \mathcal{A}_a^l = \mathcal{A}^l \setminus \mathcal{A}_-^l \tag{20}$$

where "\" is the logic NOT, $\mathcal{A}_a^l$ denotes active basis functions at hierarchy level l, see Figure 5(b), and $\mathcal{H}$ denotes standard hierarchical basis function space [6].

From the linear combination between basis functions at hierarchy level l and $l+1$, a space of truncated hierarchical basis functions is obtained [7], see Figure 5(c):

$$\mathcal{H}_T = \bigcup_{l=0}^{P-1} \mathcal{A}_{T,a}^l \qquad \text{with} \qquad \mathcal{A}_{T,a}^l = \left\{ \tau_i^l \in \mathcal{A}_a^l : \operatorname{supp} \tau_i^l \nsubseteq E_A^{l+1} \right\} \tag{21}$$

where τ_i^l is defined as $\tau_i^l = \left\{ \tau_i^l \in \mathcal{T}^l : \tau_i^l = \sum S_{ij}^{l,l+1} N_j^{l+1} \right\}$, see Equation (11).

Considering the active elements and basis functions, one can implement hierarchical basis functions to obtain the stiffness matrix in a multi-level, adaptive manner. First, using Bézier extraction, the stiffness matrix of active elements at each hierarchy level is obtained, without consideration of possible interaction of multi-level basis functions. Assembling the stiffness matrix at each hierarchy level, a global system of equations is obtained:

$$\mathbf{KU} = \mathbf{F} \tag{22}$$

where $\mathbf{U}$ contains nodal degrees of freedom at each hierarchy level. The force vector is given by $\mathbf{F}$ and $\mathbf{K}$ is the stiffness matrix, with the submatrices $\mathbf{K}^i$ along the diagonal, $\mathbf{K}^i$ being the stiffness matrix of active elements at hierarchy level i, a square sparse matrix of size $2n_c^i \times 2n_c^i$. The number of control points at hierarchy level i is given by n_c^i. It is noted that the submatrices $\mathbf{K}^i$ are also highly sparse.

To enforce the interaction between multi-level hierarchical bases in Equation (22), a hierarchical subdivision operator $\mathbf{M}_h$ is introduced, resulting in the following hierarchical system of equations [30]:

$$\mathbf{K}_h\mathbf{U}_h = \mathbf{F}_h \qquad \text{with} \qquad \mathbf{K}_h = \mathbf{M}_h\mathbf{K}\mathbf{M}_h^{\mathrm{T}} \quad \text{and} \quad \mathbf{F}_h = \mathbf{M}_h\mathbf{F} \tag{23}$$

Again, $\mathbf{K}_h$ is a very sparse matrix. The hierarchical subdivision operator $\mathbf{M}_h$ is defined as:

$$\mathbf{M}_h = \begin{bmatrix} \mathbf{I}^0 & \hat{\mathbf{M}}^{0,1} & \hat{\mathbf{M}}^{0,2} & \dots & \hat{\mathbf{M}}^{0,P-1} \\ & \mathbf{I}^1 & \hat{\mathbf{M}}^{1,2} & \dots & \hat{\mathbf{M}}^{1,P-1} \\ & & \mathbf{I}^2 & \dots & \hat{\mathbf{M}}^{2,P-1} \\ & \mathbf{0} & & \ddots & \\ & & & & \mathbf{I}^{P-1} \end{bmatrix} \tag{24}$$

in which

$$I_{IJ}^l = \begin{cases} 1 & \text{for} \quad I = J \quad \text{and} \quad N_I^l \in \mathcal{A}_a^l \\ 0 & \text{else} \end{cases} \tag{25}$$

The matrix $\hat{\mathbf{M}}^{l,k}$ in Equation (24) is defined as follows:

$$\begin{aligned} &\text{Standard hierarchical bases:} \qquad && \hat{M}_{IJ}^{l,k} = \begin{cases} S_{IJ}^{l,k} & \text{for} \quad N_I^l \in \mathcal{A}_+^l \\ 0 & \text{else} \end{cases} \\ &\text{Trunctated hierarchical bases:} \qquad && \hat{M}_{IJ}^{l,k} = \begin{cases} S_{IJ}^{l,k} & \text{for} \quad N_I^l \in \mathcal{A}_+^l \quad \text{and} \quad N_J^k \in \mathcal{A}_-^k \\ 0 & \text{else} \end{cases} \end{aligned} \tag{26}$$

where $S_{IJ}^{l,k}$ has been defined in Equation (11).

In this contribution, standard hierarchical T-splines and truncated hierarchical T-splines have been considered in a unified framework, see Equation (23). Explicit basis function operations are avoided, as only matrix manipulations have to be carried out. Hence, the efficiency of the proposed approach is superior to that of recent implementations of hierarchical T-splines [8]. We note that, in particular, truncated hierarchical T-splines can be applied without explicit truncation of the basis functions. Therefore, the innovative aspect of this work is the multi-level implementation of hierarchical T-splines by using Bézier extraction in an efficient way without basis function operations. The extension of the method to three dimensions case is straightforward by employing Bézier extraction to analyse cases as in [31, 32].

Solution of Equation (23) yields the displacements $\mathbf{U}_h$ for the control points associated with hierarchical bases. However, a non-linear solution scheme requires the displacement vector $\mathbf{U}$ rather than $\mathbf{U}_h$ from previous iteration, cf. Equation (22):

$$\mathbf{U} = \mathbf{M}_h^{\mathrm{T}}\mathbf{U}_h \tag{27}$$

4.3. Implication of truncation mechanism

When constructing hierarchical T-spline bases, the hierarchical and truncated hierarchical basis functions are never computed explicitly. Instead, the hierarchical subdivision operator $\mathbf{M}_h$ builds the hierarchical T-spline bases in an implicit manner. For the hierarchical basis functions, only the contributions of active basis functions in the set $\mathcal{A}_+$ are considered, see Equation (26). During the (multi-level) Bézier extraction of T-spline bases, the numerical integration is performed separately for each active element, at each hierarchy level, regardless of whether the basis functions, which have a support over the active element, are part of hierarchical bases or not. The hierarchical subdivision operator $\mathbf{M}_h$ accounts for the activity of basis functions and recovers the correct support of active basis functions, see Figure 6. For the truncated hierarchical T-spline bases, the truncation mechanism is implemented by the hierarchical subdivision operator $\mathbf{M}_h$ in Equation (26). The contribution of basis functions in the sets $\mathcal{A}_-$ and $\mathcal{A}_+$ is considered next. In the calculation of the stiffness matrix $\mathbf{K}_h$, the mapping of the contribution of basis functions in $\mathcal{A}_-^{l+1}$ to that of basis functions in $\mathcal{A}_+^l$ is included [17]. Eventually, the stiffness matrix elements resulting from basis functions in $\mathcal{A}_-^{l+1}$ are mapped onto those related to $\mathcal{A}_+^l$ and stored afterwards in $\mathcal{A}_+^l$. Thus, the support domain of $\mathcal{A}_+^l$ is truncated. This is explained graphically in Figure 7.

5. Adaptive hierarchical refinement

We now proceed to adaptive mesh refinement, and we will present the steps and algorithms for element refinement as well as element coarsening:

$\mathcal{S}1$ Solve Equation (23) to obtain the displacement $\mathbf{U}_h$ and compute $\mathbf{U}$ using Equation (27).

$\mathcal{S}2$ Estimate the approximation error. The H_1 norm and the energy norm of the element residual are employed herein.

$\mathcal{S}3$ Mark elements for refinement and coarsening on the basis of $\mathcal{S}2$.

$\mathcal{S}4$ Refine or coarsen the marked elements. If no refinement or coarsening is required, stop the calculation, otherwise return to $\mathcal{S}1$.

5.1. *Element marking*

For element refinement, element marking can be done either using quantile marking or Dörfler marking [29]. For a domain Ω with E elements, element errors $\left\{\varepsilon_Q \middle| Q \in \Omega\right\} \subset R$ are obtained from Step $\mathcal{S}2$. Then, we define a marking parameter $\eta \in [0,\ 1]$ to determine elements which must be refined. Let $\Omega = \{Q_1, \cdots, Q_E\}$ and $\varepsilon_{Q_1} \geqslant \cdots \geqslant \varepsilon_{Q_E}$, the list of elements to be refined then reads:

$$\begin{aligned}
&\text{Quantile marking:} && \mathcal{M} = \{Q_1, \cdots, Q_k\} && \text{with} \quad k = \mathrm{ceil}\,(\eta E)\\
&\text{Dörfler marking:} && \mathcal{M} = \{Q_1, \cdots, Q_k\} && \text{with} \quad \sum_{j=1}^{k-1} \varepsilon_{Q_j} < \eta \sum_{j=1}^{E} \varepsilon_{Q_j} \quad \text{and} \quad \sum_{j=1}^{k} \varepsilon_{Q_j} \geq \eta \sum_{j=1}^{E} \varepsilon_{Q_j}
\end{aligned} \tag{28}$$

where ceil stands for the ceiling function, which rounds up to the nearest integer of ηE.

For element coarsening, the rule for element marking is problem-dependent. For example, in crack propagation analysis, elements marked for coarsening are those with large crack opening [30].

Remark 1: To obtain a well-conditioned stiffness matrix $\mathbf{K}_h$ in Equation (23) the elements adjacent to marked elements must be from the same, or at most from two consecutive hierarchy levels [17].

5.2. *Refinement procedure*

With the data structure in Section 4.1 at hand, we proceed to adaptive hierarchical refinement. Two sets of logical vectors are defined to indicate the state of elements – active or inactive – at each hierarchy level. They are initialised as **false**:

(1) $\mathbf{E}_a$: indicator of active elements. $\mathbf{E}_a^i = \{$**true** : element i is **active**$\}$.

(2) $\mathbf{E}_{ac}$: indicator of active child elements. $\mathbf{E}_{ac}^i = \{$**true** : child elements of element i are **active**$\}$.

From $\mathbf{E}_a$ and $\mathbf{E}_{ac}$, three sets of logical vectors are obtained, which indicate the state of basis functions – active or inactive – at each hierarchy level. They are initialised as **false**. Here, we define all basis functions at P hierarchy levels as: $\mathbf{N} = \left\{N^i\right\},\ (i = 1, 2, \cdots, n_{bT})$. n_{bT} being the total number of basis functions at P hierarchy levels:

(1) $\mathbf{A}_a$: indicator of basis function in the hierarchical basis function space $\mathcal{H}$ or $\mathcal{H}_T$, Equations (20) and (21). $\mathbf{A}_a^i = \left\{\textbf{true} : N^i \in \mathcal{H} \quad \text{or} \quad \mathcal{H}_T\right\}$.

(2) $\mathbf{A}_-$: indicator of basis function in set $\mathcal{A}_-$, Equation (19). $\mathbf{A}_-^i = \left\{\textbf{true} : N^i \in \mathcal{A}_-\right\}$.

(3) $\mathbf{A}_+$: indicator of basis function in set $\mathcal{A}_+$, Equation (19). $\mathbf{A}_+^i = \left\{\textbf{true} : N^i \in \mathcal{A}_+\right\}$.

A pseudo code to obtain $\mathbf{A}_a$, $\mathbf{A}_-$ and $\mathbf{A}_+$ can be found in [17]. The procedure for adaptive hierarchical refinement is given in Algorithm 1.

Algorithm 1 Adaptive hierarchical refinement

$\mathcal{A}1$ Read the geometry data to obtain the initial local knot vector $\left(\boldsymbol{\Xi}_0^1,\ \boldsymbol{\Xi}_0^2\right)$ and the initial control points $\mathbf{P}_0$.

$\mathcal{A}2$ Carry out successive cell subdivision to generate $\left(\boldsymbol{\Xi}_I^1,\ \boldsymbol{\Xi}_I^2\right)$ and $\mathbf{P}_I$ for each hierarchy level I from $\left(\boldsymbol{\Xi}_0^1,\ \boldsymbol{\Xi}_0^2\right)$ and $\mathbf{P}_0$.

$\mathcal{A}3$ Compute the subdivision operator $\mathbf{S}^{l,l+1}$ between two consecutive hierarchy levels l and $l + 1$.

$\mathcal{A}4$ Obtain the list of active elements and active child elements. For the first load step, the active elements are provided by initial T-spline mesh.

$\mathcal{A}5$ Compute the logic vectors $\mathbf{A}_a$, $\mathbf{A}_+$, $\mathbf{A}_-$ and the subdivision operator $\mathbf{M}_h$.

$\mathcal{A}6$ Solve Equation (23) and employ Equation (27) to obtain the displacement vector $\mathbf{U}$.

$\mathcal{A}7$ Check if elements should be refined or coarsened and mark them accordingly, see Algorithm 2. If there is no marked element for refinement or coarsening, stop the calculation for current load step and go to next load step. Otherwise obtain new list of active elements and active child elements on the basis of marked elements and return to $\mathcal{A}4$.

Algorithm 2 Element refinement and coarsening in adaptive hierarchical refinement.

$\mathcal{B}1$ Compute the element error ε_Q of each active element. If coarsening is required, compute the corresponding error indicators.

$\mathcal{B}2$ Mark elements for refinement and coarsening.

$\mathcal{B}3$ Refine the elements in order to obtain new list of active elements and active child elements. Here, the elements to be refined are represented as E_r and all child elements of E_r as E_{rc}.

- $\mathcal{B}3.1$ Get the old list of $\mathbf{E}_a$ and $\mathbf{E}_{ac}$.
- $\mathcal{B}3.2$ Set $\mathbf{E}_a(E_r) =$ **false**, $\mathbf{E}_{ac}(E_r) =$ **true**.
- $\mathcal{B}3.3$ Set $\mathbf{E}_a(E_{rc}) =$ **true**, $\mathbf{E}_{ac}(E_{rc}) =$ **false**.

$\mathcal{B}4$ Coarsen the elements in order to obtain the updated list of active elements and active child elements, $\mathbf{E}_a$ and $\mathbf{E}_{ac}$. Elements to be coarsened are denoted as E_c, the parent elements of E_c as E_p, the parent elements of E_p as E_{gp}, and all child elements of E_p as E_{Ac}.

- $\mathcal{B}4.1$ Get the list of $\mathbf{E}_a$ and $\mathbf{E}_{ac}$ after element refinement.
- $\mathcal{B}4.2$ Set $\mathbf{E}_a(E_p) =$ **true**, $\mathbf{E}_{ac}(E_{gp}) =$ **true**.
- $\mathcal{B}4.3$ Set $\mathbf{E}_a(E_{Ac}) =$ **false**, $\mathbf{E}_{ac}(E_p) =$ **false**.

5.3. *Update process*

During mesh refinement and coarsening, elements are introduced in the set of active elements. For non-linear problems, it is then necessary to transfer the displacement from previous time step t to provide initial values for newly activated elements at $t + \Delta t$. The displacements that result from a transfer from coarse elements to fine elements are exact. However, in the reverse process, some information is generally lost.

When the displacement vector ${}^{t}\mathbf{U}$ is mapped to generate a new initial state vector during mesh refinement, ${}^{t+\Delta t}_{0}\mathbf{U}$, the process is exact:

$$ {}^{t+\Delta t}_{0}\mathbf{U}^{l+1} = \left(\tilde{\mathbf{S}}^{l,l+1}\right)^{\mathrm{T}}\, {}^{t}\mathbf{U}^{l} \tag{29} $$

where l is the hierarchy level, and $\tilde{\mathbf{S}}^{l,l+1}$ denotes the modified subdivision operator from Equation (12)

$$ \tilde{S}^{\,l,l+1}_{IJ} = \begin{cases} S^{\,l,l+1}_{IJ} & \text{for} \quad N^{l+1}_{J} \in {}^{t+\Delta t}\mathcal{A}^{l+1} \quad \text{or} \quad {}^{t+\Delta t}\mathcal{A}^{l+1}_{T} \\ 0 & \text{else} \end{cases} \tag{30} $$

with ${}^{t+\Delta t}\mathcal{A}$ or ${}^{t+\Delta t}\mathcal{A}_{T}$ the space of hierarchical basis functions at $t + \Delta t$.

For coarsening, a global least-square fit is employed to perform the mapping, and we minimise:

$$ \psi = \int_{\Omega} \left\| {}^{t+\Delta t}_{0}\mathbf{u} - {}^{t}\mathbf{u} \right\| \mathrm{d}\Omega = \int_{\Omega} \left\| {}^{t+\Delta t}\mathbf{N}_{A}\; {}^{t+\Delta t}_{0}\mathbf{U} - {}^{t}\mathbf{u} \right\| \mathrm{d}\Omega \tag{31} $$

with respect to ${}^{t+\Delta t}_{0}\mathbf{U}$, yielding:

$$ \mathbf{M}_{N}\; {}^{t+\Delta t}_{0}\mathbf{U} = \mathbf{p} \quad \text{with} \quad \mathbf{M}_{N} = \int_{\Omega} \left({}^{t+\Delta t}\mathbf{N}_{A}\right)^{\mathrm{T}}\, {}^{t+\Delta t}\mathbf{N}_{A} \mathrm{d}\Omega; \quad \mathbf{p} = \int_{\Omega} \left({}^{t+\Delta t}\mathbf{N}_{A}\right)^{\mathrm{T}}\, {}^{t}\mathbf{u} \mathrm{d}\Omega \tag{32} $$

where $\mathbf{u}$ contains the displacements, and ${}^{t+\Delta t}\mathbf{N}_{A}$ the basis functions associated with active elements at $t + \Delta t$. $\mathbf{M}_{N}$ can be computed directly through the Gaussian quadrature on each active element at $t + \Delta t$. However, the integration of $\mathbf{p}$ needs to be modified as below:

$$ \mathbf{p} = \int_{\Omega} \left({}^{t+\Delta t}\mathbf{N}_{A}\right)^{\mathrm{T}}\, {}^{t}\mathbf{u} \mathrm{d}\Omega = \int_{\Omega} \left({}^{t+\Delta t}\mathbf{N}_{A}\right)^{\mathrm{T}} \left({}^{t}\mathbf{N}_{A}\right) {}^{t}\mathbf{U} \mathrm{d}\Omega \tag{33} $$

where the second integration is carried out on each active element at t. ${}^{t}\mathbf{N}_{A}$ contains basis functions associated with active elements at t.

6. Examples

We will now present four examples, aimed to validate the method in terms of accuracy and assess its performance. Truncated hierarchical basis functions will be used throughout to describe the geometry of domain and to approximate the field variable.

In the examples, only element refinement will be considered, no coarsening. The error of each element is computed using the H^1-norm and the energy norm $||\mathbf{e}||$ [33], which are defined as:

$$ \begin{aligned} ||\mathbf{u} - \bar{\mathbf{u}}||_{H^1(\Omega_e)} &= \sqrt{\left(\int_{\Omega_e} (\mathbf{u} - \bar{\mathbf{u}})^{\mathrm{T}} \cdot (\mathbf{u} - \bar{\mathbf{u}})\; dS + \int_{\Omega_e} (\mathbf{u} - \bar{\mathbf{u}})'^{\mathrm{T}} \cdot (\mathbf{u} - \bar{\mathbf{u}})'\; dS \right)} \\ ||\mathbf{e}||_{e} &= \sqrt{\frac{1}{2} \int_{\Omega_e} (\boldsymbol{\sigma} - \bar{\boldsymbol{\sigma}})^{\mathrm{T}} \cdot \mathbb{C}^{-1} \cdot (\boldsymbol{\sigma} - \bar{\boldsymbol{\sigma}})\; dS} \end{aligned} \tag{34} $$

where $\mathbf{u}$ and $\boldsymbol{\sigma}$ stand for the analytical solution, $\bar{\mathbf{u}}$ and $\bar{\boldsymbol{\sigma}}$ denote the approximate solution, $\forall \mathbf{u}, \bar{\mathbf{u}} \in \left(H^1(\Omega_e)\right)^2$, and $\mathbb{C}$ is the elastic stiffness tensor. $(\mathbf{u} - \bar{\mathbf{u}})'$ is the first derivative of $(\mathbf{u} - \bar{\mathbf{u}})$ with respect to x and y, respectively. These error measures are employed for convergence studies, where mesh refinement is considered. The domain error is obtained by summing up the element error.

$$||\mathbf{u} - \bar{\mathbf{u}}||_{H^1(\Omega)} = \sqrt{\sum\nolimits_e \left(||\mathbf{u} - \bar{\mathbf{u}}||_{H^1(\Omega_e)}\right)^2} \qquad ||\mathbf{e}|| = \sqrt{\sum\nolimits_e ||\mathbf{e}||_e^2} \tag{35}$$

In marking elements for refinement and coarsening, the relative error of each element is employed to define the element-wise error ε_e:

$$\varepsilon_e = \frac{||\mathbf{u} - \bar{\mathbf{u}}||_{H^1(\Omega_e)}}{\sqrt{\left(\int_{\Omega_e} \mathbf{u}^{\mathrm{T}} \cdot \mathbf{u}\, dS + \int_{\Omega_e} \mathbf{u}'^{\mathrm{T}} \cdot \mathbf{u}'\, dS\right)}} \qquad \varepsilon_e = \frac{||\mathbf{e}||_e}{\sqrt{\frac{1}{2}\int_{\Omega_e} \boldsymbol{\sigma}^{\mathrm{T}} \cdot \mathbb{C}^{-1} \cdot \boldsymbol{\sigma}\, dS}} \tag{36}$$

In general, T-spline meshes are generated by adaptive local refinement of NURBS meshes [28]. To provide a good illustration of hierarchical T-spines, we will directly give the initial mesh of the first two examples as a T-mesh with T-junctions. For the rest examples, the initial mesh will be defined by quadratic NURBS surface, since it is a special case of T-spline mesh. For all examples the geometry is modelled with same polynomial degree p in both parametric directions.

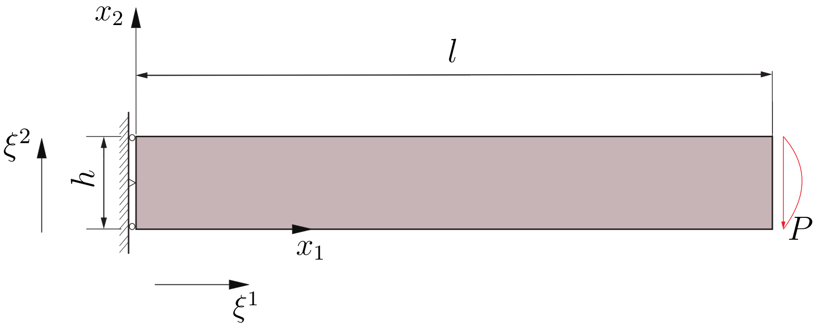


Figure 8: Cantilever beam subjected to a parabolic traction: problem definition.

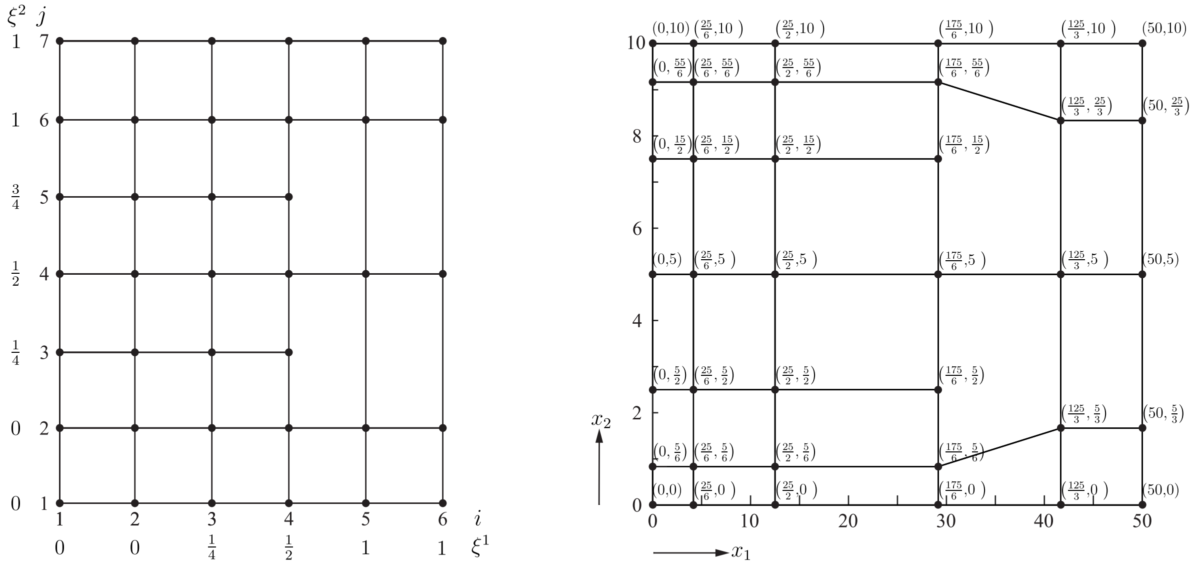
6.1. *Cantilever beam under a parabolic traction*

First, we will test the method to construct a T-spline mesh T^α from an initial T-spline mesh T^0, see Sections 3.1 and 4. A cantilever beam of length l and height h is considered, see Figure 8. It is fixed at $x_1 = 0$ and subjected to a parabolic traction at the free end $x_1 = l$ with a resultant force P. The geometry and material parameters read: length $l = 50$ m, height $h = 10$ m, Young's module $E = 100$ Pa, Poisson's ratio $\nu = 0.3$, and $P = 1$ N. The analytical solution of the displacement field can be found in [34].

The domain is discretised by cubic T-splines. The initial T-spline mesh T^0 is given in Figure 9, which shows the index domain, the parameter domain and the physical domain. A hierarchy of 3 levels is constructed from T^0. The T-spline mesh T^2 is shown in Figure 10(a).

In the analysis we apply Dirichlet boundary conditions at $x_1 = 0$ and the traction $\bar{\mathbf{t}} = [0, \sigma_{12}]^{\mathrm{T}}$ at $x_1 = L$. For the Dirichlet boundary condition, we could not directly impose the exact displacement at control points due to the non-interpolatory nature of T-splines. They are computed from the displacement and the parameter value $\left(\xi^1, \xi^2\right)$ at Gauss integration points.

For the computation of stiffness matrix $\mathbf{K}$ and right-hand force vector $\mathbf{F}$, three-point Gaussian quadrature is employed to avoid shear locking effect in the simulation. Since we utilize cubic T-splines, the numerical solution should be exact up to machine precision. The stress components σ_{11} and σ_{12} are given in Figures 11. The error in the stress value, defined as the difference between numerical solution and analytical solution, is shown. The values are indeed around the machine precision, which implies the proposed method of constructing hierarchical T-meshes works well.



ξ^2 j
1 7
1 6
$\frac{3}{4}$ 5
$\frac{1}{2}$ 4
$\frac{1}{4}$ 3
0 2
0 1
1 2 3 4 5 6 i
0 0 $\frac{1}{4}$ $\frac{1}{2}$ 1 1 ξ^1
(0,10) ($\frac{25}{6}$,10) ($\frac{25}{2}$,10) ($\frac{175}{6}$,10) ($\frac{125}{3}$,10) (50,10)
(0, $\frac{55}{6}$) ($\frac{25}{6}$, $\frac{55}{6}$) ($\frac{25}{2}$, $\frac{55}{6}$) ($\frac{175}{6}$, $\frac{55}{6}$)
($\frac{125}{3}$, $\frac{25}{3}$) (50, $\frac{25}{3}$)
(0, $\frac{15}{2}$) ($\frac{25}{6}$, $\frac{15}{2}$) ($\frac{25}{2}$, $\frac{15}{2}$) ($\frac{175}{6}$, $\frac{15}{2}$)
(0,5) ($\frac{25}{6}$,5) ($\frac{25}{2}$,5) ($\frac{175}{6}$,5) ($\frac{125}{3}$,5) (50,5)
(0, $\frac{5}{2}$) ($\frac{25}{6}$, $\frac{5}{2}$) ($\frac{25}{2}$, $\frac{5}{2}$) ($\frac{175}{6}$, $\frac{5}{2}$)
($\frac{125}{3}$, $\frac{5}{3}$) (50, $\frac{5}{3}$)
(0, $\frac{5}{6}$) ($\frac{25}{6}$, $\frac{5}{6}$) ($\frac{25}{2}$, $\frac{5}{6}$) ($\frac{175}{6}$, $\frac{5}{6}$)
(0,0) ($\frac{25}{6}$,0) ($\frac{25}{2}$,0) ($\frac{175}{6}$,0) ($\frac{125}{3}$,0) (50,0)
10 8 6 4 2 0
x_2
0 10 20 30 40 50
x_1

(a) T-spline mesh in the index and the parameter domain

(b) T-spline mesh in the physical domain

Figure 9: Initial T-spline mesh T^0 for the cantilever beam. Here, the beam is given in the index domain $(i,\ j)$, in the parameter domain $\left(\xi^1,\ \xi^2\right)$, and in the physical domain $(x_1,\ x_2)$. The anchors are indicated by circular dots. The polynomial degree of T-splines is $p = 3$ and the weights of anchors are $w = 1$.

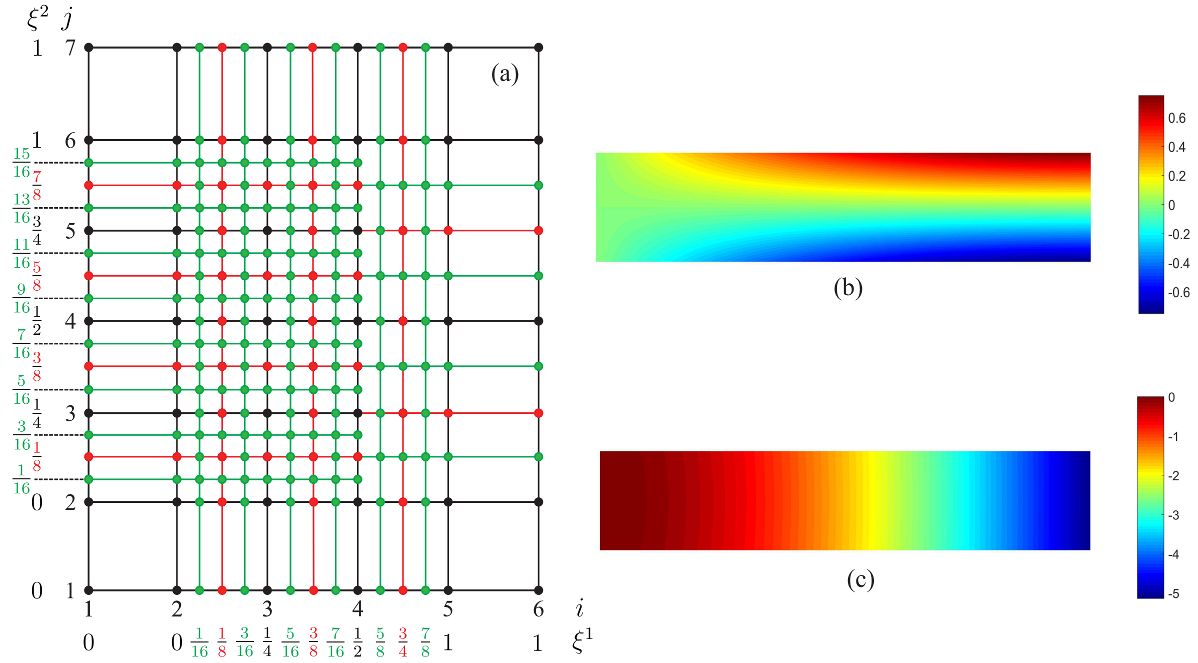
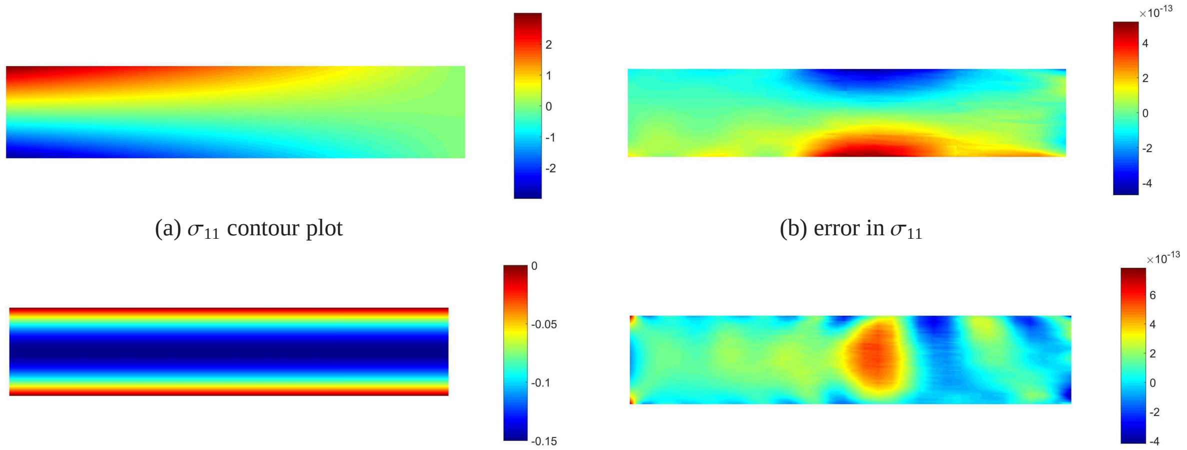


Figure 10: T-spline mesh T^2 and the displacement distribution for the cantilever beam. (a) T-spline mesh T^2 in the index and the parameter domains. The anchors are indicated by circular dots. Black denotes the mesh and the anchors of initial T-spline mesh, T^0, red stands for the mesh and the anchors generated for T^1, and green represents the mesh and the anchors generated for T^2; (b) contour plot for u_1; (c) contour plot for u_2.



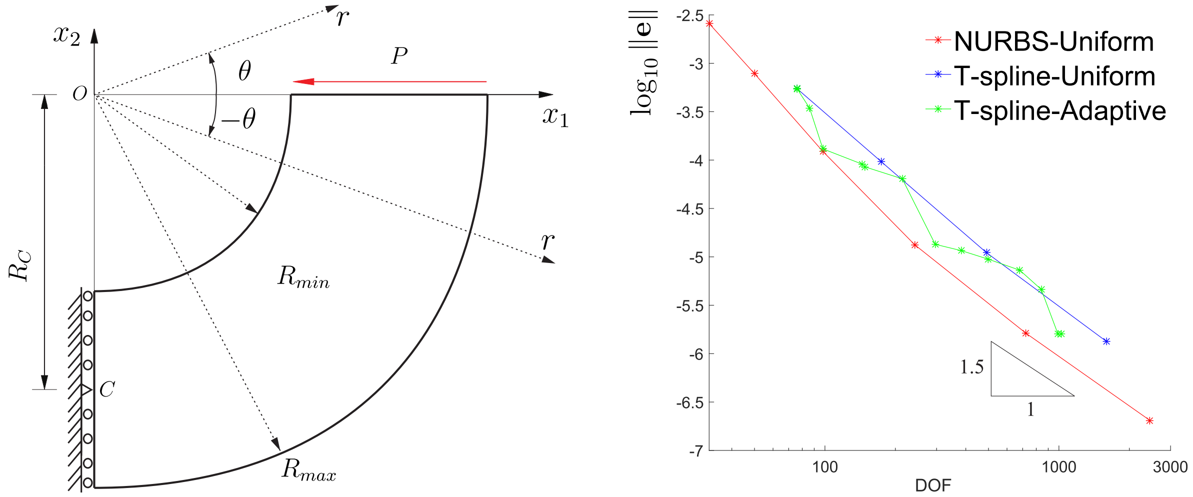
(a) σ_{11} contour plot

(b) error in σ_{11}

(c) σ_{12} contour plot

(d) error in σ_{12}

Figure 11: Stress distribution for the cantilever beam.



(a) problem setting

(b) energy norm

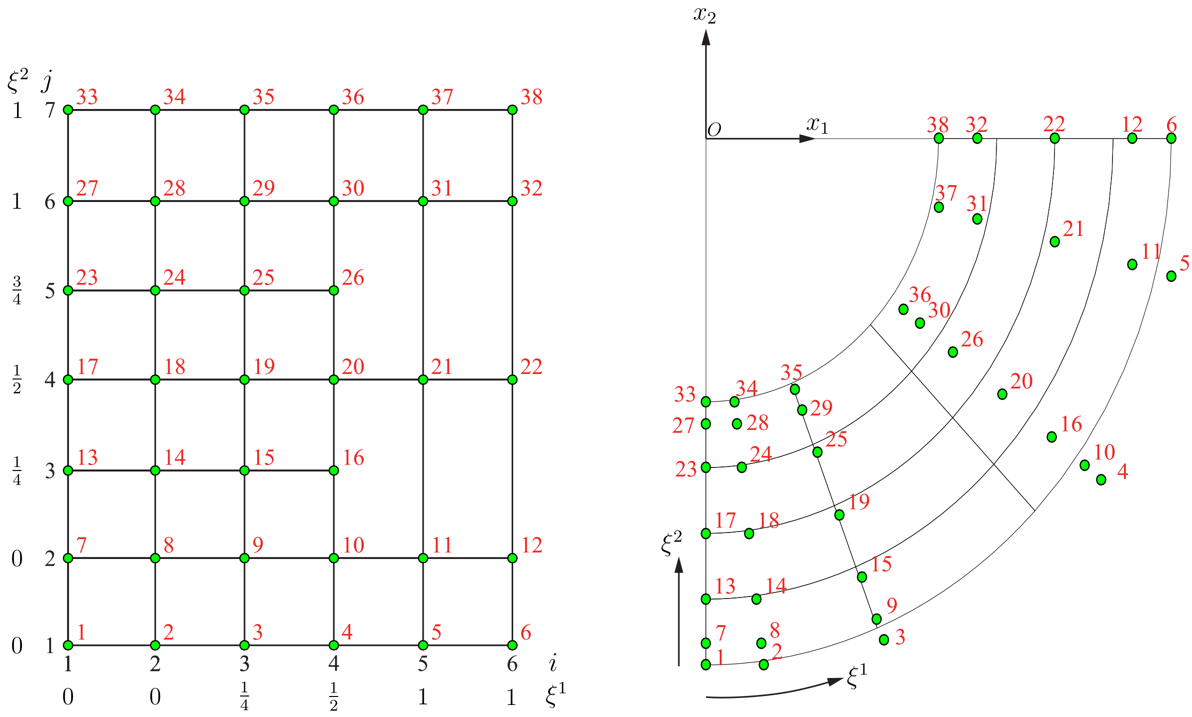
Figure 12: Bending of a curved beam by a force at the end.

6.2. *Bending of a curved beam by a force at the end*

A curved beam with a unit thickness is considered next, see Figure 12(a). The beam is subjected to shear tractions at the free end $x_2 = 0$, with resultant force P. The geometry parameters are: inner radius $R_{min} = 5\,\mathrm{m}$ and outer radius $R_{max} = 10\,\mathrm{m}$. A linear elastic material is considered with: Young's module $E = 100\,\mathrm{kPa}$, Poisson's ratio $\nu = 0.3$, while $P = -1\,\mathrm{N}$. The analytical solution of displacements in polar coordinate system can be found in [34].

The problem has been solved using cubic T-spline bases with an initial T-spline mesh T^0, see Figure 13. A hierarchy of four levels is constructed from $\underline{T}^0$. The Dirichlet boundary conditions are weakly imposed at the boundary $x_1 = 0$. The Neumann boundary condition $\bar{\mathbf{t}} = [\sigma_{12}, 0]^T$ is applied at $x_2 = 0$ in an exact manner.

During adaptive refinement, elements are selected for refinement using Dörfler marking with $\eta = 0.4$. The convergence in the energy norm, $\|\mathbf{e}\|$, is shown in Figure 12(b) for cubic NURBS and cubic T-splines. For uniform refinement, the convergence rate will approach the optimal rate of convergence $k = -p/2 = -1.5$. The error level of the solution for T-spline bases is higher than that for NURBS bases. A possible explanation is that the pre-defined T-splines do not efficiently model the problem localisation. For the case of adaptive refinement, the convergence plot exhibits a



(a) T-spline mesh in the index and parameter domain

(b) T-spline mesh in the physical domain

Figure 13: Initial T-spline mesh T^0 for the curved beam. The beam is shown in the index domain (i, j), in the parameter domain $\left(\xi^1, \xi^2\right)$, and in the physical domain (x_1, x_2). The anchors are indicated by circular dots. The polynomial degree of T-splines is $p = 3$. The local knot vectors and the coordinates of all anchors are given in Appendix A.

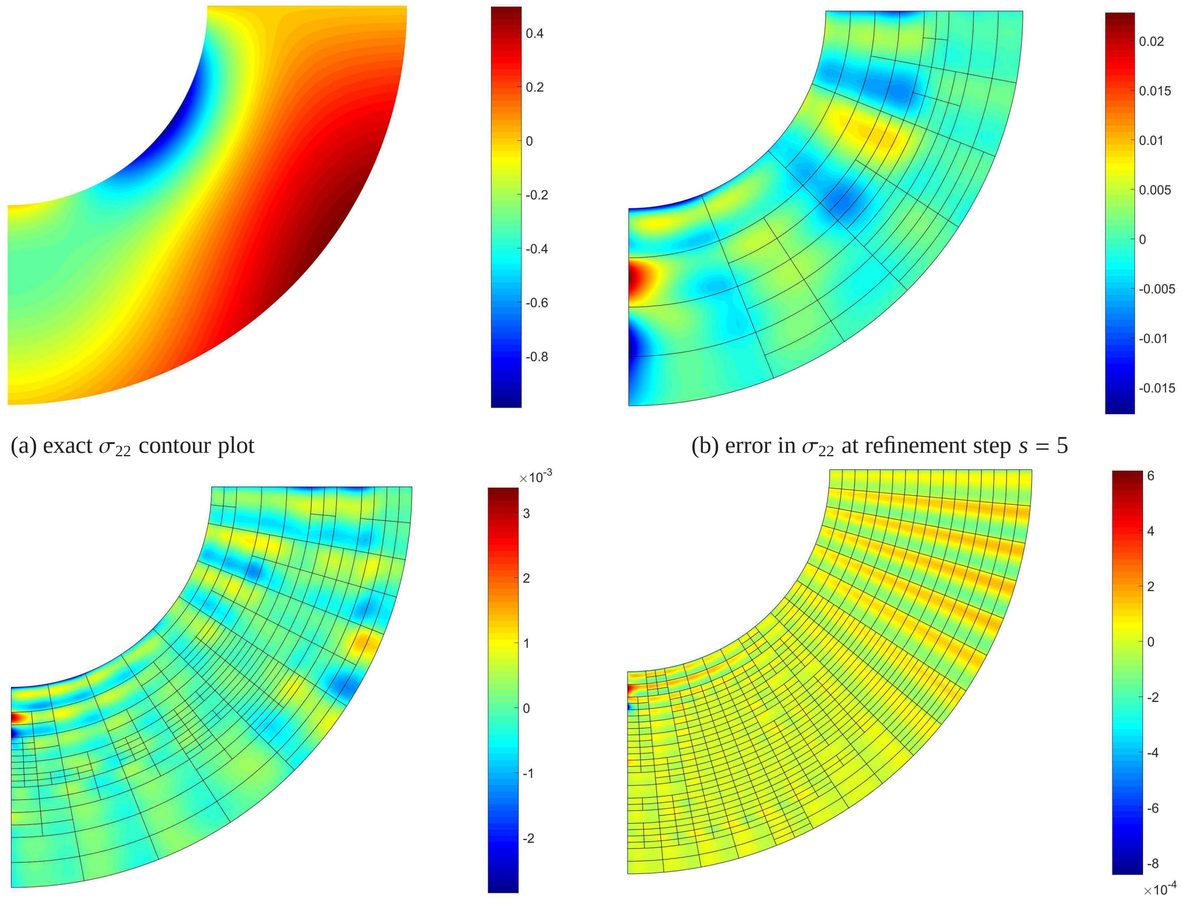
zig-zag behaviour with a convergence rate which approximates the optimal rate $k = -p/2 = -1.5$. In general, the error level for adaptive mesh refinement is between those of uniform T-spline and NURBS mesh refinement.

Figure 14 gives the contour plot of exact solution and errors in the stress component σ_{22} obtained numerically. In the figure, Bézier meshes at refinement steps $s = 5, 9, 14$ are indicated by solid lines. They show a local refinement of the mesh, giving an improved resolution of stress gradients, especially in the part with T-junctions in initial T-spline mesh T^0.

6.3. *Infinite plate with a circular hole*

The third example considers the adaptive hierarchical refinement using only NURBS. We have employed the cell subdivision approach to construct hierarchical basis functions, see Sections 3 and 4. An infinite plate with a circular hole is considered, with a radius $R = 1$ m, see Figure 15. The material properties are: Young's modulus $E = 100$ N/m^2, Poisson's ratio $\nu = 0.0$ and thickness $h = 1$ m. The exact solutions of radial and tangential displacement are:

$$\begin{aligned} u_r &= \frac{T_x r\cos(2\theta)}{2E}\left[(1+\nu) + 4\frac{R^2}{r^2} - (1+\nu)\frac{R^4}{r^4}\right] + \frac{T_x r}{2E}\left[(1-\nu) + (1+\nu)\frac{R^2}{r^2}\right] \\ u_\theta &= -\frac{T_x r\sin(2\theta)}{2E}\left[(1+\nu) + 2(1-\nu)\frac{R^2}{r^2} + (1+\nu)\frac{R^4}{r^4}\right] \end{aligned} \tag{37}$$



(a) exact σ_{22} contour plot

(b) error in σ_{22} at refinement step $s = 5$

(c) error in σ_{22} at refinement step $s = 9$

(d) error in σ_{22} at refinement step $s = 14$

Figure 14: Stress distribution for the curved beam. Bézier meshes of active elements at each refinement step are indicated by solid lines.

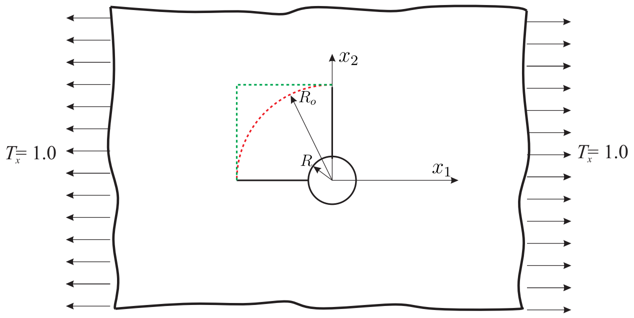
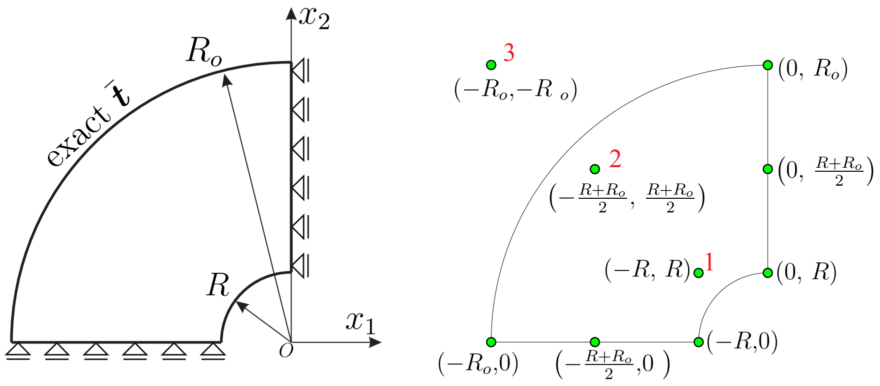


Figure 15: Infinite plate with a circular hole.

where θ is the azimuthal coordinate. The corresponding stress components read:

$$\begin{aligned}\sigma_r &= \frac{T_x}{2}\left(1-\frac{R^2}{r^2}\right)+\frac{T_x\cos2\theta}{2}\left(\frac{3R^4}{r^4}-\frac{4R^2}{r^2}+1\right)\\ \sigma_\theta &= \frac{T_x}{2}\left(1+\frac{R^2}{r^2}\right)-\frac{T_x\cos2\theta}{2}\left(\frac{3R^4}{r^4}+1\right)\\ \sigma_{r\theta} &= \frac{T_x\sin2\theta}{2}\left(\frac{3R^4}{r^4}-\frac{2R^2}{r^2}-1\right)\end{aligned} \tag{38}$$

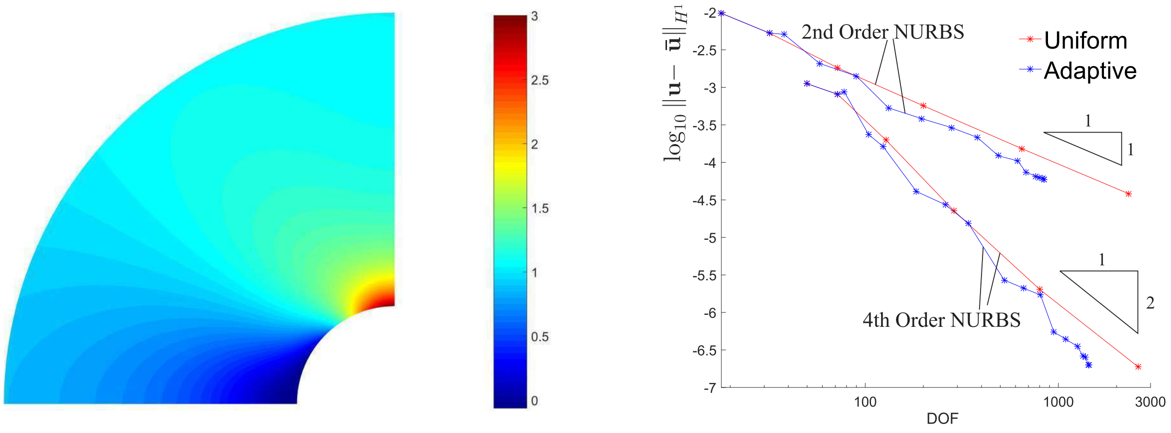
By virtue of symmetry only one quarter of the plate has to be modelled. Two different geometries are used to represent the problem: a quarter of an annulus and a finite quarter plate. The exact traction from analytical solution is imposed at the free boundary, see e.g. [35, 36].



(a) problem definition of the quarter of an annulus

(b) Bézier physical mesh and control points

Figure 16: Infinite plate with a circular hole: a quarter of an annulus representation and initial T-spline mesh T^0 in the physical domain. The weights of control points 1, 2 and 3 are $\sqrt{2}/2$; the weights of other control points are 1.



(a) exact σ_{11} contour plot

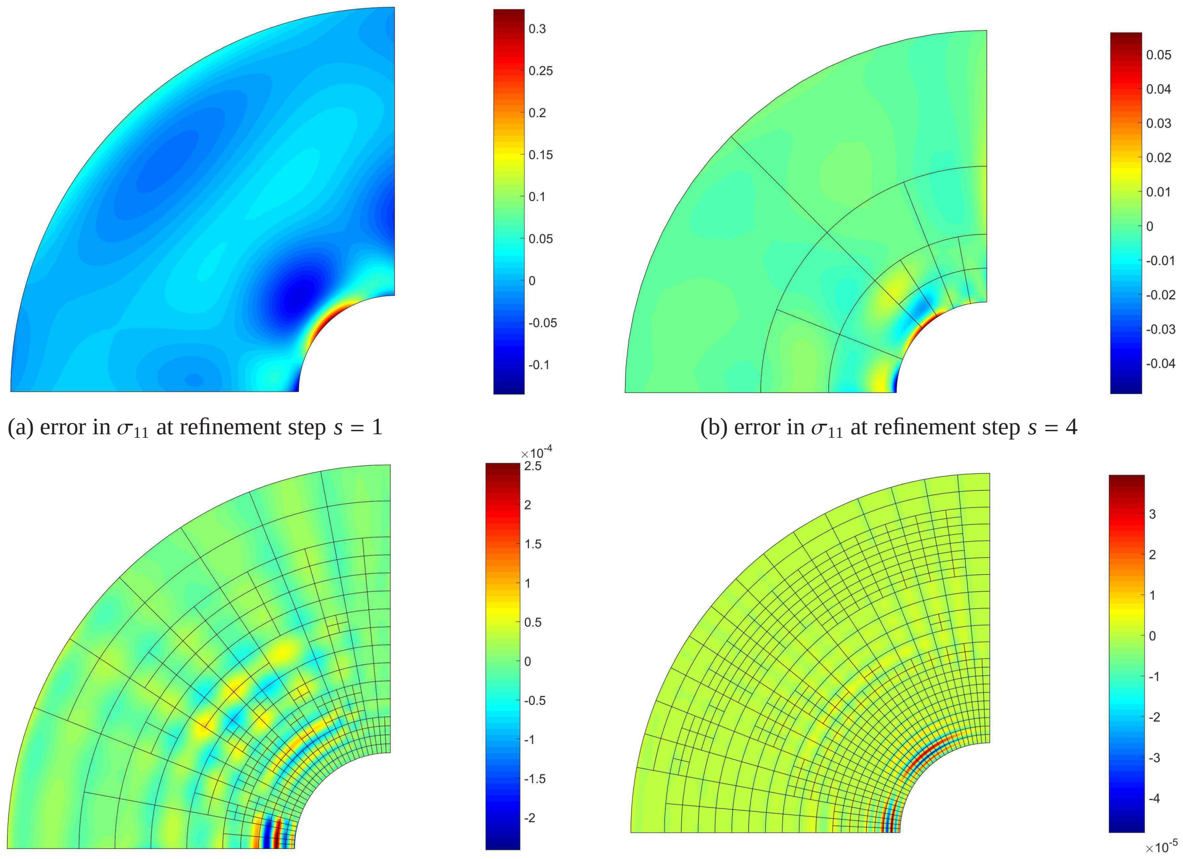
(b) H^1 norm

Figure 17: Quarter of an annulus representation: exact solution for σ_{11} and H^1 norm.

6.3.1. Quarter of an annulus representation

The geometry and boundary conditions of the annulus are shown in Figure 16a. The domain Ω is initially discretised by NURBS with a polynomial degree $p = 2$, with knot vectors $\mathbf{\Xi}^1 = [0,\ 0,\ 0,\ 1,\ 1,\ 1]$ and $\mathbf{\Xi}^2 = [0,\ 0,\ 0,\ 1,\ 1,\ 1]$. The coordinates of control points $\mathbf{P}$ are given in Figure 16(b). In this example, we consider NURBS with polynomial degrees $p = 2$ and $p = 4$. For the fourth-order NURBS, the knot vector and the control points are updated by order elevation from second-order NURBS. The NURBS discretizations for $p = 2, 4$ are taken as initial T-spline mesh T^0.

In the analysis, a hierarchy of six levels is constructed from initial mesh T^0. In adaptive refinement, elements chosen for refinement are decided by quantile marking with $\eta = 0.2$. The convergence in the H^1 norm is shown in Figure 17(b). Uniform mesh refinement can achieve an optimal convergence rate $k = -p/2$. Local mesh refinement can reduce the error, but not the convergence rate. The reduced error level may be due to the effect of the resolution of stress gradient by local mesh refinement. The exact solution of σ_{11} points to a stress concentration at $(x_1,\ x_2) = (0,\ 1)$,



(a) error in σ_{11} at refinement step $s = 1$

(b) error in σ_{11} at refinement step $s = 4$

(c) error in σ_{11} at refinement step $s = 10$

(d) error in σ_{11} at refinement step $s = 15$

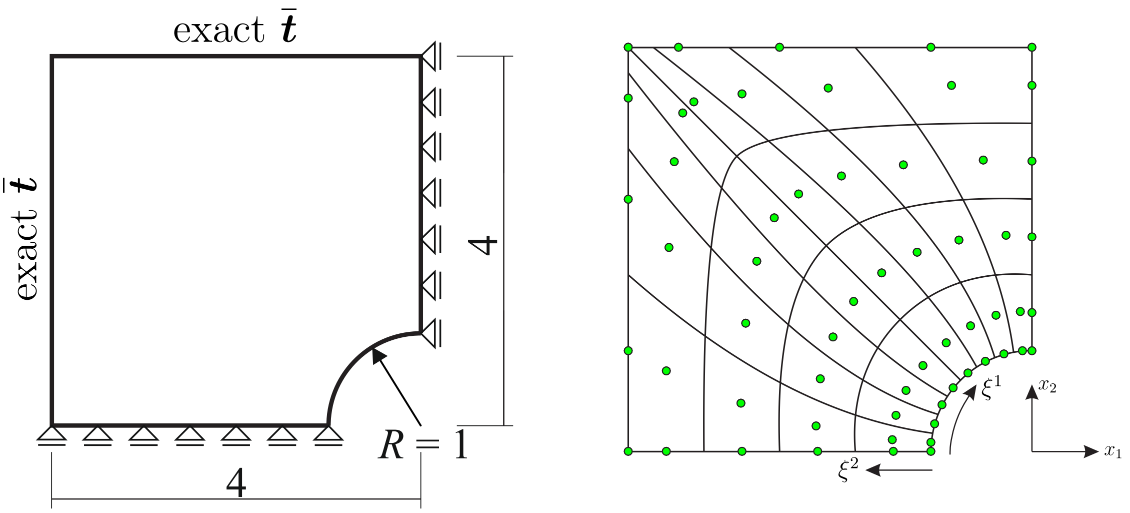
Figure 18: Bézier meshes and error in σ_{11} for active elements at each refinement step for fourth-order NURBS bases. The Bézier meshes of active elements are indicated by solid lines.

see Figure 17(a). Figure 18 presents the error of σ_{11} at each refinement step for the fourth-order NURBS. The figures show a local refinement of the mesh around the hole, where smooth stress gradient is achieved.

6.3.2. Finite quarter plate representation

For this case the geometry and boundary conditions are shown in Figure 16(a). The domain Ω is discretised by NURBS with a polynomial degree $p = 2$, with knot vectors $\mathbf{\Xi}^1 = [0, 0, 0, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1, 1, 1]$ and $\mathbf{\Xi}^2 = [0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1]$. These knot vectors are obtained through h-refinement of open knot vectors, see [1]. Accordingly, the number of control points $\mathbf{P}$ is adapted. The corresponding physical mesh and anchors are shown in Figure 16(b). Herein, we consider second-order as well as fourth-order NURBS bases to discretize the domain. For the fourth-order NURBS bases, the knot vector and control points are obtained by order elevation from second-order NURBS bases. For the construction of initial T-spline mesh, T^0, the initial local knot vectors $\left(\mathbf{\Xi}_0^1, \mathbf{\Xi}_0^2\right)$ and the coordinates of anchors can be derived in a straightforward manner from $\mathbf{\Xi}^1$, $\mathbf{\Xi}^2$ and $\mathbf{P}$.

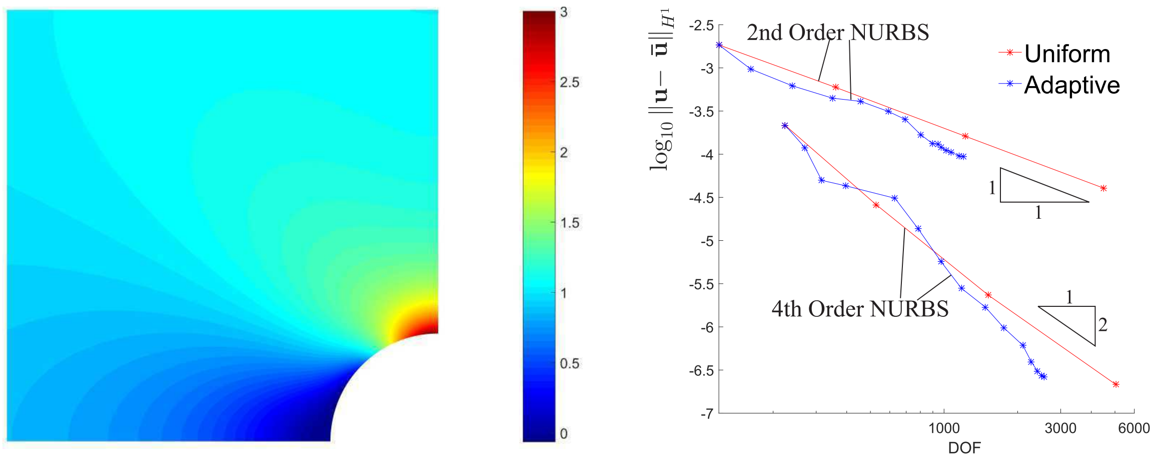
We consider a hierarchy of four levels to construct hierarchical basis functions. It is constructed from initial T-spline mesh T^0. Elements are refined by adaptive hierarchical refinement with quantile marking ($\eta = 0.2$). The convergence plot of the H^1 norm is shown in Figure 20(b). As expected, the optimal convergence rate $k = -p/2$ is achieved for uniform and for adaptive mesh refinement. The error level for adaptive mesh refinement is lower than that for uniform mesh refinement. This is due to the fact that adaptive mesh refinement smoothens the stress gradient. The phenomenon of smoothing the stress gradient is illustrated in Figure 21. The mesh around the hole with stress concentration is refined gradually until the lowest hierarchy level. With adaptive mesh refinement, the error level is



(a) problem definition for the plate

(b) Bézier mesh and anchors

Figure 19: Infinite plate with circular hole: problem definition and initial T-spline mesh T^0 in the physical domain.



(a) exact σ_{11} contour plot

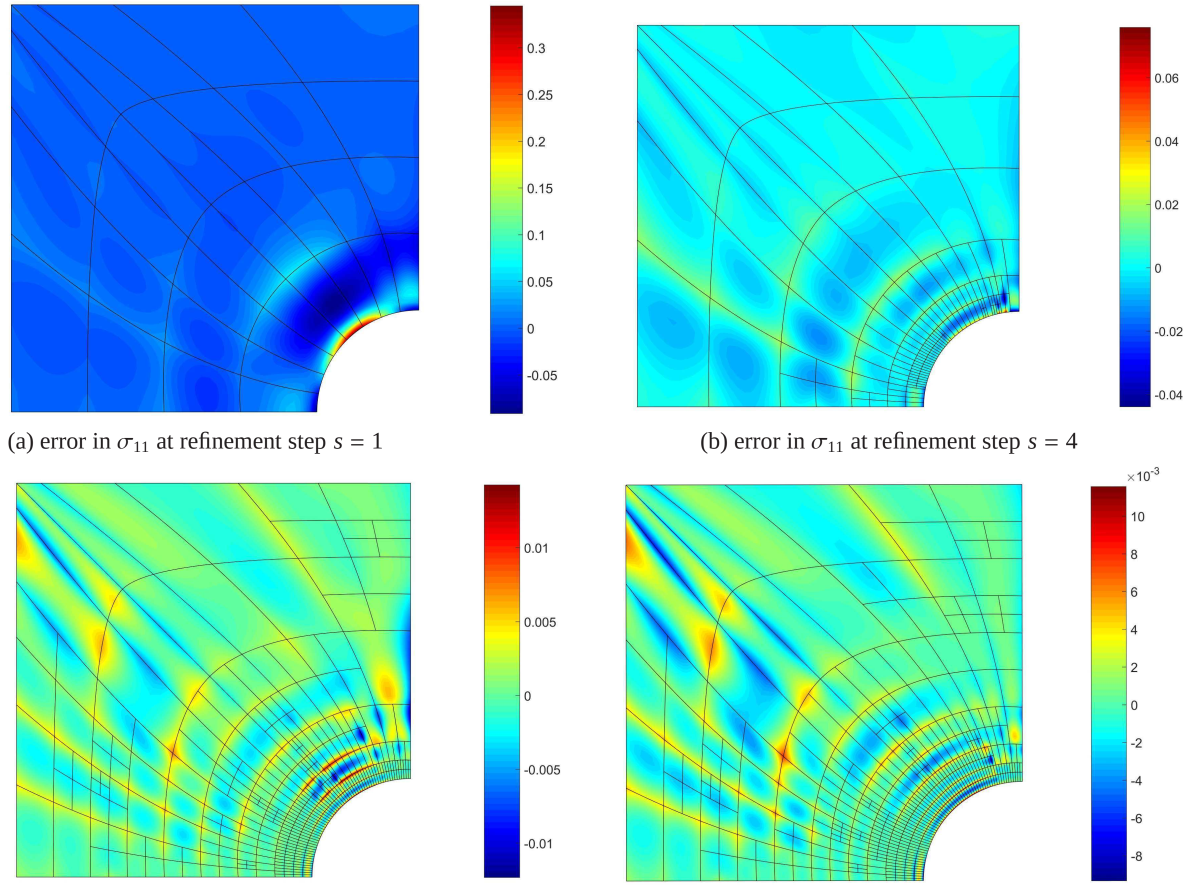
(b) H^1 norm

Figure 20: Representation of a finite quarter of a plate: exact solution of σ_{11} and H^1 norm.

reduced for the whole domain, which indicates that adaptive mesh refinement could not only efficiently model the localisation, but also improve the accuracy globally.

6.4. Poisson problem on an L-shaped domain

As final example we consider a Poisson problem, which solves for the temperature u on an L-shaped domain, Figure 22(a). The definition of the L-shaped domain and the analytical solution of the problem can be found in [17]. The L-shaped domain has been modelled by a single C^1 continuous quadratic B-spline patch, Figure 22(b). To define the patch, the knot vectors are given as $\boldsymbol{\Xi}^1 = [0, 0, 0, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1, 1, 1]$ and $\boldsymbol{\Xi}^2 = [0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1]$. The corresponding Bézier physical mesh and anchors are shown in Figure 22(b). For the construction of initial T-spline mesh, T^0, the local knot vectors $\left(\boldsymbol{\Xi}_0^1, \boldsymbol{\Xi}_0^2\right)$ are derived from $\boldsymbol{\Xi}^1$ and $\boldsymbol{\Xi}^2$. A hierarchy of five levels is constructed from the initial mesh T^0, which constitutes the hierarchical T-spline basis function space. During the adaptive refinement, elements are selected for refinement using quantile marking with $\eta = 0.2$.



(a) error in σ_{11} at refinement step $s = 1$

(b) error in σ_{11} at refinement step $s = 4$

(c) error in σ_{11} at refinement step $s = 8$

(d) error in σ_{11} at refinement step $s = 10$

Figure 21: Bézier meshes and the error in σ_{11} for active elements at each refinement step for second-order NURBS. The Bézier meshes of active elements are indicated by solid lines.

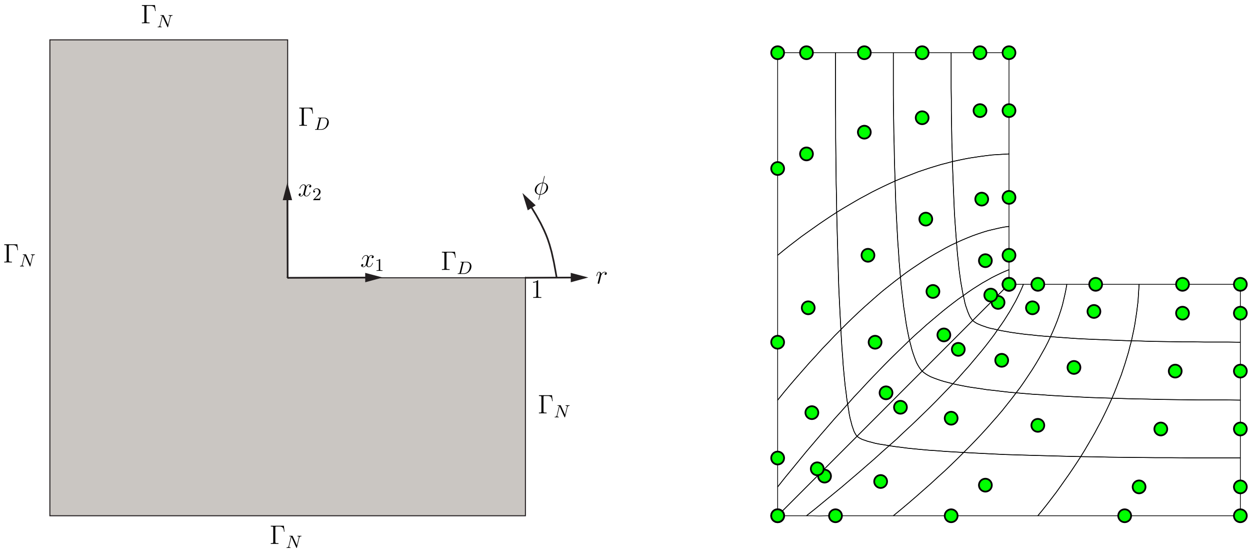
Due to the singularity at the re-entrant corner $(x_1, x_2) = (0, 0)$, the rate of convergence k of the H^1 norm with respect to the total number of degrees of freedom is given as:

$$k = -\frac{1}{2}\ \min\left(p,\ \frac{\pi}{2\pi - \beta}\right) = -\frac{1}{2}\ \min\left(p,\ \frac{2}{3}\right) = -\frac{1}{3} \tag{39}$$

Figure 23(b) presents a comparision of the convergence of adaptive mesh refinement and uniform mesh refinement. For uniform refinement, the corresponding rate of convergence is $k = -1/3$, Figure 23(b). The optimal convergence rate $k = -1$ is recovered by adaptive refinement, see Figure 23(b). It shows that the error level for adaptive refinement is smaller than that for uniform refinement. This is because adaptive refinement better captures the gradient around the re-entrant corner, Figure 24. From these figures it is observed that the mesh around the re-entrant corner is refined gradually until the lowest hierarchy level.

7. Concluding remarks

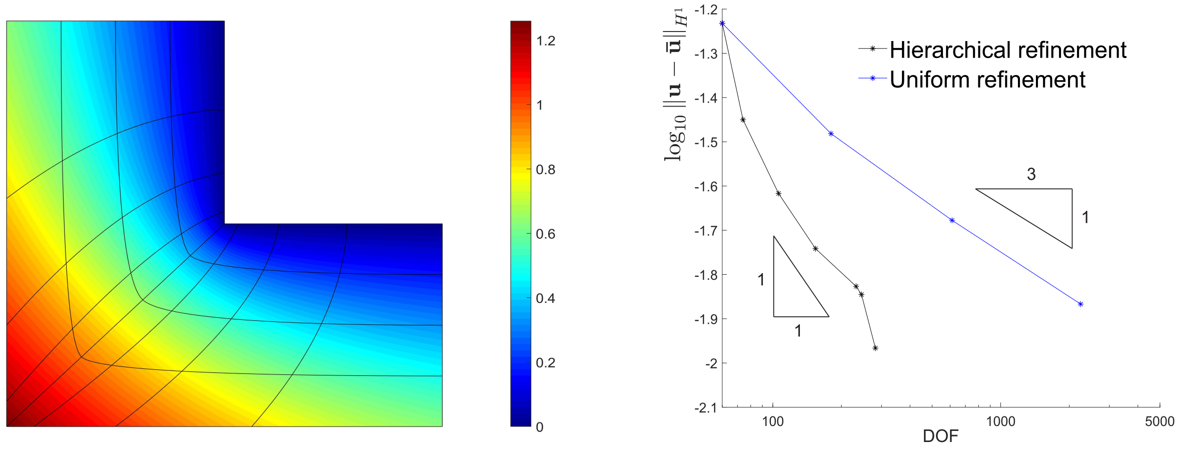
We have developed a Bézier extraction framework for hierarchical T-splines, treating standard hierarchical T-splines and truncated hierarchical T-splines in a unified and straightforward manner. Explicit basis function operations are avoided, and only matrix manipulations have to be carried out. In particular, no explicit truncation of basis functions is needed for the application of truncated hierarchical T-splines. The use of an element-wise point of view



(a) problem definition of the plate

(b) Bézier physical mesh and anchors

Figure 22: Poisson problem on an L-shaped domain: problem definition and initial quadratic T-spline mesh in the physical domain.



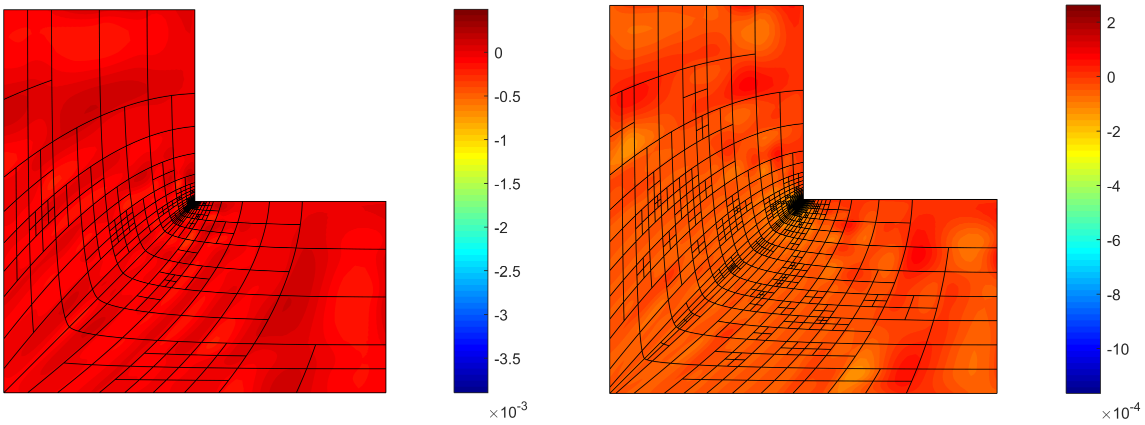
(a) exact u contour plot

(b) H^1 error norm

Figure 23: Poisson problem on an L-shaped domain: exact solution of u and H^1 error norm.

facilitates the implementation in existing finite element codes. Moreover, adaptive refinement can be incorporated in the analysis directly.

Algorithmically, a multi-level T-spline mesh is generated by successive cell subdivisions of an initial coarse T-spline mesh. Subsequently, on each hierarchy level, Bézier extraction is applied to obtain the stiffness matrix. Initially, this is done without consideration of any multi-level interaction, and this interaction is then enforced by a subdivision operator. It has been detailed how the algorithms can be implemented. Numerical examples illustrate the accuracy of the proposed method. Optimal convergence rates are obtained for all cases. However, the error level for adaptive mesh refinement is generally lower than that for uniform mesh refinement, as to be expected. Upon local mesh refinement, the error in the stress is reduced in the entire domain, which indicates that adaptive mesh refinement also improves the accuracy globally.



(a) error in u at refinement step $s = 10$

(b) error in u at refinement step $s = 22$

Figure 24: Bézier meshes and error in u at each refinement step for quadratic T-spline bases. The error is given as the difference between numerical solution and exact solution.

Acknowledgement

Financial support from the European Research Council (ERC Advanced Grant 664734 PoroFrac) is gratefully acknowledged.

References

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

[2] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, Isogeometric analysis: toward integration of CAD and FEA, John Wiley & Sons, Chichester, 2009.

[3] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, ACM Transactions on Graphics 22 (2003) 477–484.

[4] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, T. Lyche, T-spline simplification and local refinement, ACM Transactions on Graphics 23 (2004) 276–283.

[5] M. A. Scott, X. Li, T. W. Sederberg, T. J. R. Hughes, Local refinement of analysis-suitable T-splines, Computer Methods in Applied Mechanics and Engineering 213 (2012) 206–222.

[6] E. J. Evans, M. A. Scott, X. Li, D. C. Thomas, Hierarchical T-splines: Analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 284 (2015) 1–20.

[7] X. Wei, Y. Zhang, L. Liu, T. J. R. Hughes, Truncated t-splines: Fundamentals and methods, Computer Methods in Applied Mechanics and Engineering 316 (2017) 349–372.

[8] D. D′Angella, S. Kollmannsberger, E. Rank, A. Reali, Multi-level bézier extraction for hierarchical local refinement of isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 328 (2018) 147–174.

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

[10] A. Bressan, Some properties of LR-splines, Computer Aided Geometric Design 30 (2013) 778–794.

[11] K. A. Johannessen, T. Kvamsdal, T. Dokken, Isogeometric analysis using LR B-splines, Computer Methods in Applied Mechanics and Engineering 269 (2014) 471–514.

[12] L. Chen, R. de Borst, Locally refined t-splines, International Journal for Numerical Methods in Engineering (2018) DOI: 10.1002/nme.5759.

[13] A. V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 200 (2011) 3554–3567.

[14] C. Giannelli, B. Jüttler, H. Speleers, THB-splines: The truncated basis for hierarchical splines, Computer Aided Geometric Design 29 (2012) 485–498.

[15] P. Bornemann, F. Cirak, A subdivision-based implementation of the hierarchical b-spline finite element method, Computer Methods in Applied Mechanics and Engineering 253 (2013) 584–598.

[16] A. Buffa, C. Giannelli, Adaptive isogeometric methods with hierarchical splines: error estimator and convergence, Mathematical Models and Methods in Applied Sciences 26 (2016) 1–25.

[17] P. Hennig, S. Müller, M. Kästner, Bézier extraction and adaptive refinement of truncated hierarchical NURBS, Computer Methods in Applied Mechanics and Engineering 305 (2016) 316–339.

[18] E. M. Garau, R. Vázquez, Algorithms for the implementation of adaptive isogeometric methods using hierarchical b-splines, Applied Numerical Mathematics 123 (2018) 58–87.

[19] X. Li, J. Deng, F. Chen, Surface modeling with polynomial splines over hierarchical T-meshes, The Visual Computer 23 (2007) 1027–1033.

[20] 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.

[21] 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, Computer Methods in Applied Mechanics and Engineering 200 (2011) 1892–1908.

[22] R. de Borst, L. Chen, The role of beźier extraction in adaptive isogeometric analysis: Local refinement and hierarchical refinement, International Journal for Numerical Methods in Engineering 113 (6) (2018) 999–1019.

[23] 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, Computer Methods in Applied Mechanics and Engineering 199 (2010) 229–263.

[24] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, T. J. Hughes, Isogeometric finite element data structures based on Bézier extraction of T-splines, International Journal for Numerical Methods in Engineering 88 (2011) 126–156.

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

[26] S. May, J. Vignollet, R. de Borst, The role of the Bézier extraction operator for T-splines of arbitrary degree: linear dependencies, partition of unity property, nesting behaviour and local refinement, International Journal for Numerical Methods in Engineering 103 (2015) 547–581.

[27] P. Morgenstern, D. Peterseim, Analysis-suitable adaptive T-mesh refinement with linear complexity, Computer Aided Geometric Design 34 (2015) 50–66.

[28] M. R. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis by local h-refinement with T-splines, Computer Methods in Applied Mechanics and Engineering 199 (2010) 264–275.

[29] P. Hennig, M. Kästner, P. Morgenstern, D. Peterseim, Adaptive mesh refinement strategies in isogeometric analysis – A computational comparison, Computer Methods in Applied Mechanics and Engineering 316 (2017) 424–448.

[30] L. Chen, E. J. Lingen, R. de Borst, Adaptive hierarchical refinement of nurbs in cohesive fracture analysis, International Journal for Numerical Methods in Engineering 112 (13) (2017) 2151–2173.

[31] G. Beer, B. Marussig, J. Zechner, C. Dünser, T.-P. Fries, Boundary element analysis with trimmed nurbs and a generalized iga approach, arXiv preprint arXiv:1406.3499.

[32] B. Marussig, J. Zechner, G. Beer, T.-P. Fries, Fast isogeometric boundary element method based on independent field approximation, Computer Methods in Applied Mechanics and Engineering 284 (2015) 458–488.

[33] L. Chen, W. Dornisch, S. Klinkel, Hybrid collocation - Galerkin approach for the analysis of surface represented 3D-solids employing SB-FEM, Computer Methods in Applied Mechanics and Engineering 295 (2015) 268–289.

[34] S. P. Timoshenko, J. N. Goodier, Theory of Elasticity, 2nd Edition, McGraw-Hill, New York, NY, 1951.

[35] S. Klinkel, L. Chen, W. Dornisch, A NURBS based hybrid collocation – Galerkin method for the analysis of boundary represented solids, Computer Methods in Applied Mechanics and Engineering 284 (2015) 689–711.

[36] L. Chen, B. Simeon, S. Klinkel, A NURBS based Galerkin approach for the analysis of solids in boundary representation, Computer Methods in Applied Mechanics and Engineering 305 (2016) 777–805.

Appendix A. Examples of anchors of T-spline meshes

Table A lists the local knot vector and the global coordinates of T-spline anchors in Figure 13.

Table A: Local knot vector $\left(\Xi^1, \Xi^2\right)$, homogeneous coordinates $(x_1 w, x_2 w)$ and weights (w) of T-spline anchors in Figure 13; $a = \sqrt{2}/2$, $b = R_{min}$, $c = R_{max}$, $d = (R_{min} + 11R_{max})/12$, $e = (R_{min} + 5R_{max})/6$, $f = (R_{min} + 3R_{max})/4$, $g = (R_{min} + R_{max})/2$, $h = (3R_{min} + R_{max})/4$, $m = (11R_{min} + R_{max})/12$, $n = (5R_{min} + R_{max})/6$.

Anchor	Ξ^1	Ξ^2	$x_1 w$	$x_2 w$	w
1	$\left\{\xi^1_1, \xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_1, \xi^2_2, \xi^2_3\right\}$	0	$-c$	1
2	$\left\{\xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_1, \xi^2_2, \xi^2_3\right\}$	$\frac{ac}{6}$	$-\frac{(a+5)c}{6}$	$\frac{(a+5)}{6}$
3	$\left\{\xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_1, \xi^2_2, \xi^2_3\right\}$	$\frac{(10a+1)c}{24}$	$-\frac{(10a+13)c}{24}$	$\frac{(5a+7)}{12}$
4	$\left\{\xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_1, \xi^2_2, \xi^2_3\right\}$	$\frac{(14a+7)c}{24}$	$-\frac{(14a+3)c}{24}$	$\frac{(7a+5)}{12}$
5	$\left\{\xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6, \xi^1_6\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_1, \xi^2_2, \xi^2_3\right\}$	$\frac{(a+2)c}{3}$	$-\frac{ac}{3}$	$\frac{(a+2)}{3}$
6	$\left\{\xi^1_4, \xi^1_5, \xi^1_6, \xi^1_6, \xi^1_6\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_1, \xi^2_2, \xi^2_3\right\}$	c	0	1
7	$\left\{\xi^1_1, \xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_2, \xi^2_3, \xi^2_4\right\}$	0	$-d$	1
8	$\left\{\xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_2, \xi^2_3, \xi^2_4\right\}$	$\frac{ad}{6}$	$-\frac{(a+5)d}{6}$	$\frac{(a+5)}{6}$
9	$\left\{\xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_2, \xi^2_3, \xi^2_4\right\}$	$\frac{(10a+1)d}{24}$	$-\frac{(10a+13)d}{24}$	$\frac{(5a+7)}{12}$
10	$\left\{\xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_2, \xi^2_3, \xi^2_4\right\}$	$\frac{(14a+7)d}{24}$	$-\frac{(14a+3)d}{24}$	$\frac{(7a+5)}{12}$

Table A: Local knot vector $\left(\mathbf{\Xi}^1, \mathbf{\Xi}^2\right)$, homogeneous coordinates (x_1w, x_2w) and weights (w) of T-spline anchors in Figure 13; $a = \sqrt{2}/2$, $b = R_{min}$, $c = R_{max}$, $d = (R_{min} + 11R_{max})/12$, $e = (R_{min} + 5R_{max})/6$, $f = (R_{min} + 3R_{max})/4$, $g = (R_{min} + R_{max})/2$, $h = (3R_{min} + R_{max})/4$, $m = (11R_{min} + R_{max})/12$, $n = (5R_{min} + R_{max})/6$.

Anchor	$\mathbf{\Xi}^1$	$\mathbf{\Xi}^2$	x_1w	x_2w	w
11	$\left\{\xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6, \xi^1_6\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_2, \xi^2_4, \xi^2_6\right\}$	$\frac{(a+2)e}{3}$	$-\frac{ae}{3}$	$\frac{(a+2)}{3}$
12	$\left\{\xi^1_4, \xi^1_5, \xi^1_6, \xi^1_6, \xi^1_6\right\}$	$\left\{\xi^2_1, \xi^2_1, \xi^2_2, \xi^2_4, \xi^2_6\right\}$	e	0	1
13	$\left\{\xi^1_1, \xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3\right\}$	$\left\{\xi^2_1, \xi^2_2, \xi^2_3, \xi^2_4, \xi^2_5\right\}$	0	$-f$	1
14	$\left\{\xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4\right\}$	$\left\{\xi^2_1, \xi^2_2, \xi^2_3, \xi^2_4, \xi^2_5\right\}$	$\frac{af}{6}$	$-\frac{(a+5)f}{6}$	$\frac{(a+5)}{6}$
15	$\left\{\xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5\right\}$	$\left\{\xi^2_1, \xi^2_2, \xi^2_3, \xi^2_4, \xi^2_5\right\}$	$\frac{(10a+1)f}{24}$	$-\frac{(10a+13)f}{24}$	$\frac{(5a+7)}{12}$
16	$\left\{\xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6\right\}$	$\left\{\xi^2_1, \xi^2_2, \xi^2_3, \xi^2_4, \xi^2_5\right\}$	$\frac{(14a+7)f}{24}$	$-\frac{(14a+3)f}{24}$	$\frac{(7a+5)}{12}$
17	$\left\{\xi^1_1, \xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3\right\}$	$\left\{\xi^2_2, \xi^2_3, \xi^2_4, \xi^2_5, \xi^2_6\right\}$	0	$-g$	1
18	$\left\{\xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4\right\}$	$\left\{\xi^2_2, \xi^2_3, \xi^2_4, \xi^2_5, \xi^2_6\right\}$	$\frac{ag}{6}$	$-\frac{(a+5)g}{6}$	$\frac{(a+5)}{6}$
19	$\left\{\xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5\right\}$	$\left\{\xi^2_2, \xi^2_3, \xi^2_4, \xi^2_5, \xi^2_6\right\}$	$\frac{(10a+1)g}{24}$	$-\frac{(10a+13)g}{24}$	$\frac{(5a+7)}{12}$
20	$\left\{\xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6\right\}$	$\left\{\xi^2_2, \xi^2_3, \xi^2_4, \xi^2_5, \xi^2_6\right\}$	$\frac{(14a+7)g}{24}$	$-\frac{(14a+3)g}{24}$	$\frac{(7a+5)}{12}$
21	$\left\{\xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6, \xi^1_6\right\}$	$\left\{\xi^2_1, \xi^2_2, \xi^2_4, \xi^2_6, \xi^2_7\right\}$	$\frac{(a+2)g}{3}$	$-\frac{ag}{3}$	$\frac{(a+2)}{3}$
22	$\left\{\xi^1_4, \xi^1_5, \xi^1_6, \xi^1_6, \xi^1_6\right\}$	$\left\{\xi^2_1, \xi^2_2, \xi^2_4, \xi^2_6, \xi^2_7\right\}$	g	0	1
23	$\left\{\xi^1_1, \xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3\right\}$	$\left\{\xi^2_3, \xi^2_4, \xi^2_5, \xi^2_6, \xi^2_7\right\}$	0	$-h$	1
24	$\left\{\xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4\right\}$	$\left\{\xi^2_3, \xi^2_4, \xi^2_5, \xi^2_6, \xi^2_7\right\}$	$\frac{ah}{6}$	$-\frac{(a+5)h}{6}$	$\frac{(a+5)}{6}$
25	$\left\{\xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5\right\}$	$\left\{\xi^2_3, \xi^2_4, \xi^2_5, \xi^2_6, \xi^2_7\right\}$	$\frac{(10a+1)h}{24}$	$-\frac{(10a+13)h}{24}$	$\frac{(5a+7)}{12}$
26	$\left\{\xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6\right\}$	$\left\{\xi^2_3, \xi^2_4, \xi^2_5, \xi^2_6, \xi^2_7\right\}$	$\frac{(14a+7)h}{24}$	$-\frac{(14a+3)h}{24}$	$\frac{(7a+5)}{12}$
27	$\left\{\xi^1_1, \xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3\right\}$	$\left\{\xi^2_4, \xi^2_5, \xi^2_6, \xi^2_7, \xi^2_7\right\}$	0	$-m$	1
28	$\left\{\xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4\right\}$	$\left\{\xi^2_4, \xi^2_5, \xi^2_6, \xi^2_7, \xi^2_7\right\}$	$\frac{am}{6}$	$-\frac{(a+5)m}{6}$	$\frac{(a+5)}{6}$
29	$\left\{\xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5\right\}$	$\left\{\xi^2_4, \xi^2_5, \xi^2_6, \xi^2_7, \xi^2_7\right\}$	$\frac{(10a+1)m}{24}$	$-\frac{(10a+13)m}{24}$	$\frac{(5a+7)}{12}$
30	$\left\{\xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6\right\}$	$\left\{\xi^2_4, \xi^2_5, \xi^2_6, \xi^2_7, \xi^2_7\right\}$	$\frac{(14a+7)m}{24}$	$-\frac{(14a+3)m}{24}$	$\frac{(7a+5)}{12}$
31	$\left\{\xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6, \xi^1_6\right\}$	$\left\{\xi^2_2, \xi^2_4, \xi^2_6, \xi^2_7, \xi^2_7\right\}$	$\frac{(a+2)m}{3}$	$-\frac{am}{3}$	$\frac{(a+2)}{3}$
32	$\left\{\xi^1_4, \xi^1_5, \xi^1_6, \xi^1_6, \xi^1_6\right\}$	$\left\{\xi^2_2, \xi^2_4, \xi^2_6, \xi^2_7, \xi^2_7\right\}$	m	0	1
33	$\left\{\xi^1_1, \xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3\right\}$	$\left\{\xi^2_5, \xi^2_6, \xi^2_7, \xi^2_7, \xi^2_7\right\}$	0	$-n$	1
34	$\left\{\xi^1_1, \xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4\right\}$	$\left\{\xi^2_5, \xi^2_6, \xi^2_7, \xi^2_7, \xi^2_7\right\}$	$\frac{an}{6}$	$-\frac{(a+5)n}{6}$	$\frac{(a+5)}{6}$
35	$\left\{\xi^1_1, \xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5\right\}$	$\left\{\xi^2_5, \xi^2_6, \xi^2_7, \xi^2_7, \xi^2_7\right\}$	$\frac{(10a+1)n}{24}$	$-\frac{(10a+13)n}{24}$	$\frac{(5a+7)}{12}$
36	$\left\{\xi^1_2, \xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6\right\}$	$\left\{\xi^2_5, \xi^2_6, \xi^2_7, \xi^2_7, \xi^2_7\right\}$	$\frac{(14a+7)n}{24}$	$-\frac{(14a+3)n}{24}$	$\frac{(7a+5)}{12}$
37	$\left\{\xi^1_3, \xi^1_4, \xi^1_5, \xi^1_6, \xi^1_6\right\}$	$\left\{\xi^2_4, \xi^2_6, \xi^2_7, \xi^2_7, \xi^2_7\right\}$	$\frac{(a+2)n}{3}$	$-\frac{an}{3}$	$\frac{(a+2)}{3}$
38	$\left\{\xi^1_4, \xi^1_5, \xi^1_6, \xi^1_6, \xi^1_6\right\}$	$\left\{\xi^2_4, \xi^2_6, \xi^2_7, \xi^2_7, \xi^2_7\right\}$	n	0	1