Article:
Chen, L. and de Borst, R. orcid.org/0000-0002-3457-3574 (2018) Locally Refined T-splines. International Journal for Numerical Methods in Engineering, 114 (6). pp. 637-659. ISSN 0029-5981

https://doi.org/10.1002/nme.5759

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2017; **00**:1–25
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme

Locally Refined T-splines

L. Chen[1], R. de Borst[1]*

[1] *University of Sheffield, Department of Civil and Structural Engineering, Sheffield S1 3JD, UK*

SUMMARY

We extend Locally Refined (LR) B-splines to Locally Refined T-splines (LR T-splines) within the Bézier extraction framework. This discretisation technique combines the advantages of T-splines to model the geometry of engineering objects exactly with the ability to flexibly carry out local mesh refinement. In contrast to LR B-splines, LR T-splines take a T-mesh as input instead of a tensor-product mesh. The LR T-mesh is defined and examples are given how to construct it from an initial T-mesh by repeated meshline insertions. The properties of LR T-splines are investigated by exploiting the Bézier extraction operator, including the nested nature, linear independence and the partition of unity property. A technique is presented to remove possible linear dependencies between LR T-splines. Like for other spline technologies, the Bézier extraction framework enables to fully use existing finite element datastructures. Copyright © 2017 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: LR T-splines; LR T-mesh; Bézier extraction; partition of unity property; linear dependency

1. INTRODUCTION

In isogeometric analysis (IGA), NURBS basis functions, which are commonly used in the Computer Aided Design (CAD) model, are directly employed in the computational model [1]. The approximation error that that can arise from the geometry description is reduced, or even completely removed, which is unlike when curved surfaces are modelled using standard finite element technology. However, the standard manner to extend B-splines or NURBS, which are essentially one-dimensional concepts, to two or three dimensions is not sufficiently flexible. Indeed, the tensor-product structure which is used to define meshes that use B-splines or NURBS in two or three dimensions, precludes local mesh refinement.

To remove this deficiency, various local refinement strategies have been developed, including T-splines [2–4], hierarchical and truncated T-splines [5–7], LR B-splines [8–10], hierarchical and truncated B-splines [11–14], and PHT-splines [15–17].

Locally Refined (LR) B-splines were introduced by Dokken *et al.* [8] and were subsequently studied in [9, 10, 18, 19]. The basic idea of the LR B-splines is to locally enrich the space of basis functions by replacing coarse grid B-splines by fine grid B-splines, thereby breaking the above mentioned tensor-product structure. Further, an a posteriori error estimator has been proposed [20, 21], which enables to obtain super-convergence rates. Finally, Locally Refined Non-Uniform Rational B-Splines (LR NURBS) have been developed and used in contact analysis as an extension to LR B-splines [22].

*Correspondence to: R. de Borst, University of Sheffield, Department of Civil and Structural Engineering, Sheffield S1 3JD, UK. E-mail: r.deborst@sheffield.ac.uk

LR B-splines and LR NURBS are constructed by meshline insertions into an initial B-spline or NURBS mesh with a tensor-product structure. While more flexible than B-spline or NURBS meshes in the sense that local mesh refinement is possible, an even more flexible representation of the object can be desirable. To achieve such an enhanced flexibility, we extend Locally Refined B-splines to Locally Refined T-splines (LR T-splines), which are directly constructed from an initial T-mesh. To implement the technology into any existing finite element codes, the Bézier extraction framework will be employed. This framework also enables to determine important properties of LR T-splines, like the nested nature, the linear independence and the partition of unity property, and it provides a way to remove possible linear dependencies in a straightforward manner.

After a concise summary of T-splines and Bézier extraction in Section 2 and an introduction into LR-splines, LR T-splines are defined, including the update of anchors after meshline insertions. A central part of the manuscript is the development of a Bézier extraction framework for LR T-splines after subsequent meshline insertions. Examples are given to illustrate how the methodology works. An in-depth discussion of the properties of LR T-splines follows in Section 6.

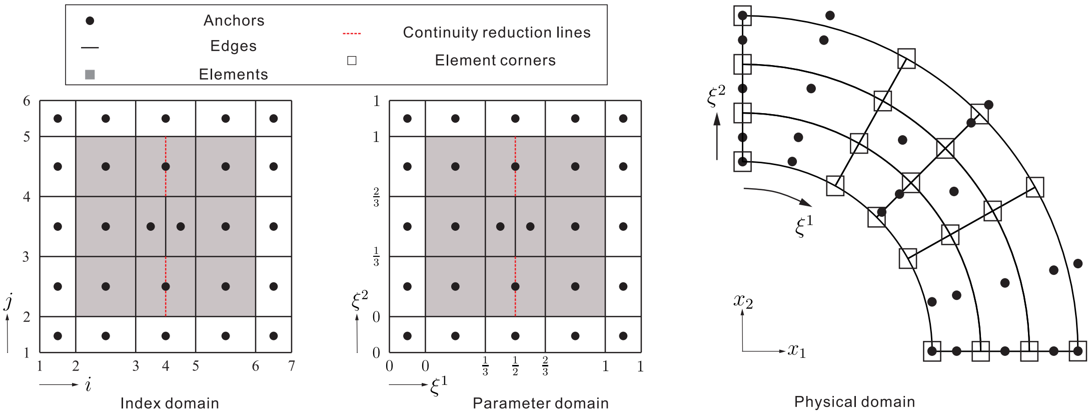


Figure 1. Example of a quadratic T-spline mesh. The object in the index domain $(i,\, j)$, the physical domain $(x_1,\, x_2)$, the parameter domain $\left(\xi^1,\, \xi^2\right)$ is given.

2. T-SPLINES AND BÉZIER EXTRACTION

In this section we review some basic concepts of T-splines and Bézier extraction [23–25].

2.1. *T-splines*

The fundamental object in a T-spline discretisation is the T-mesh, which is a quadrilateral mesh in the physical space. In the T-mesh, T-junctions are allowed to split element edges. An example of a quadratic T-spline mesh is given in Figure 1. Conceptually, T-spines can be considered as a generalisation of B-splines. A univariate B-spline is defined on a knot vector $\Xi = \{\xi_1, \xi_2, \cdots, \xi_{n+p+1}\}$. For standard B-splines we obtain n linearly independent basis functions $N_{i,p}\left(\xi\right)$ of degree p using the Cox-de Boor recursion formula [26, 27]. The extension to the multi-variate case follows in a straightforward manner by exploiting the tensor-product structure.

The concept of constructing basis functions by their local knot vectors is also applicable to T-splines. In a T-mesh, several anchors are prescribed in the index domain and in the parameter domain, Figure 1. For each anchor, the local knot vector is defined by radiating rays in two parametric directions [25]. The intersections on the left and the right are collected to construct the local knot vector of anchor i, Ξ_i $(i = 1, \cdots, n)$. Now, n denotes the number of anchors in a T-mesh. A multi-variate blending function $N_\alpha\left(\xi^1,\, \xi^2\right)$ is attached to each anchor, see Reference [25] for the construction of local knot vectors and blending functions. A T-spline surface $\mathcal{S}\left(\xi^1,\, \xi^2\right)$

DOI: 10.1002/nme

is subsequently constructed by the anchors and the blending functions:

$$\mathcal{S}\left(\xi^1,\,\xi^2\right)=\sum_{\alpha\in\mathcal{A}}\mathbf{P}_\alpha N_\alpha\left(\xi^1,\,\xi^2\right) \tag{1}$$

where $\mathcal{A}$ is the index set of anchors. In a T-mesh the elements are non-zero parametric areas defined by the edges of the T-mesh and the continuity reduction lines, Figure 1.

2.2. *Bézier extraction*

Bézier extraction allows to cast T-spline technology in a standard finite element data structure, while preserving their exact representation [24]. We divide the domain into E elements with n anchors. The local knot vectors of anchor i are $\mathbf{\Xi}_i^1$ and $\mathbf{\Xi}_i^2$. Then, the blending function N_i of anchor i can be expressed as:

$$N_i^e\left(\xi^1,\,\xi^2\right)=[\mathbf{C}_i^e]^{\mathrm{T}}\mathbf{B}^e\left(\xi^1,\,\xi^2\right) \tag{2}$$

over element e. $\mathbf{B}^e\left(\xi^1,\,\xi^2\right)$ contains the element-local Bernstein shape functions with dimension $(p+1)^2\times 1$ [25]. $\mathbf{C}_i^e$ represents the Bézier extraction operator for anchor i over element e [7]. We consider T-splines with the same polynomial degree p in the ξ^1 and ξ^2 parametric directions. Similarly, the Bézier extraction operator for anchor i over E elements can be derived to read:

$$\mathbf{C}_i=\begin{bmatrix}\mathbf{C}_i^1\\ \vdots\\ \mathbf{C}_i^E\end{bmatrix} \tag{3}$$

with $E\left(p+1\right)^2\times 1$ the dimension of $\mathbf{C}_i$. Considering the local support property of N_i the matrix $\mathbf{C}_i$ will have a sparse character.

Subsequently, the Bézier extraction operator of n anchors is also cast in a matrix form:

$$\mathbf{N}\left(\xi^1,\,\xi^2\right)=\mathbf{C}\mathbf{B}\left(\xi^1,\,\xi^2\right)=\begin{bmatrix}N_1\left(\xi^1,\,\xi^2\right)\\ \vdots\\ N_n\left(\xi^1,\,\xi^2\right)\end{bmatrix}=\begin{bmatrix}\mathbf{C}_1^{\mathrm{T}}\\ \vdots\\ \mathbf{C}_n^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\mathbf{B}^1\\ \vdots\\ \mathbf{B}^E\end{bmatrix} \tag{4}$$

in which $\mathbf{C}$ has the dimension $n\times E\left(p+1\right)^2$ and $\mathbf{B}\left(\xi^1,\,\xi^2\right)$ has the dimension $E\left(p+1\right)^2\times 1$. Finally, the blending functions with support over element e are expressed as:

$$\mathbf{N}_e\left(\xi^1,\,\xi^2\right)=\mathbf{C}_e\mathbf{B}_e\left(\xi^1,\,\xi^2\right) \tag{5}$$

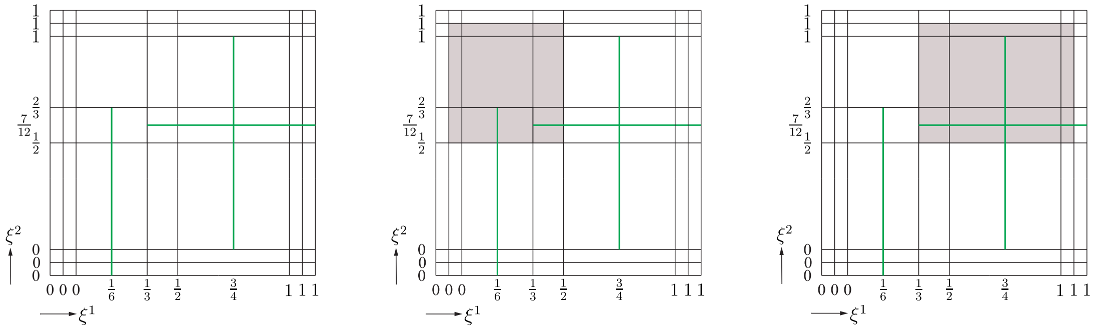
with $\mathbf{C}_e$ the element Bézier extraction operator.

3. FUNDAMENTALS OF LR B-SPLINES

The basic idea of LR B-splines is to locally enrich the space of basis functions by replacing coarse grid B-splines by fine grid B-splines, such that it breaks the tensor-product structure. An LR B-spline surface $\mathcal{S}$ is built by a set of control points $\mathbf{P}_\alpha$ and LR B-spline basis functions:

$$\mathcal{S}\left(\xi^1,\,\xi^2\right)=\sum_{\alpha\in\mathcal{A}}\mathbf{P}_\alpha B_\alpha\left(\xi^1,\,\xi^2\right)\gamma_\alpha \tag{6}$$

with $B_\alpha\left(\xi^1,\,\xi^2\right)$ the LR B-spline basis function, which is computed through the Cox-de Boor formula using local knot vectors [10]. γ_α denotes the scaling weight, which enables the LR B-splines to satisfy the partition of unity property.



(a) LR mesh (b) minimal support (c) no minimal support

Figure 2. Example of an LR mesh in the parameter domain. The green line indicates a meshline insertion into the tensor mesh. (a) LR mesh constructed from a tensor mesh; (b) B-spline defined over the grey area with minimal support; (c) B-spline defined over the grey area without minimal support.

3.1. A locally refined mesh

A Locally Refined mesh (LR mesh) $\mathcal{M}$ is generated by a series of meshline insertions. As stated, this breaks the tensor-product structure and enables a local mesh refinement. Figure 2(a) gives an example. Initially, a quadratic tensor mesh (box mesh) is constructed by open knot vectors in the ξ^1 and ξ^2 parametric directions. Then, green mesh lines, denoted by ε, are inserted into the tensor mesh. They must:

(i) not stop in the centre of an element (knot span);
(ii) insert one line at a time;
(iii) span across at least $p + 2$ knots.

A meshline insertion can be either a new meshline, an elongation of an existing meshline, a joining of two existing meshlines, or increasing the multiplicity of an existing line.

On the basis of an LR mesh $\mathcal{M}$, we can define an LR B-spline space $\mathcal{B}$, if a function B: $\mathbb{R}^2 \to \mathbb{R}$ in $\mathcal{B}$ satisfies:

- $B_{\Xi}\left(\xi^1, \xi^2\right) = \gamma B_{\Xi^1}\left(\xi^1\right) B_{\Xi^2}\left(\xi^2\right)$ is a weighted B-spline.
- B has a minimal support on $\mathcal{M}$, Figures 2(b) and 2(c). Minimal support means that no other meshline traverses the interior of the support of B [10].

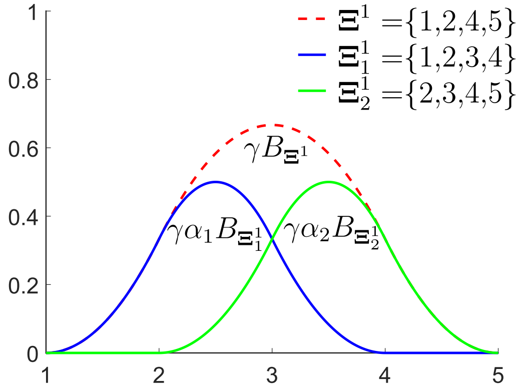


Figure 3. B-spline with the local knot vector Ξ^1 split into two B-splines with respective local knot vectors Ξ^1_1 and Ξ^1_2, by a knot insertion at $\xi^1 = 3$.

DOI: 10.1002/nme

3.2. *Local refinement of B-splines*

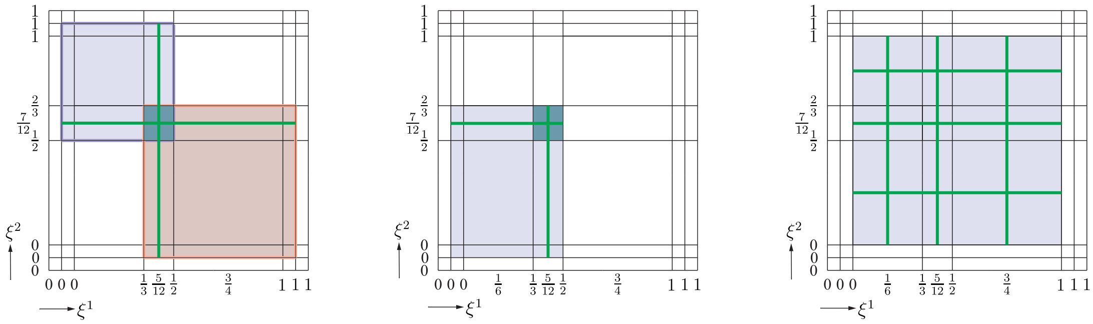
The basic idea of locally refined B-splines is to maintain the minimal support property of LR B-splines after inserting meshlines into an LR mesh $\mathcal{M}$. This refinement is realised by knot insertions in each parametric direction. As an example we take a knot insertion in the ξ^1 parametric direction. We consider an LR B-spline which is defined by a local knot vector Ξ^1. A new knot vector $\hat{\xi}$ is inserted in Ξ^1, yielding two additional local knot vectors Ξ^1_1 and Ξ^1_2:

$$\begin{aligned}
\Xi^1 &= \left[\xi^1_1,\ \xi^1_2,\ \cdots,\ \xi^1_{i-1},\ \phantom{\hat{\xi},}\ \xi^1_i,\ \cdots,\ \xi^1_{p+1},\ \xi^1_{p+2}\right] \\
\Xi^1_1 &= \left[\xi^1_1,\ \xi^1_2,\ \cdots,\ \xi^1_{i-1},\ \hat{\xi},\ \xi^1_i,\ \cdots,\ \xi^1_{p+1}\phantom{,\ \xi^1_{p+2}}\right] \\
\Xi^1_2 &= \left[\ \xi^1_2,\ \cdots,\ \xi^1_{i-1},\ \hat{\xi},\ \xi^1_i,\ \cdots,\ \xi^1_{p+1},\ \xi^1_{p+2}\right]
\end{aligned} \tag{7}$$

The LR B-spline in the ξ^1 parametric direction is then given by:

$$\gamma B_{\Xi^1} = \gamma\alpha_1 B_{\Xi^1_1} + \gamma\alpha_2 B_{\Xi^1_2} \tag{8}$$

where γ is the scaling weight of B_{Ξ^1}, α_1 and α_2 are the refinement operators [10], γB_{Ξ^1} denotes the old LR B-spline, and $\gamma\alpha_1 B_{\Xi^1_1}$ and $\gamma\alpha_2 B_{\Xi^1_2}$ represent the new LR B-splines. Figure 3 illustrates the refinement.



(a) full span

(b) minimum span

(c) structured mesh

Figure 4. Different refinement strategies for LR B-splines (the green lines indicate meshline insertions): (a) full span, which splits all B-splines with support over one element; (b) minimum span, which splits one B-spline with support over one element; (c) structured mesh, which splits all knot spans constructing one B-spline. Options (a) and (b) refine an element, while option (c) refines a B-spline.

There exist different refinement strategies for LR B-splines [10], see Figure 4, which shows the full span and minimal span refinement strategies, which are built on the assumption of element refinement, and structured mesh refinement, which refines the LR B-spline function itself rather than elements. We will come back to the different refinement strategies in Section 5.

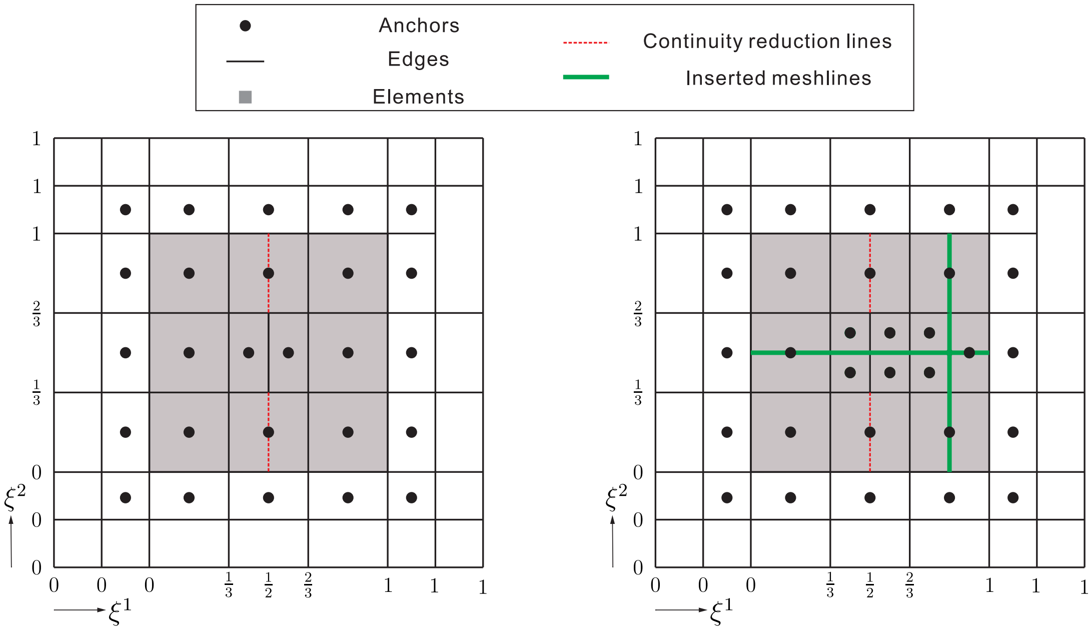
4. LR T-SPLINES

4.1. *Definitions*

The introduction of Locally Refined T-splines (LR T-splines) is now relatively straightforward. We consider an initial T-mesh with n anchors. Each anchor is associated with a local knot vector $\Xi_i\ (i = 1, \cdots, n)$ and a blending function $N_i\left(\xi^1, \xi^2\right)$. We restrict the initial T-mesh to be a structured T-mesh with a nested spline space, such as a semi-standard T-spline mesh.

Definition 1. The Locally Refined T-mesh (LR T-mesh) $\mathcal{T}_n$ is a box mesh which results from a series of single meshline insertion $\{\varepsilon_i\}^n_{i=1}$, starting from an initial T-mesh $\mathcal{T}_0$, i.e. $\mathcal{T}_n \supset \mathcal{T}_{n-1} \supset \cdots \supset \mathcal{T}_1 \supset \mathcal{T}_0$. Each intermediate state $\mathcal{T}_{i+1} = \{\mathcal{T}_i \cup \varepsilon_i\}$ is also a box mesh.

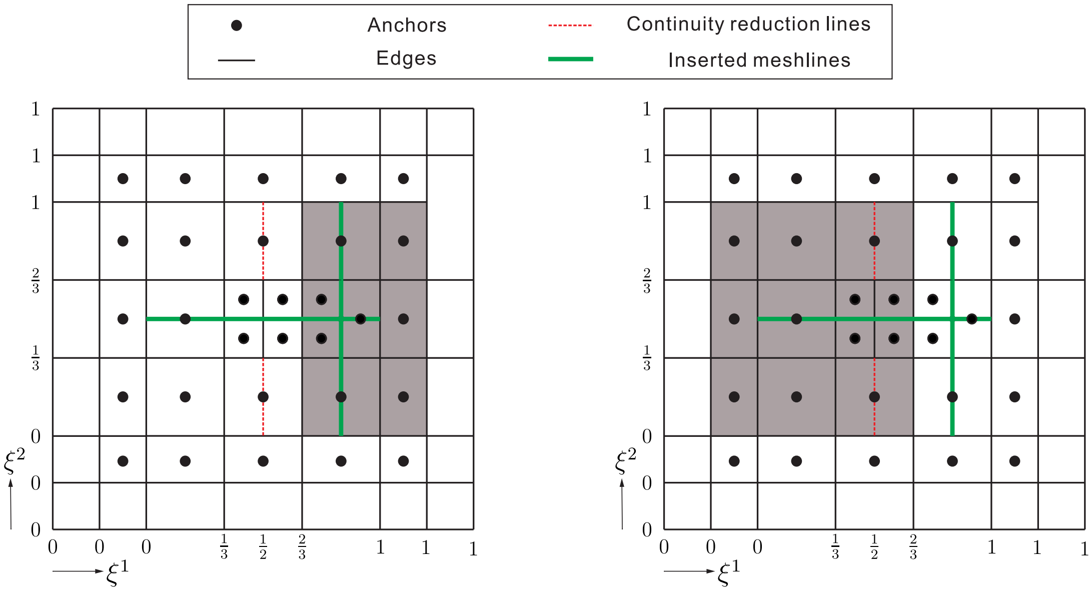
DOI: 10.1002/nme



(a) initial T-mesh $\mathcal{T}_0$

(b) LR T-mesh $\mathcal{T}$

Figure 5. Example of an LR T-mesh in the parameter domain (the green lines indicate meshline insertions in the initial T-mesh).



(a) minimal support

(b) without minimal support

Figure 6. Example of blending functions (defined over the grey area) with and without minimal support on an LR T-mesh $\mathcal{T}$. The green lines indicate meshline insertions into the initial T-mesh $\mathcal{T}_0$.

Definition 2. In an LR T-mesh, elements are non-zero parametric areas defined by the edges of a T-mesh, by continuity reduction lines and by inserted meshlines, Figure 5(b).

Definition 3. $\mathcal{T}$ is an LR T-mesh. A function $N\colon \mathbb{R}^2 \to \mathbb{R}$ is then called an LR T-spline blending function on $\mathcal{T}$ if:

- $N_{\Xi}\left(\xi^1, \xi^2\right) = \gamma N_{\Xi^1}\left(\xi^1\right) N_{\Xi^2}\left(\xi^2\right)$ is a weighted blending function.

- N has a minimal support on $\mathcal{T}$, see Figure 6. Here, the term minimal support implies that there is no other meshline or continuity reduction line that traverses the interior space of $\mathcal{N}$.

The union of LR T-spline functions N is called an LR T-spline space $\mathcal{N}$, $\mathcal{N} = \{N_i : \text{supp}N_i \in \mathcal{T}\}$, $\mathcal{T}$ being the T-mesh in the parameter domain.

Definition 4. A meshline insertion ε in an LR T-mesh $\mathcal{T}_n$ is either

(i) a new meshline,
(ii) an elongation of an existing meshline or a continuity reduction line,
(iii) a joining of two existing meshlines or two existing continuity reduction lines,
(iv) a joining of an existing meshline and a continuity reduction line,
(v) increasing the multiplicity of an existing line, including existing meshlines and continuity reduction lines.

Remark 1: When a meshline insertion is an elongation or a joining of existing meshlines or continuity reduction lines, we use the union of the meshline, the existing meshlines and the continuity reduction lines to perform the LR T-spline splitting.

Remark 2: For an LR T-mesh, the meshline insertions should meet the following requirements:

(i) Pass through an element (knot span),
(ii) Insert one meshline at a time,
(iii) Span across at least $p + 2$ knots.

Remark 3: The initial T-mesh is defined as a rectangular tiling of a region in $\mathbb{R}^2$. Classically, the initial T-mesh is refined by inserting new anchors (vertex T-grid). Herein, we refine this initial T-mesh in the parameter domain by meshline insertions like for the LR B-splines. Thus, the refinement is directly related to the sequence of splines space. T-splines, LR B-splines and LR T-splines are related approaches to local refinement of B-splines. We open up the possibility to start from a T-spline type vertex mesh but driving the LR B-spline type refinement in the parameter domain [8].

For completeness, we also consider Locally Refined Rational T-splines, which are defined by:

$$R_\alpha(\xi) = \frac{w_\alpha N_\alpha\left(\xi^1, \xi^2\right)}{W\left(\xi^1, \xi^2\right)} = \frac{w_\alpha N_\alpha\left(\xi^1, \xi^2\right)}{\sum_{\alpha \in \mathcal{A}} w_\alpha N_\alpha\left(\xi^1, \xi^2\right)} \tag{9}$$

where $N_\alpha\left(\xi^1, \xi^2\right)$ is the standard LR T-spline blending function, $\mathcal{A}$ is the index set of anchors and w_α denotes the weight of anchor α. For LR rational T-splines the LR T-spline surface is defined as:

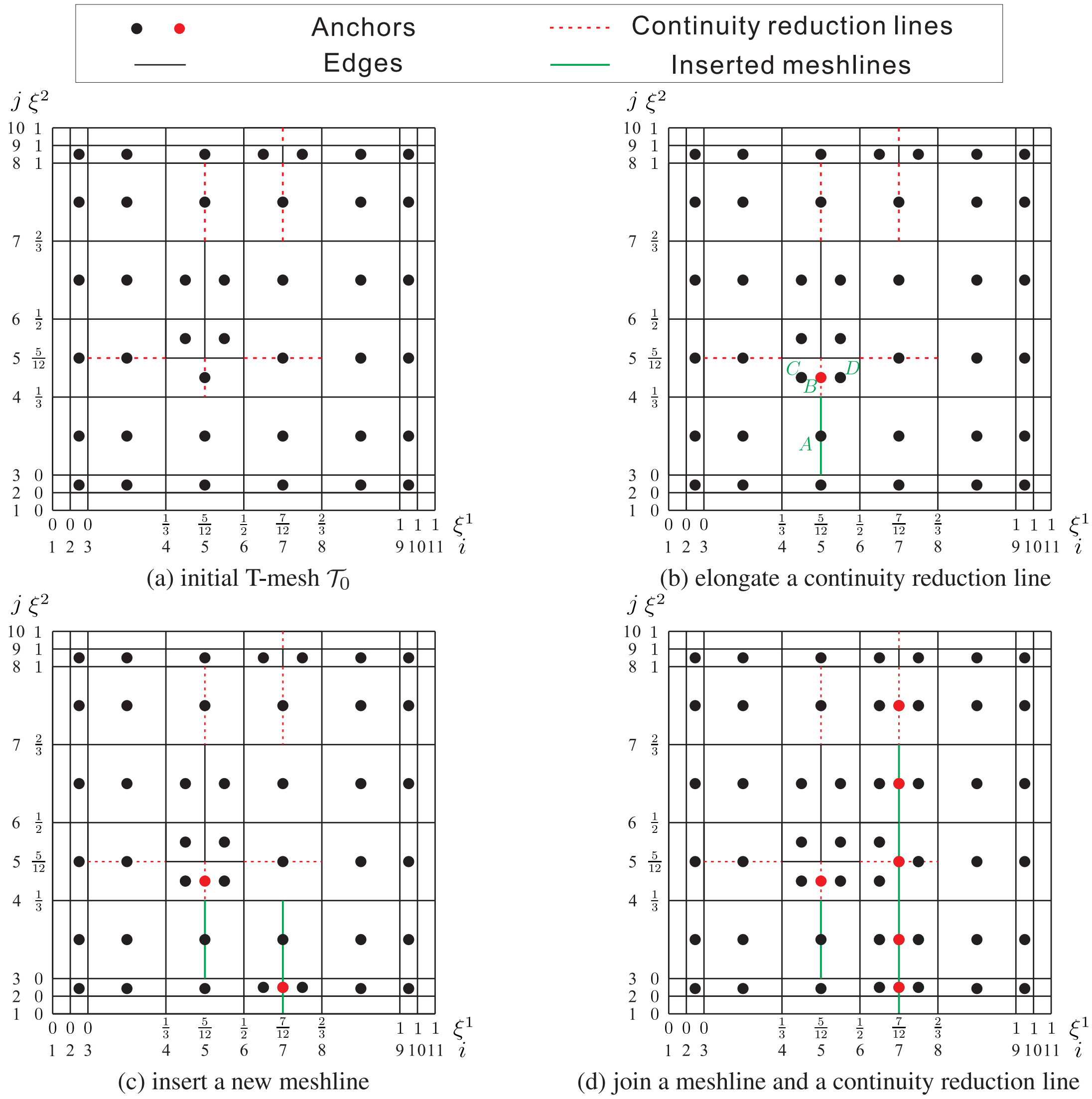
$$\mathcal{S}\left(\xi^1, \xi^2\right) = \sum_{\alpha \in \mathcal{A}} \mathbf{P}_\alpha R_\alpha\left(\xi^1, \xi^2\right) \gamma_\alpha \tag{10}$$

where $\mathbf{P}_\alpha = \left(x^1_\alpha, x^2_\alpha, w_\alpha\right)$ contains the coordinates of anchor α. The weighted coordinates of anchor α are $\mathbf{P}^w_\alpha = \left(w_\alpha x^1_\alpha, w_\alpha x^2_\alpha, w_\alpha\right)$. γ_α represents the scaling weight of $R_\alpha\left(\xi^1, \xi^2\right)$, see Section 5 for details.

4.2. *Update of anchors after meshline insertions*

A meshline is defined by its parameter values [10]. For instance, a horizontal line is defined as: $\varepsilon = \left[\xi^1_\alpha, \xi^1_\beta\right] \times \xi^2_\eta$ and a vertical line as: $\varepsilon = \xi^1_\eta \times \left[\xi^2_\alpha, \xi^2_\beta\right]$. In Figure 7 we give an example of meshline insertions into an LR T-mesh of an even degree. Figure 7(a) displays the initial T-mesh $\mathcal{T}_0$, while Figures 7(b)-(d) represent LR T-meshes $\mathcal{T}$ after meshline insertions. Figure 7(b) represents the case of an elongation of the continuity reduction line $\xi^1_5 \times \left[\xi^2_4, \xi^2_8\right]$ by a meshline $\xi^1_5 \times \left[\xi^2_3, \xi^2_4\right]$. Figure 7(c) indicates the case of a new meshline insertion, $\xi^1_7 \times \left[\xi^2_1, \xi^2_4\right]$. Figure 7(d) finally presents the case of a joining of the existing meshline $\xi^1_7 \times \left[\xi^2_1, \xi^2_4\right]$ and the continuity reduction line $\xi^1_7 \times \left[\xi^2_7, \xi^2_{10}\right]$ by a meshline $\xi^1_7 \times \left[\xi^2_4, \xi^2_7\right]$.

DOI: 10.1002/nme



(a) initial T-mesh $\mathcal{T}_0$

(b) elongate a continuity reduction line

(c) insert a new meshline

(d) join a meshline and a continuity reduction line

Figure 7. Determination of anchors for an LR T-mesh after meshline insertions (quadratic case). The anchors are given in the parameter domain and in the index domain. Anchors deleted after meshline insertions are shown in red.

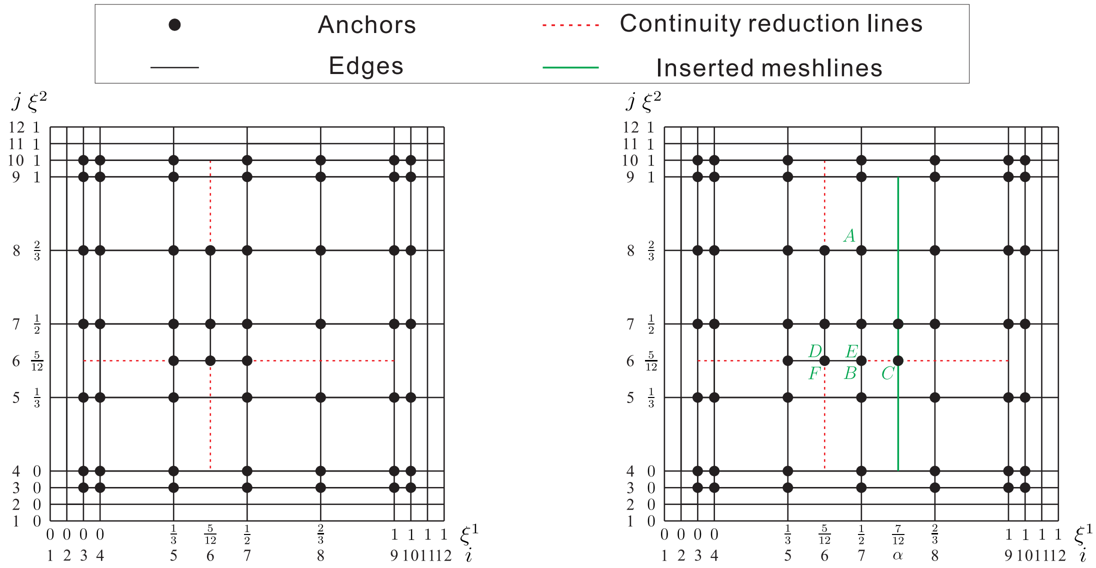
For an LR T-mesh $\mathcal{T}_0$, either of an even degree or of an odd degree, the intermediate LR T-mesh after meshline insertions $\{\varepsilon_i\}_{i=1}^{m}$, is indicated by $\mathcal{T}_{i+1} = \{\mathcal{T}_i \cup \varepsilon_i\}$. The anchors for each intermediate LR T-mesh $\mathcal{T}_{i+1}$ are then updated as follows:

$\mathcal{S}1$ Determine the parameter values of the inserted meshline ε_i: horizontal line, $\varepsilon_i = \left[\xi^1_\alpha,\, \xi^1_\beta\right] \times \xi^2_\eta$; vertical line, $\varepsilon_i = \xi^1_\eta \times \left[\xi^2_\alpha,\, \xi^2_\beta\right]$. The type of meshline insertion is given in **Definition 4**.

$\mathcal{S}2$ Check for the local knot vectors of each anchor whether they are traversed by the meshline ε_i. If so, delete the anchor and split the local knot vector of the anchor by ε_i in order to introduce new anchors, see Figures 7(b) and 8(b).

In Figure 7(b), a vertical meshline is introduced by an elongation of a continuity reduction line: $\varepsilon_i = \xi^1_5 \times \left[\xi^2_3,\, \xi^2_8\right]$. The local knot vectors of anchor A are $\Xi^1_A = \left\{\xi^1_3, \xi^1_4, \xi^1_6,\, \xi^1_8\right\}$ and $\Xi^2_A = \left\{\xi^2_2, \xi^2_3, \xi^2_4,\, \xi^2_5\right\}$. While the meshline ε_i is not included in the interior space confined by Ξ^1_A and Ξ^2_A, anchor A should be kept. This is different for anchor B, with the local knot vectors $\Xi^1_B = \left\{\xi^1_3, \xi^1_4, \xi^1_6,\, \xi^1_8\right\}$ and $\Xi^2_B = \left\{\xi^2_3, \xi^2_4, \xi^2_5,\, \xi^2_6\right\}$. Since the meshline ε_i is included in

DOI: 10.1002/nme



(a) initial cubic T-mesh $\mathcal{T}_0$

(b) inset a new meshline

Figure 8. The update of anchors for a cubic LR T-mesh after a meshline insertion. The anchors are shown in the parameter domain and in the index domain.

the interior space confined by Ξ_B^1 and Ξ_B^2, anchor B must be deleted and the local knot vectors must be split to introduce new anchors, C and D, which have the local knot vectors: $\Xi_C^1 = \{\xi_3^1, \xi_4^1, \xi_5^1, \xi_6^1\}$, $\Xi_C^2 = \{\xi_3^2, \xi_4^2, \xi_5^2, \xi_6^2\}$, and $\Xi_D^1 = \{\xi_4^1, \xi_5^1, \xi_6^1, \xi_8^1\}$, $\Xi_D^2 = \{\xi_3^2, \xi_4^2, \xi_5^2, \xi_6^2\}$, respectively.

In Figure 8(b), a vertical meshline, $\varepsilon_i = \xi_\alpha^1 \times [\xi_4^2, \xi_9^2]$, is inserted. The local knot vectors of anchor A are $\Xi_A^1 = \{\xi_5^1, \xi_6^1, \xi_7^1, \xi_8^1, \xi_9^1\}$ and $\Xi_A^2 = \{\xi_6^2, \xi_7^2, \xi_8^2, \xi_9^2, \xi_{10}^2\}$. The meshline ε_i is not included in the interior space confined by Ξ_A^1 and Ξ_A^2, so that anchor A must be kept. The meshline ε_i is included in the local knot vectors of anchors B and F, and these anchors therefore have to be deleted. New anchors should be introduced: C, D and E. For these anchors, the local knot vectors are: $\Xi_C^1 = \{\xi_6^1, \xi_7^1, \xi_\alpha^1, \xi_8^1, \xi_9^1\}$, and $\Xi_C^2 = \{\xi_4^2, \xi_5^2, \xi_6^2, \xi_7^2, \xi_8^2\}$, $\Xi_D^1 = \{\xi_4^1, \xi_5^1, \xi_6^1, \xi_7^1, \xi_\alpha^1\}$ and $\Xi_D^2 = \{\xi_4^2, \xi_5^2, \xi_6^2, \xi_7^2, \xi_8^2\}$, and $\Xi_E^1 = \{\xi_5^1, \xi_6^1, \xi_7^1, \xi_\alpha^1, \xi_8^1\}$ and $\Xi_E^2 = \{\xi_4^2, \xi_5^2, \xi_6^2, \xi_7^2, \xi_8^2\}$, respectively.

$\mathcal{S}3$ Check the support of each new anchor whether it is completely traversed by an existing meshline or continuity reduction lines. If not, stop the process of updating anchors for the current meshline insertion and insert the next meshline. Otherwise return to $\mathcal{S}2$ and further split the local knot vectors of newly introduced anchors by the traversing meshlines.

5. BÉZIER EXTRACTION OF LR T-SPLINES AFTER MESHLINE INSERTIONS

We will now extend the Bézier extraction framework to allow for the implementation of LR T-splines after meshline insertions.

We consider an LR T-mesh, $\mathcal{T}$, with n anchors. We carry out a series of single meshline insertions $\{\varepsilon_i\}_{i=1}^n$ in $\mathcal{T}$, which yields a new LR T-mesh $\mathcal{T}_r$ with n_r anchors and updated local knot vectors. Consequently, the LR T-splines $\mathcal{N}$ associated with $\mathcal{T}$ can now be described by the LR T-splines $\mathcal{N}_r$ associated with $\mathcal{T}_r$:

$$\mathbf{\Gamma N}\left(\xi^1, \xi^2\right) = \mathbf{\Gamma S N}_r\left(\xi^1, \xi^2\right) \tag{11}$$

with $\mathbf{N}$ and $\mathbf{N}_r$ the LR T-spline blending functions associated with the LR T-meshes $\mathcal{T}$ and $\mathcal{T}_r$, respectively. $\mathbf{S}$ is the refinement operator [7, 28], $\mathbf{\Gamma}$ is a diagonal matrix with the scaling weight γ of $\mathbf{N}$ along the diagonal. This makes it possible that the LR T-splines $\mathcal{N}$ satisfy the partition of unity

DOI: 10.1002/nme

property. The refinement operator $\mathbf{S}$ can be obtained from Equations (4) and (11):

$$\mathbf{\Gamma N} = \mathbf{\Gamma C B}_r = \mathbf{\Gamma S C}_r \mathbf{B}_r \tag{12}$$

where $\mathbf{C}$ denotes the Bézier extraction operator of each anchor on the LR T-mesh $\mathcal{T}$ over the elements on the LR T-mesh $\mathcal{T}_r$. $\mathbf{C}_r$ represents the Bézier extraction operator of each anchor on $\mathcal{T}_r$ over the elements in $\mathcal{T}_r$. $\mathbf{B}_r$ contains the Bernstein polynomials of the elements in $\mathcal{T}_r$.

Expanding Equation (12) in a vector form and eliminating $\mathbf{\Gamma}$ lead to:

$$\begin{bmatrix} \mathbf{C}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{C}_n{}^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{S}_n^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{r1}^{\mathrm{T}} \\ \vdots \\ \mathbf{C}_{rn_r}^{\mathrm{T}} \end{bmatrix} \tag{13}$$

where $\mathbf{C}_i$ denotes the Bézier extraction operator of anchor i in the LR T-mesh $\mathcal{T}$ over the elements on the LR T-mesh $\mathcal{T}_r$, with dimension $E_r\,(p+1)^2 \times 1$. $\mathbf{C}_{ri}$ represents the Bézier extraction operator of anchor i on the LR T-mesh $\mathcal{T}_r$ over the elements in $\mathcal{T}_r$, with dimension $E_r\,(p+1)^2 \times 1$. E_r is the number of elements in $\mathcal{T}_r$. The row values of $\mathbf{S}$ are then obtained from:

$$\mathbf{C}_i = \mathbf{C}_r^T \mathbf{S}_i \qquad \text{for} \qquad i = 1, \cdots, n \tag{14}$$

Next, we define the function

$$f\left(\xi^1, \xi^2\right) = \sum_{i=1}^{n} \gamma_i N_i\left(\xi^1, \xi^2\right) \tag{15}$$

with γ_i the ith diagonal term in $\mathbf{\Gamma}$, and $N_i\left(\xi^1, \xi^2\right)$ denoting the ith LR T-spline blending function associated with the LR T-mesh $\mathcal{T}$. n is the number of anchors on $\mathcal{T}$. Substitution of Equation (11) into (15) yields:

$$\begin{aligned} f\left(\xi^1, \xi^2\right) &= \sum_{i=1}^{n} \gamma_i N_i\left(\xi^1, \xi^2\right) = \sum_{i=1}^{n} \gamma_i \sum_{j=1}^{n_r} S_{ij} N_{rj}\left(\xi^1, \xi^2\right) \\ &= \sum_{j=1}^{n_r} \left(\sum_{i=1}^{n} \gamma_i S_{ij}\right) N_{rj}\left(\xi^1, \xi^2\right) = \sum_{j=1}^{n_r} \gamma_j^r N_{rj}\left(\xi^1, \xi^2\right) \end{aligned} \tag{16}$$

with $N_{rj}\left(\xi^1, \xi^2\right)$ the jth LR T-spline blending function associated with $\mathcal{T}_r$. n_r is the number of anchors on $\mathcal{T}_r$. From the definition of LR T-splines, it is known that γ_j^r denotes the scaling weight of the LR T-spline blending function N_{rj}:

$$\gamma_j^r = \sum_{i=1}^{n} \gamma_i S_{ij} \tag{17}$$

which represents the column summation of the matrix $\mathbf{\Gamma S}$. Using Equation (10), the weighted surface $\mathcal{S}^w$ is then given by:

$$\mathcal{S}^w\left(\xi^1, \xi^2\right) = \sum_{\alpha=1}^{n} \gamma_\alpha N_\alpha\left(\xi^1, \xi^2\right) \mathbf{P}_\alpha^w \tag{18}$$

We require that the weighted surface defined by the LR T-meshes $\mathcal{T}$ and $\mathcal{T}_r$ represents the same geometry:

$$\mathcal{S}^w\left(\xi^1, \xi^2\right) = \mathcal{S}_r^w\left(\xi^1, \xi^2\right) \tag{19}$$

Substitution of Equation (18) into Equation (19) leads to:

$$\sum_{\alpha=1}^{n} \gamma_\alpha N_\alpha\left(\xi^1, \xi^2\right) \mathbf{P}_\alpha^w = \sum_{\beta=1}^{n_r} \gamma_\beta^r N_{r\beta}\left(\xi^1, \xi^2\right) \mathbf{P}_{r\beta}^w \tag{20}$$

where γ_α, $N_\alpha\left(\xi^1, \xi^2\right)$ and $\mathbf{P}^w_\alpha$ are geometrical properties associated with the LR T-mesh $\mathcal{T}$. γ^r_β, $N_{r\beta}\left(\xi^1, \xi^2\right)$ and $\mathbf{P}^w_{r\beta}$ are geometrical properties associated with the LR T-mesh $\mathcal{T}_r$.

Employing the Bézier extraction operator results in:

$$\sum_{\alpha=1}^{n} \gamma_\alpha \mathbf{C}^{\mathrm{T}}_\alpha \mathbf{B}_r \mathbf{P}^w_\alpha = \sum_{\beta=1}^{n_r} \gamma^r_\beta \mathbf{C}^{\mathrm{T}}_{r\beta} \mathbf{B}_r \mathbf{P}^w_{r\beta} \tag{21}$$

and considering the linear independence of Bernstein polynomials $\mathbf{B}_r$, one obtains:

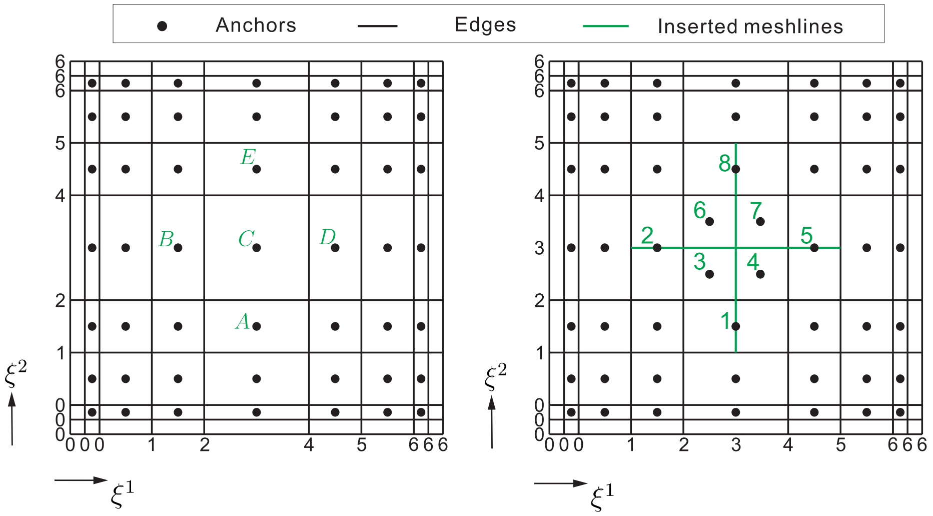
$$\sum_{\alpha=1}^{n} \gamma_\alpha \mathbf{C}^{\mathrm{T}}_\alpha \mathbf{P}^w_\alpha = \sum_{\beta=1}^{n_r} \gamma^r_\beta \mathbf{C}^{\mathrm{T}}_{r\beta} \mathbf{P}^w_{r\beta} \tag{22}$$

Elaborating Equation (22) in a matrix form and considering Equation (12) yield:

$$\mathbf{C}^{\mathrm{T}} \mathbf{\Gamma} \mathbf{P}^w = \mathbf{C}^{\mathrm{T}}_r \mathbf{\Gamma}_r \mathbf{P}^w_r \tag{23}$$

with $\mathbf{C}$ and $\mathbf{C}_r$ defined in Equation (12). $\mathbf{\Gamma}$ is a diagonal matrix with the scaling weight γ_α of the LR T-splines $\mathbf{N}$ along the diagonal, while $\mathbf{\Gamma}_r$ is a diagonal matrix with the scaling weight γ^r_β of the LR T-splines $\mathbf{N}_r$, see Equation (17). $\mathbf{P}^w$ and $\mathbf{P}^w_r$ are column vectors which contain the control points $\mathbf{P}^2_\alpha$ and $\mathbf{P}^w_{r\beta}$, respectively. Using Equations (12), (13) and (23), the following result is obtained:

$$\mathbf{P}^w_r = \mathbf{\Gamma}^{-1}_r \mathbf{S}^{\mathrm{T}} \mathbf{\Gamma} \mathbf{P}^w \tag{24}$$



(a) initial LR B-spine mesh $\mathcal{T}$ (b) LR B-spline mesh after meshline insertions $\mathcal{T}_r$

Figure 9. Inserting meshlines into an LR B-spline mesh. The green lines indicate meshline insertions. The circles denote the anchors associated with the corresponding LR B-spline meshes $\mathcal{T}$ and $\mathcal{T}_r$, respectively.

In the following subsections, we present several examples of the LR T-splines after meshline insertions. The first two examples consider LR B-splines and LR NURBS as a special case of the LR T-splines. The last two examples directly illustrate LR T-splines after meshline insertions. An example of error-guided refinement in isogeometric analysis can be found in [29].

5.1. Example 1: Bézier extraction of LR B-splines after meshline insertions

We first consider an example to illustrate the method, Figure 9 [10]. Noting that B-splines are a special case of T-splines, LR B-splines after meshline insertions can be generated using basis function subdivisions. The scaling weights γ of the initial LR B-splines in Figure 9(a) are equal to one [10]. From Figure 9 the Bézier extraction operators $\mathbf{C}$ and $\mathbf{C}_r$ are obtained, while the refinement operator $\mathbf{S}$ follows from Equation (14). Because $\mathbf{S}$ is rather large, only a sub-matrix of $\mathbf{S}$ is shown below, namely that which is associated with the modified anchors from $\mathcal{T}$ to $\mathcal{T}_r$, Figure 9:

$$\begin{bmatrix} B_A \\ B_B \\ B_C \\ B_D \\ B_E \end{bmatrix} = \begin{bmatrix} 1 & 0 & \frac{2}{9} & \frac{2}{9} & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{2}{9} & 0 & 0 & \frac{2}{9} & 0 & 0 \\ 0 & 0 & \frac{4}{9} & \frac{4}{9} & 0 & \frac{4}{9} & \frac{4}{9} & 0 \\ 0 & 0 & 0 & \frac{2}{9} & 1 & 0 & \frac{2}{9} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{9} & \frac{2}{9} & 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \end{bmatrix} \tag{25}$$

with the LR B-spline functions defined as:

$$\begin{aligned} &B_A = B\,[1245,\,0124]\,; && B_B = B\,[0124,\,1245]\,; && B_C = B\,[1245,\,1245] \\ &B_D = B\,[2456,\,1245]\,; && B_E = B\,[1245,\,2456] && \\ &B_1 = B\,[1245,\,0123]\,; && B_2 = B\,[0123,\,1245]\,; && B_3 = B\,[1234,\,1234] \\ &B_4 = B\,[2345,\,1234]\,; && B_5 = B\,[3456,\,1245]\,; && B_6 = B\,[1234,\,2345] \\ &B_7 = B\,[2345,\,2345]\,; && B_8 = B\,[1245,\,3456] && \end{aligned} \tag{26}$$

The scaling weights of the LR B-spline functions $B_i\,(i = 1, \cdots, 8)$ in Figure 9(b) are obtained from Equation (17):

$$\begin{aligned} &\gamma_1^r = 1; && \gamma_2^r = 1; && \gamma_3^r = \frac{8}{9}; && \gamma_4^r = \frac{8}{9} \\ &\gamma_5^r = 1; && \gamma_6^r = \frac{8}{9}; && \gamma_7^r = \frac{8}{9}; && \gamma_8^r = 1 \end{aligned} \tag{27}$$

It is noted that for the scaling weights of the remaining LR B-spline functions B in Figure 9(b) $\gamma_j^r = 1$. The control points associated with the LR B-spline functions $B_i\,(i = 1, \cdots, 8)$ read, cf. Equation (24):

$$\begin{aligned} &\mathbf{P}_{r1} = \mathbf{P}_A; \mathbf{P}_{r2} = \mathbf{P}_B; \mathbf{P}_{r3} = \frac{1}{8}\left(2\mathbf{P}_A + 2\mathbf{P}_B + 4\mathbf{P}_C\right); \mathbf{P}_{r4} = \frac{1}{8}\left(2\mathbf{P}_A + 4\mathbf{P}_C + 2\mathbf{P}_D\right) \\ &\mathbf{P}_{r5} = \mathbf{P}_D; \mathbf{P}_{r6} = \frac{1}{8}\left(2\mathbf{P}_B + 4\mathbf{P}_C + 2\mathbf{P}_E\right); \mathbf{P}_{r7} = \frac{1}{8}\left(4\mathbf{P}_C + 2\mathbf{P}_D + 2\mathbf{P}_E\right); \mathbf{P}_{r8} = \mathbf{P}_E \end{aligned} \tag{28}$$

the control points associated with the other LR B-spline functions B in Figure 9(b) are unchanged with respect to the original control points in Figure 9(a). The results in Equations (27) and (28) equal those in Reference [10].

5.2. *Example 2: Bézier extraction of LR NURBS after meshline insertions*

Next, we consider the quadratic NURBS surface of Figure (10). This is taken as the initial LR NURBS surface and has been generated by h-refinement of open knot vectors introduced in [7]. The three local refinement strategies of Figure 4 are all employed. The prerequisite for each strategy is that a certain element is marked for refinement.

Figure 11 illustrates the idea of a full span refinement strategy. It refines every NURBS function with support on the element marked in grey. This element is subdivided into four child elements. Moreover, the neighbouring elements are split by a single line, which will render these elements having poor aspect ratios.

The minimum span refinement strategy inserts a cross through the centre of the element marked in grey, Figure 12. The inserted meshline should be as short as possible, but splitting at least one NURBS function. Figure 13 shows the concept of the structured mesh refinement strategy. It inserts a net of meshlines which halve the largest supported knot intervals of a NURBS function.

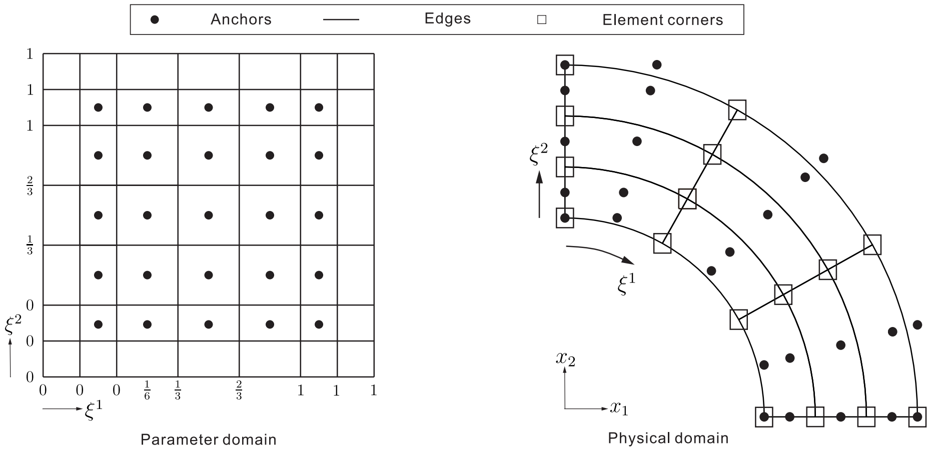


Figure 10. Initial NURBS surface.

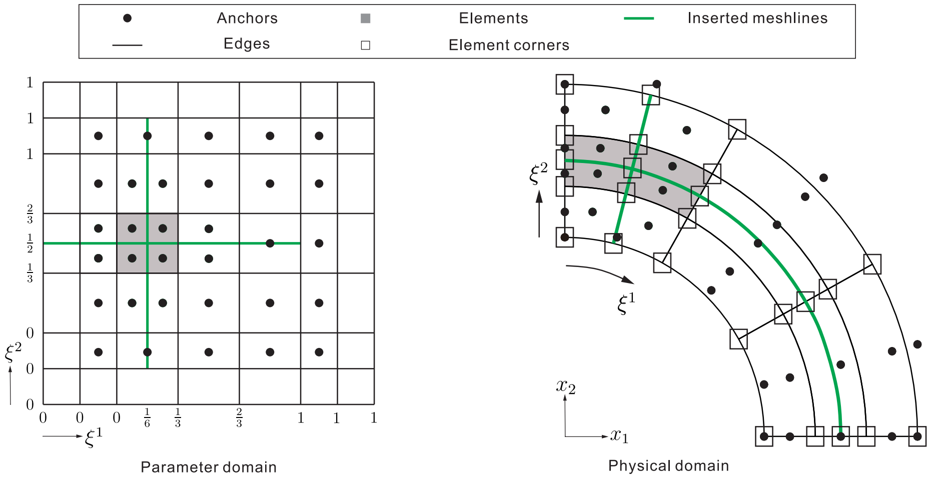


Figure 11. LR NURBS surface generated by the full span refinement strategy.

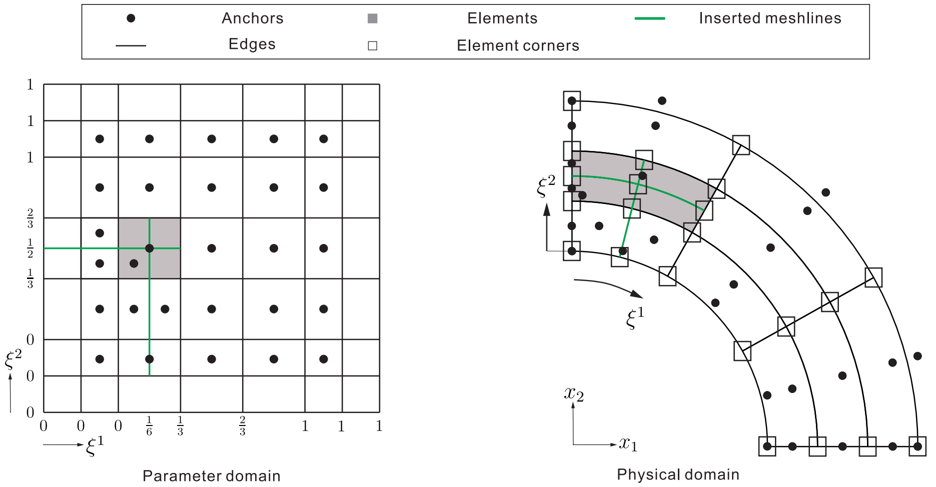


Figure 12. LR NURBS surface generated by the minimum span refinement strategy.

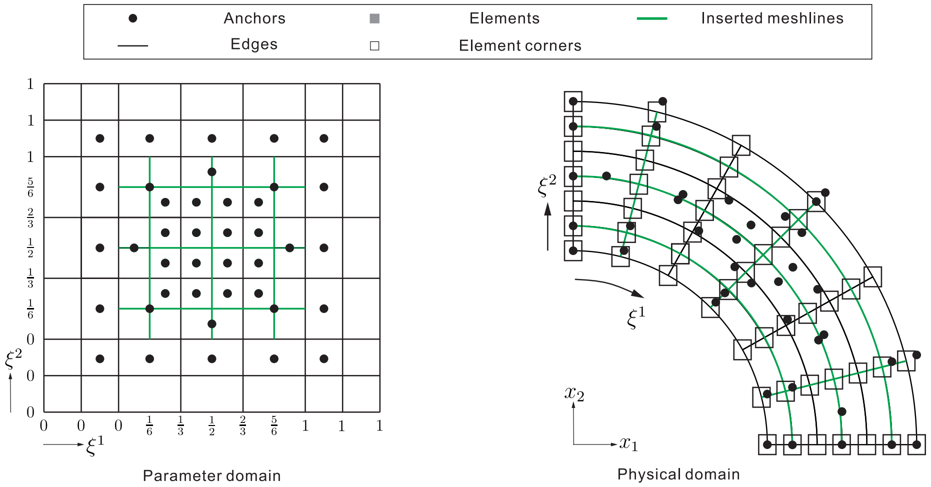


Figure 13. LR NURBS surface generated by the structured mesh refinement strategy.

5.3. Example 3: LR T-mesh with multiplicities

In the third example, an LR T-mesh $\mathcal{T}$ with multiplicities is considered, Figure 14, and is generated from the initial T-mesh of Figure 1. The mesh represents a discontinuous interface at $\xi^2 = 1/2$ in elements $e1$ and $e2$. To produce such interface, $\mathcal{C}^{-1}$ continuous blending functions have to be created, which is achieved by using knot lines of multiplicity $m = p + 1$ [28]. In Figure 14(a), the number of knot lines at $\xi^2 = 1/2$ is $m = 2 + 1 = 3$. Due to this discontinuity, the coordinates of control points $1, \cdots, 4$ equal those of the control points $5, \cdots, 8$, Figures 14(a) and 14(b). To better illustrate the discontinuity in Figure 14(c) a shift $\left(\delta^1, \delta^2\right)$ has artificially been applied to the control points $1, \cdots, 8$:

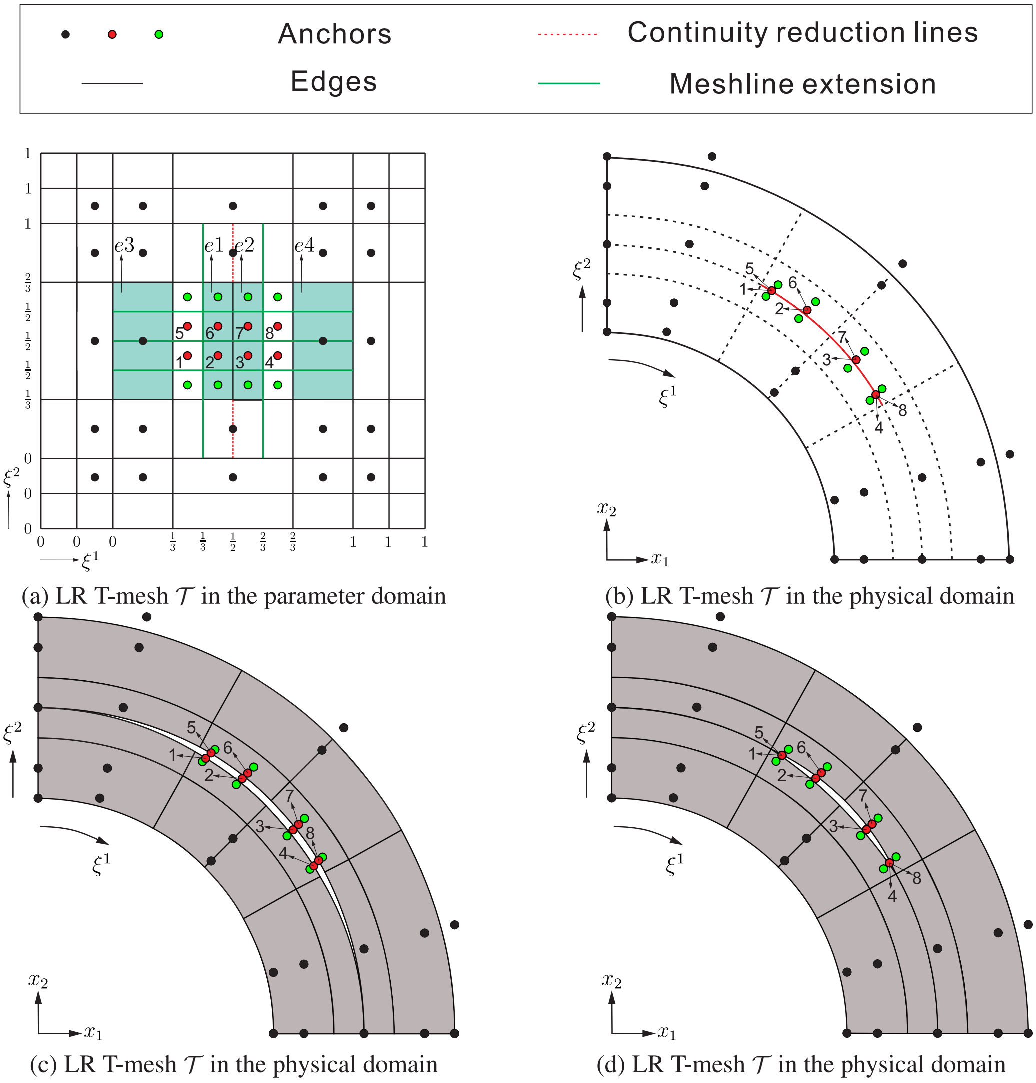
$$\begin{array}{lll} \left(\delta^1_1, \delta^2_1\right) = (-0.15, -0.15) & \left(\delta^1_2, \delta^2_2\right) = (-0.15, -0.15) & \left(\delta^1_3, \delta^2_3\right) = (-0.15, -0.15) \\ \left(\delta^1_4, \delta^2_4\right) = (-0.15, -0.15) & \left(\delta^1_5, \delta^2_5\right) = (0.15, 0.15) & \left(\delta^1_6, \delta^2_6\right) = (0.15, 0.15) \\ \left(\delta^1_7, \delta^2_7\right) = (0.15, 0.15) & \left(\delta^1_8, \delta^2_8\right) = (0.15, 0.15) & \end{array} \tag{29}$$

Figure 14(c) shows that a crack passes through the element boundary at $\xi^2 = 1/2$. This crack not only separates the elements $e1$ and $e2$, but also the elements $e3$ and $e4$. In the analysis of crack propagation, crack opening will be enforced only in the elements $e1$ and $e2$, Figure 14(d). This is achieved by applying the following relation to the control points $1, \cdots, 8$:

$$\begin{array}{lll} \left(\delta^1_1, \delta^2_1\right) = \left(\delta^1_5, \delta^2_5\right) = (0, 0) & \left(\delta^1_2, \delta^2_2\right) = (-0.15, -0.15) & \left(\delta^1_3, \delta^2_3\right) = (-0.15, -0.15) \\ \left(\delta^1_4, \delta^2_4\right) = \left(\delta^1_8, \delta^2_8\right) = (0, 0) & \left(\delta^1_6, \delta^2_6\right) = (0.15, 0.15) & \left(\delta^1_7, \delta^2_7\right) = (0.15, 0.15) \end{array} \tag{30}$$

5.4. Example 4: LR T-mesh with diagonal refinement

In the final example, we consider diagonal refinement of a cubic LR T-mesh. The initial T-mesh, $\mathcal{T}_0$, in the parameter domain and the physical domain is shown in Figures 15(a) and 15(b). The latter figure also shows the anchors on $\mathcal{T}_0$. The refinement is carried out for the diagonal elements and the full span refinement strategy has been used. Figures 15(c) and 15(d) present the results of the third step of the refinement process. The refinement has clearly been carried out in a band along the diagonal. Figure 15(d) illustrates the LR T-mesh $\mathcal{T}$ in the physical domain. It shows that more elements have been created along the diagonal, which leads to a high resolution in this area.



(a) LR T-mesh $\mathcal{T}$ in the parameter domain

(b) LR T-mesh $\mathcal{T}$ in the physical domain

(c) LR T-mesh $\mathcal{T}$ in the physical domain

(d) LR T-mesh $\mathcal{T}$ in the physical domain

Figure 14. Example of an LR T-mesh $\mathcal{T}$ with multiplicities.

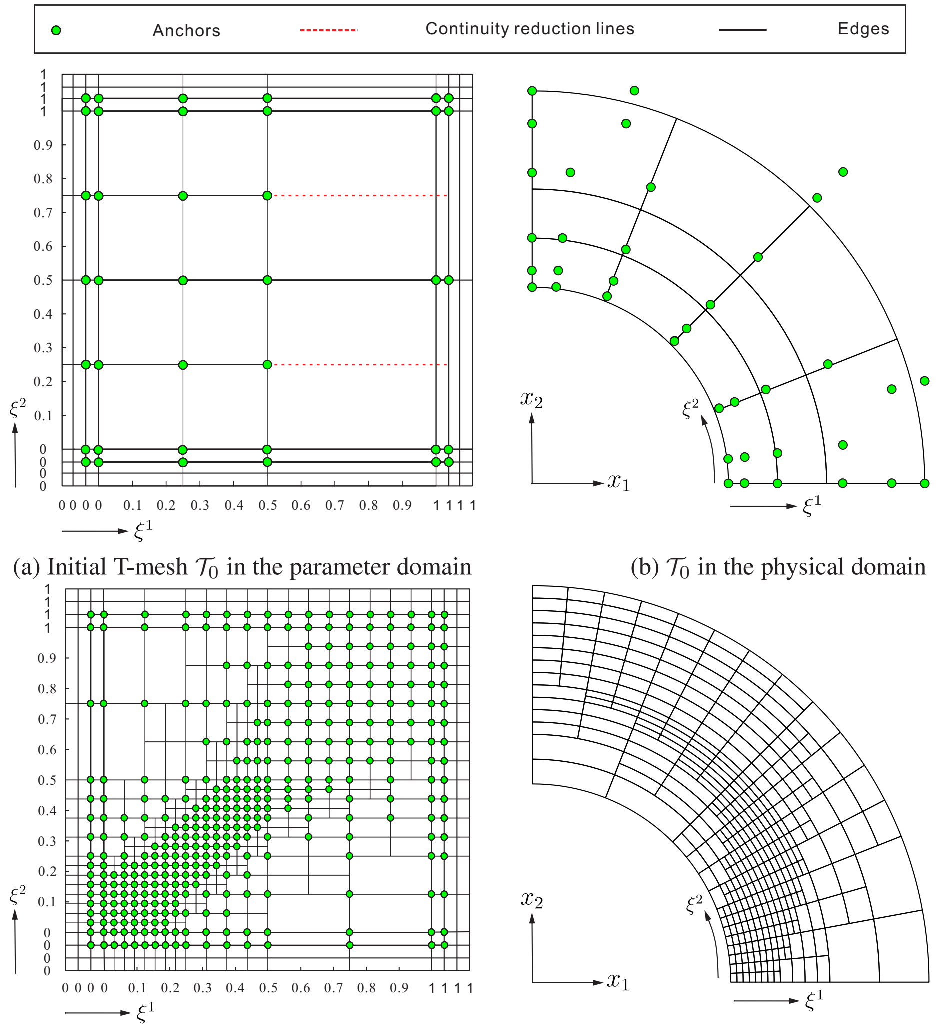
6. PROPERTIES OF LR T-SPLINES

Now, we will investigate some properties of LR T-splines, in particular the partition of unity property, the nested nature and linear dependence. The proofs are carried out by exploiting the Bézier extraction framework. Furthermore, we will propose a new approach to remove the linear dependencies of LR T-splines by the means of Bézier extraction operation.

6.1. *Partition of unity property*

We consider an initial LR T-mesh, $\mathcal{T}$, with n anchors. The LR T-splines $\mathcal{N}$ associated with $\mathcal{T}$ satisfy the partition of unity property, $\mathcal{N} = \{N_j : \text{supp}N_j \subseteq \mathcal{T}\}$. When we carry out a series of single meshline insertions $\{\varepsilon_i\}_{i=1}^{n}$ into $\mathcal{T}$, a new LR T-mesh, $\mathcal{T}_r$, is generated with n_r anchors and a corresponding LR T-spline space $\mathcal{N}_r$. We will prove that $\mathcal{N}_r$ also satisfies the partition of unity property. Rewriting Equation (15) using the Bézier extraction operator yields:

$$f\left(\xi^1, \xi^2\right) = \sum_{i=1}^{n} \gamma_i N_i\left(\xi^1, \xi^2\right) = \sum_{\alpha=1}^{n} \gamma_\alpha \mathbf{C}_\alpha^{\mathrm{T}} \mathbf{B}_r = 1 \tag{31}$$



(a) Initial T-mesh $\mathcal{T}_0$ in the parameter domain

(b) $\mathcal{T}_0$ in the physical domain

(c) LR T-mesh $\mathcal{T}$ of the third refinement

(d) $\mathcal{T}$ in the physical domain

Figure 15. Example of a cubic LR T-mesh $\mathcal{T}$ with diagonal refinement. The full span refinement strategy is employed with single knot lines.

in which $\mathbf{C}_\alpha$ denotes the Bézier extraction operator of each anchor in the LR T-mesh $\mathcal{T}$ over the elements in the LR T-mesh $\mathcal{T}_r$. $\mathbf{B}_r\left(\xi^1, \xi^2\right)$ is the Bernstein polynomial over the elements in $\mathcal{T}_r$.

Substitution of Equations (3) and (4) into Equation (31) leads to:

$$f\left(\xi^1, \xi^2\right) = \left(\gamma_1[\mathbf{C}_1^1]^{\mathrm{T}} + \cdots + \gamma_n[\mathbf{C}_n^1]^{\mathrm{T}}\right)\mathbf{B}^1 + \cdots + \left(\gamma_1[\mathbf{C}_1^E]^{\mathrm{T}} + \cdots + \gamma_n[\mathbf{C}_n^E]^{\mathrm{T}}\right)\mathbf{B}^E \qquad (32)$$

We now rewrite Equation (32) in a matrix form,

$$f\left(\xi^1, \xi^2\right) = \begin{bmatrix}[\mathbf{B}^1]^{\mathrm{T}} & \cdots & [\mathbf{B}^E]^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\gamma_1\mathbf{C}_1^1 + \cdots + \gamma_n\mathbf{C}_n^1 \\ \vdots \\ \gamma_1\mathbf{C}_1^E + \cdots + \gamma_n\mathbf{C}_n^E\end{bmatrix} \qquad (33)$$

or, equivalently,

$$f\left(\xi^1, \xi^2\right) = \begin{bmatrix}[\mathbf{B}^1]^{\mathrm{T}} & \cdots & [\mathbf{B}^E]^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\mathbf{C}_1 & \cdots & \mathbf{C}_n\end{bmatrix}\begin{bmatrix}\gamma_1 \\ \vdots \\ \gamma_n\end{bmatrix} \qquad (34)$$

In view of Equation (13), Equation (34) can also be expressed as:

$$f\left(\xi^1,\,\xi^2\right) = \left[[\mathbf{B}^1]^{\mathrm{T}} \quad \cdots \quad [\mathbf{B}^E]^{\mathrm{T}}\right]\left[\mathbf{C}_{r1} \quad \cdots \quad \mathbf{C}_{rn_r}\right]\left[\mathbf{S}_1 \quad \cdots \quad \mathbf{S}_n\right]\begin{bmatrix}\gamma_1\\ \vdots\\ \gamma_n\end{bmatrix} \tag{35}$$

where $\mathbf{C}_{rj}\,(j = 1,\,\cdots,\,n_r)$ is the Bézier extraction operator of each anchor in the LR T-mesh $\mathcal{T}_r$ over the elements in $\mathcal{T}_r$. Using Equation (17), Equation (35) can subsequently be formulated as:

$$f\left(\xi^1,\,\xi^2\right) = \left[[\mathbf{B}^1]^{\mathrm{T}} \quad \cdots \quad [\mathbf{B}^E]^{\mathrm{T}}\right]\left[\mathbf{C}_{r1} \quad \cdots \quad \mathbf{C}_{rn_r}\right]\begin{bmatrix}\gamma_1^r\\ \vdots\\ \gamma_{n_r}^r\end{bmatrix} \tag{36}$$

where $\gamma_j^r\,(j = 1,\,\cdots,\,n_r)$ is the scaling weight of the LR T-spline blending function N_{rj}. We rewrite Equation (36) as:

$$f\left(\xi^1,\,\xi^2\right) = \left(\gamma_1^r[\mathbf{C}_{r1}^1]^{\mathrm{T}} + \cdots + \gamma_{n_r}^r[\mathbf{C}_{rn_r}^1]^{\mathrm{T}}\right)\mathbf{B}^1 + \cdots + \left(\gamma_1^r[\mathbf{C}_{r1}^E]^{\mathrm{T}} + \cdots + \gamma_{n_r}^r[\mathbf{C}_{rn_r}^E]^{\mathrm{T}}\right)\mathbf{B}^E \tag{37}$$

Using Equations (3), (4) and (31) then leads to:

$$f\left(\xi^1,\,\xi^2\right) = \sum_{i=1}^{n_r}\gamma_i^r N_{ri}\left(\xi^1,\,\xi^2\right) = 1 \tag{38}$$

which shows that LR T-splines generated by a series of meshline insertions also satisfy the partition of unity property.

6.2. *Nested space*

An LR T-spline space will be nested, i.e. $\mathcal{N}_{i-1} \subset \mathcal{N}_i$, if and only if for any $f \in \mathcal{N}_{i-1}$ there exists an $\hat{f} \in \mathcal{N}_i$ such that $f = \hat{f}$. Here, $\mathcal{N}_{i-1}$ and $\mathcal{N}_i$ are the LR T-spline spaces associated with the LR T-meshes $\mathcal{T}_{i-1}$ and $\mathcal{T}_i$, respectively. $\mathcal{T}_i$ is generated by meshline insertions in $\mathcal{T}_{i-1}$.

We define the functions f and $\hat{f}$ by their control points:

$$f = \sum_{k=1}^{n} N_{i-1}^k\left(\xi^1,\,\xi^2\right)p_{i-1}^k \qquad\qquad \hat{f} = \sum_{j=1}^{m} N_i^j\left(\xi^1,\,\xi^2\right)p_i^j \tag{39}$$

in which N_{i-1}^k and p_{i-1}^k are geometrical properties associated with the LR T-spline space $\mathcal{N}_{i-1}$, while N_i^j and p_i^j are those associated with $\mathcal{N}_i$. n and m are the number of anchors defined on the LR T-meshes $\mathcal{T}_{i-1}$ and $\mathcal{T}_i$, respectively.

To obtain the relation between an arbitrary $f = \mathbf{p}_{i-1}$ and $\hat{f} = \mathbf{p}_i$, Equation (11) is employed, leading to:

$$\mathbf{N}_{i-1}\left(\xi^1,\,\xi^2\right) = \mathbf{S}\mathbf{N}_i\left(\xi^1,\,\xi^2\right) \qquad\qquad \mathbf{p}_i = \mathbf{S}^{\mathrm{T}}\mathbf{p}_{i-1} \tag{40}$$

with

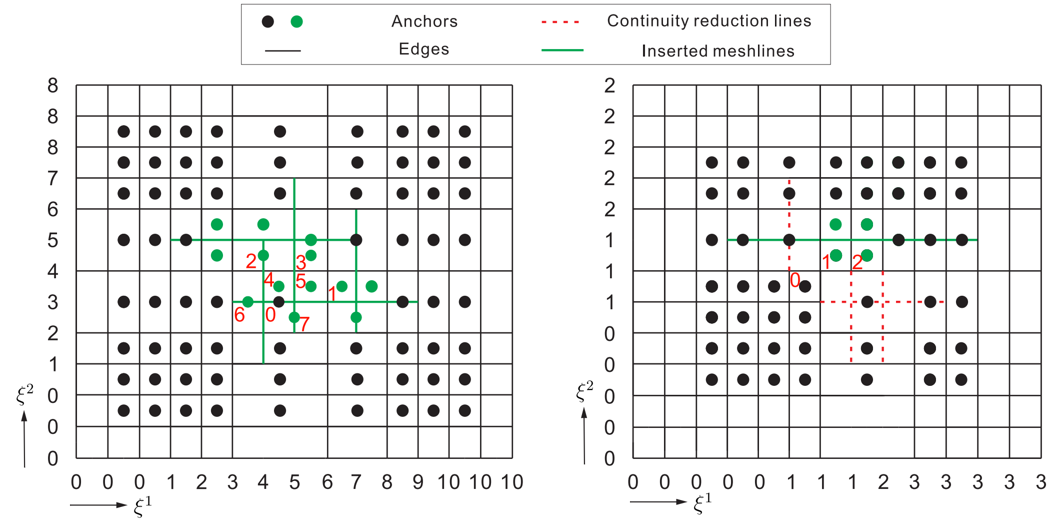
$$\begin{aligned}\mathbf{p}_{i-1} &= \left[p_{i-1}^1,\,\cdots,\,p_{i-1}^n\right]^{\mathrm{T}} \qquad\qquad & \mathbf{p}_i &= \left[p_i^1,\,\cdots,\,p_i^m\right]^{\mathrm{T}}\\ \mathbf{N}_{i-1} &= \left[N_{i-1}^1,\,\cdots,\,N_{i-1}^n\right]^{\mathrm{T}} & \mathbf{N}_i &= \left[N_i^1,\,\cdots,\,N_i^m\right]^{\mathrm{T}}\end{aligned} \tag{41}$$

Accordingly, the LR T-meshes are nested by construction.

6.3. *Linear dependence of LR T-splines*

LR T-splines can be linearly dependent, and we take the example of Reference [10] to illustrate this, see also Figure 16(a). The linear relation is given by:

$$720B_0 = 108B_1 + 135B_2 + 108B_3 + 268B_4 + 324B_5 + 360B_6 + 383B_7 \tag{42}$$



(a) Linearly dependent quadratic LR B-spline mesh

(b) Linearly dependent quartic LR T-mesh

Figure 16. Example of globally linearly dependent LR meshes.

where B represents the LR B-spline function [10]. In general, it is more difficult to derive such a relation for LR T-splines, and therefore, to check possible linear dependencies after meshline insertions. Herein, we will first present a method for verifying linear independence of LR T-splines on the basis of the Bézier extraction operator. Then, a solution for removing linear dependencies will be described. In the remainder we will distinguish between two categories of linear independence:

- globally linearly independent,
- locally linearly independent.

6.3.1. Global linear independence An LR T-mesh $\mathcal{T}$ with n anchors has globally linearly independent LR T-splines $\mathcal{N}$ if and only if the solution for

$$\sum_{i=1}^{n} \alpha_i N_i(\xi^1, \xi^2) = 0 \tag{43}$$

is $\alpha_i = 0$ for $i = 1, \cdots, n$. Considering the Bézier extraction operator this leads to:

$$\sum_{i=1}^{n} \alpha_i \mathbf{C}_\alpha^{\mathrm{T}} \mathbf{B} = 0 \tag{44}$$

where $\mathbf{C}_\alpha$ denotes the Bézier extraction operator of each anchor in $\mathcal{T}$ over the elements in $\mathcal{T}$ and $\mathbf{B}$ is the Bernstein polynomial over the elements in $\mathcal{T}$. Because of the linear independence of $\mathbf{B}$, Equation (44) can be replaced by:

$$\sum_{i=1}^{n} \alpha_i \mathbf{C}_\alpha^{T} = \mathbf{0} \tag{45}$$

Writing Equation (45) in a matrix form then leads to:

$$\begin{bmatrix} \mathbf{C}_1 & \cdots & \mathbf{C}_n \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{46}$$

with equivalence to

$$\mathbf{C}^{\mathrm{T}} \boldsymbol{\alpha} = \mathbf{0} \tag{47}$$

Because an LR T-mesh $\mathcal{T}$ has globally linearly independent LR T-splines $\mathcal{N}$ when $\boldsymbol{\alpha} = \mathbf{0}$ is the only solution for Equation (47), this condition can be checked directly by the rank inspection of $\mathbf{C}$ in Equation (47). If $\mathbf{C}$ has full rank and the rank is equal to n, the LR T-splines $\mathcal{N}$ are globally linearly independent. Otherwise, $\mathcal{N}$ is globally linearly dependent. Hence, the condition for global linear independence is:

$$\text{rank}\,(\mathbf{C}) = n \tag{48}$$

Conversely, if an LR T-mesh is globally linearly dependent, the dependences between anchors can be detected by transforming Equation (47) into a row echelon form by the Gaussian elimination. As an example we take the quartic LR T-mesh $\mathcal{T}$ of Figure 16(b). There are 51 anchors defined over $\mathcal{T}$. Applying Equation (4) to obtain the Bézier extraction operator $\mathbf{C}$, the rank of $\mathbf{C}$ is $\text{rank}\,(\mathbf{C}) = 50$, which shows that the LR T-splines defined over $\mathcal{T}$ are globally linearly dependent. The linear dependencies between the anchors can be obtained by the Gaussian elimination:

$$3N_0\left(\xi^1, \xi^2\right) = 2N_1\left(\xi^1, \xi^2\right) + N_2\left(\xi^1, \xi^2\right) \tag{49}$$

where anchors 0, 1 and 2 have the Bézier extraction operators:

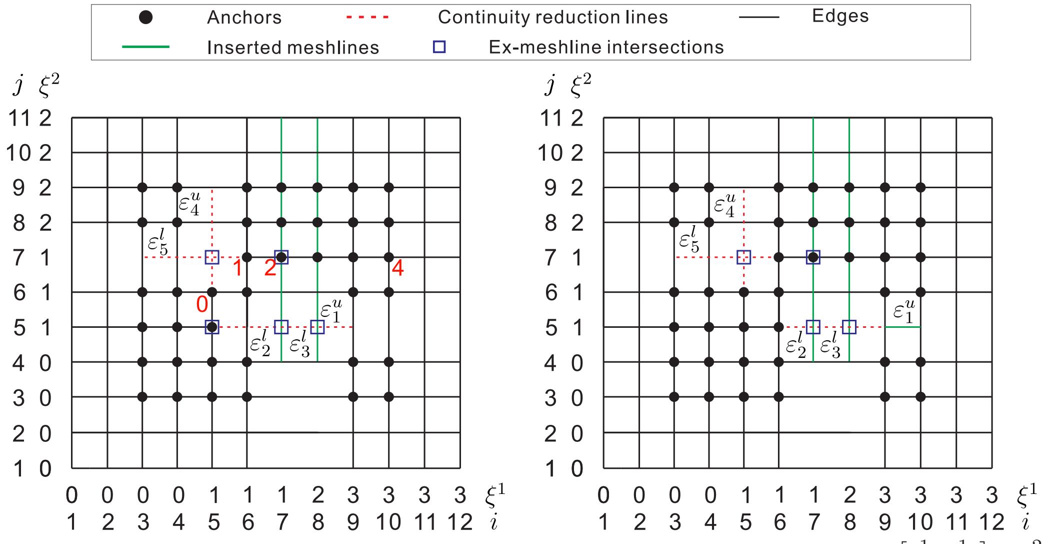
$$\begin{bmatrix} \mathbf{C}_0^{\mathrm{T}} \\ \mathbf{C}_1^{\mathrm{T}} \\ \mathbf{C}_2^{\mathrm{T}} \end{bmatrix} = \left[\begin{array}{cccccccccccc}
\frac{1}{6} & \frac{2}{9} & \frac{1}{6} & \frac{2}{9} & \frac{2}{9} & \frac{5}{18} & \frac{2}{9} & \frac{11}{72} & \frac{7}{72} & \frac{2}{9} & \frac{5}{18} & \frac{2}{9} \\
\frac{1}{4} & \frac{7}{24} & \frac{1}{4} & \frac{7}{24} & \frac{7}{24} & \frac{1}{3} & \frac{1}{6} & \frac{1}{12} & \frac{1}{24} & \frac{7}{24} & \frac{1}{3} & \frac{1}{6} \\
0 & \frac{1}{12} & 0 & \frac{1}{12} & \frac{1}{12} & \frac{1}{6} & \frac{1}{3} & \frac{7}{24} & \frac{5}{24} & \frac{1}{12} & \frac{1}{6} & \frac{1}{3} \\
\frac{11}{72} & \frac{7}{72} & \frac{7}{72} & \frac{1}{24} & \frac{7}{72} & \frac{1}{24} & \frac{1}{6} & \frac{2}{9} & \frac{1}{6} & \frac{2}{9} & \frac{2}{9} & \frac{5}{18} \\
\frac{1}{12} & \frac{1}{24} & \frac{1}{24} & 0 & \frac{1}{24} & 0 & \frac{1}{4} & \frac{7}{24} & \frac{1}{4} & \frac{7}{24} & \frac{7}{24} & \frac{1}{3} \\
\frac{7}{24} & \frac{5}{24} & \frac{5}{24} & \frac{1}{8} & \frac{5}{24} & \frac{1}{8} & 0 & \frac{1}{12} & 0 & \frac{1}{12} & \frac{1}{12} & \frac{1}{6} \\
\frac{2}{9} & \frac{11}{72} & \frac{7}{72} & \frac{2}{9} & \frac{5}{18} & \frac{2}{9} & \frac{11}{72} & \frac{7}{72} & \frac{7}{72} & \frac{1}{24} & \frac{7}{72} & \frac{1}{24} \\
\frac{1}{6} & \frac{1}{12} & \frac{1}{24} & \frac{7}{24} & \frac{1}{3} & \frac{1}{6} & \frac{1}{12} & \frac{1}{24} & \frac{1}{24} & 0 & \frac{1}{24} & 0 \\
\frac{1}{3} & \frac{7}{24} & \frac{5}{24} & \frac{1}{12} & \frac{1}{6} & \frac{1}{3} & \frac{7}{24} & \frac{5}{24} & \frac{5}{24} & \frac{1}{8} & \frac{5}{24} & \frac{1}{8}
\end{array}\right] \tag{50}$$

It is noted that only a sub-matrix of the Bézier extraction operators $\mathbf{C}_0$, $\mathbf{C}_1$ and $\mathbf{C}_2$ with non-zero values is shown.

To remove global linear dependence of LR T-splines, several approaches have been proposed, including the hand-in-hand principle, the peeling algorithm and tensor expansion [8]. Herein, we will exploit the concept of analysis-suitable T-splines [4] to enforce global linear independence of LR T-splines. Analysis-suitable T-splines are a subset of T-splines [30], for which the extended T-mesh is analysis suitable. In an analysis-suitable T-mesh, T-junctions do not intersect. It is noted that in the remainder we consider a continuity reduction line as a special case of a meshline, such as line $\varepsilon_1 = \left[\xi_1^1, \xi_9^1\right] \times \xi_5^2$.

We next introduce the following notions:

- An end anchor A_i^l, A_i^u, which is the first anchor defined over meshline ε_i. There are two first anchors for each meshline: at the start, A_i^l, and at the end point, A_i^u, for example the anchors 1 (A_5^l) and 4 (A_5^u) of meshline $\varepsilon_5 = \left[\xi_3^1, \xi_{12}^1\right] \times \xi_7^2$ in Figure 17(a). For LR T-splines of even degree, the first anchor is the one which first appears in the knot vector $\boldsymbol{\Xi}$ which defines ε_i. For LR T-splines of odd degree, the determination of the first anchor is also straightforward, as it is the first anchor on the meshline ε_i.
- ex-meshline ε_i^l, ε_i^u, which is the part of the meshline ε_i with the end, see Figure 17. The superscripts l and u denote the position: at the start (l) or at the end point (u) of ε_i. The ex-meshline is determined as follows. Taking for example the vertical meshline $\varepsilon_i = \xi_\eta^1 \times \left[\xi_\alpha^2, \xi_\beta^2\right]$, then:



(a) cubic LR T-mesh

(b) elongation of meshline ε_1 by $\left[\xi^1_9,\, \xi^1_{10}\right] \times \xi^2_5$

Figure 17. A globally linearly dependent cubic LR T-mesh in the parameter domain and in the index domain. The linear dependencies between anchors 0, 1 and 2 are: $3N_0 = 3N_1 + N_2$, Figure (a). Figure (b) presents an elongation of meshline ε_1 by a meshline $\varepsilon = \left[\xi^1_9,\, \xi^1_{10}\right] \times \xi^2_5$.

  - For LR T-splines of even degree, $\varepsilon_i^l = \xi^1_\eta \times \left[\xi^2_\alpha,\, \xi^2_\zeta\right]$ and $\varepsilon_i^u = \xi^1_\eta \times \left[\xi^2_\kappa,\, \xi^2_\beta\right]$. ξ^2_ζ is the $(p/2+2)$th knot of the local knot vector Ξ^2 of the end anchor A_i^l. ξ^2_κ is the $(p/2+1)$th knot of the local knot vector Ξ^2 of the end anchor A_i^u.
  - For LR T-splines of odd degree, $\varepsilon_i^l = \xi^1_\eta \times \left[\xi^2_\alpha,\, \xi^2_\zeta\right]$ and $\varepsilon_i^u = \xi^1_\eta \times \left[\xi^2_\kappa,\, \xi^2_\beta\right]$. ξ^2_ζ is the $\left((p+1)/2+2\right)$th knot of the local knot vector Ξ^2 of the end anchor A_i^l. ξ^2_κ is the $\left((p-1)/2+1\right)$th knot of the local knot vector Ξ^2 of the end anchor A_i^u.

- The intersection I_i^j of ex-meshline ε_i^z and ex-meshline ε_j^z, $(i, j = 1, \cdots, k;\ z = l \,\text{or}\, u)$. Herein, only the case of a horizontal ex-meshline and a vertical ex-meshline is considered, for example the blue rectangles in Figure 17.
- The intersection set $\boldsymbol{\psi}$: $\boldsymbol{\psi} = \{\psi_i,\ i = 1, \cdots, k\}$, where ψ_i denotes the number of intersections I_i^j for ex-meshline ε_i^z, $i = 1, \cdots, k$, and k is the total number of ex-meshlines.

$\boldsymbol{\psi}$ has to be an empty set in order that LR T-splines can be analysis suitable. To enforce this, new meshlines ε_I have to be added. Here, the new meshlines ε_I are only elongations of existing meshlines or continuity reduction lines by one edge length, see Figure 17(b). The algorithm to construct analysis-suitable LR T-splines reads:

$\mathcal{S}1$ Check the LR T-mesh $\mathcal{T}$ whether it is globally linearly independent. If this is not the case, go to $\mathcal{S}2$ in order to make the LR T-mesh $\mathcal{T}$ analysis suitable. It is noted that this results in a mildly restricted subset of globally linearly independent LR T-splines.

$\mathcal{S}2$ Obtain the intersection set $\boldsymbol{\psi}$ of ex-meshlines.

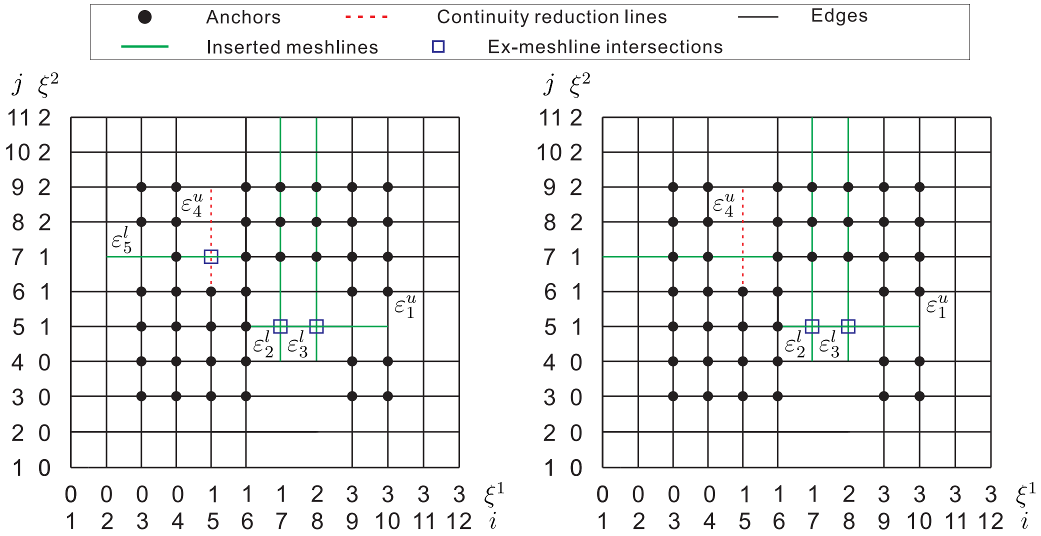
$\mathcal{S}3$ Among the possible meshline insertions ε_I, add the meshline for which $\sum_{i=1}^k \psi_i$ is smallest. For example, there are three intersections for ex-meshline ε_1^u, while the other ex-meshlines only have one or two intersections, Figure 17(a). After inserting all possible meshlines, the sum of ψ_i is updated as:

$\sum_{i=1}^k \psi_i$	ε_1^u	ε_2^l	ε_3^l	ε_4^u	ε_5^l
before meshline insertion	10	10	10	10	10
after meshline insertion	8	10	10	8	8

We thus add the new meshline $\varepsilon_I = \left[\xi_9^1,\, \xi_{10}^1\right] \times \xi_5^2$ to ε_1^u, which generates the meshline $\varepsilon_1 = \left[\xi_1^1,\, \xi_{10}^1\right] \times \xi_5^2$, Figure 17(b).

$\mathcal{S}4$ Repeat $\mathcal{S}3$ until $\boldsymbol{\psi}$ is an empty set, $\sum_{i=1}^{k} \psi_i = 0$. ε_i^z should be removed from the ex-meshline set and there should be no consideration of ε_i^z in the intersection set $\boldsymbol{\psi}$ if the ex-meshline ε_i^z is elongated to the boundary edge of the LR T-mesh $\mathcal{T}$ during the meshline insertion process. As an example, the intersection set $\boldsymbol{\psi}$ is not empty after inserting meshline $\varepsilon_I = \left[\xi_9^1,\, \xi_{10}^1\right] \times \xi_5^2$. Therefore, we have to insert a new meshline in the LR T-mesh $\mathcal{T}$. After checking all possible meshline insertions ε_I, the new meshline $\varepsilon_I = \left[\xi_2^1,\, \xi_3^1\right] \times \xi_7^2$ is inserted, which transforms ε_5 into $\varepsilon_5 = \left[\xi_2^1,\, \xi_{12}^1\right] \times \xi_7^2$, see Figure 18(a). Still, the intersection set $\boldsymbol{\psi}$ is not empty. We must therefore insert a new meshline, $\varepsilon_I = \left[\xi_1^1,\, \xi_2^1\right] \times \xi_7^2$, which is inserted to meshline ε_5, yielding $\varepsilon_5 = \left[\xi_1^1,\, \xi_{12}^1\right] \times \xi_7^2$, see Figure 18(b). However, the ex-meshline ε_5^l now touches the boundary edge of $\mathcal{T}$. Hence, we have to remove ε_5^l from the ex-meshline set and do not consider ε_5^l in the intersection set $\boldsymbol{\psi}$, Figure 18(b). Repeating step $\mathcal{S}3$, we obtain the final LR T-mesh $\mathcal{T}$, Figure 19(d), which is globally linearly independent and analysis suitable.

Remark 3: The algorithm always terminates, because the limiting case of meshline insertions in $\mathcal{T}$ renders $\mathcal{T}$ a tensor-product mesh.



(a) elongation of meshline ε_5 by $\left[\xi_2^1,\, \xi_3^1\right] \times \xi_7^2$

(b) elongation of meshline ε_5 by $\left[\xi_1^1,\, \xi_2^1\right] \times \xi_7^2$

Figure 18. Elongation of meshline ε_5. Figure (a) presents an elongation of meshline ε_5 by a meshline $\left[\xi_2^1,\, \xi_3^1\right] \times \xi_7^2$, which generates $\varepsilon_5 = \left[\xi_2^1,\, \xi_{12}^1\right] \times \xi_7^2$. Figure (b) presents an elongation of meshline ε_5 by a meshline $\left[\xi_1^1,\, \xi_2^1\right] \times \xi_7^2$, which generates $\varepsilon_5 = \left[\xi_1^1,\, \xi_{12}^1\right] \times \xi_7^2$.

6.3.2. Local linear independence When considering the Bézier extraction operator at element level, the condition for local linear independence is that $\boldsymbol{\alpha}_e = \mathbf{0}$ is the only solution of

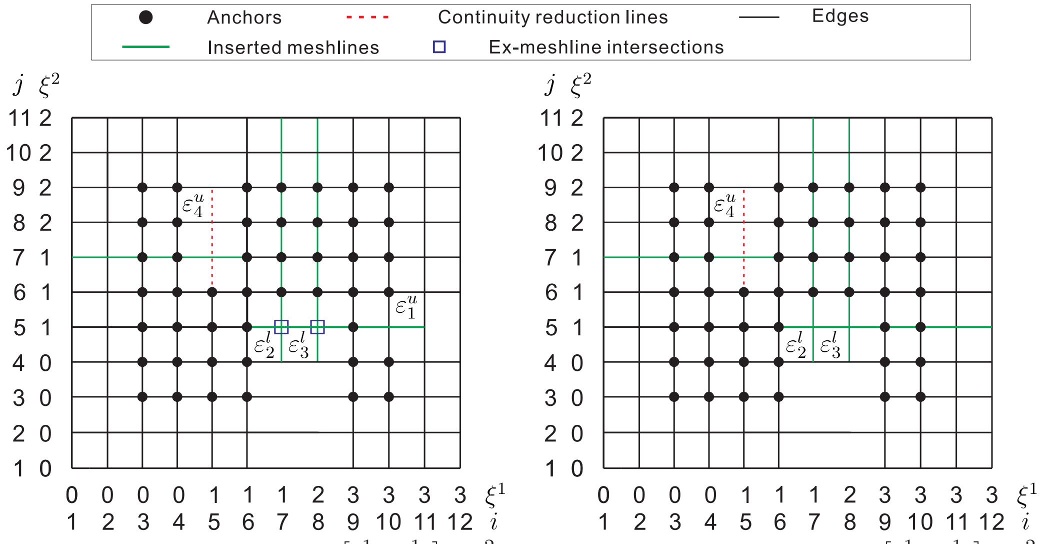
$$\mathbf{C}_e^{\mathrm{T}} \boldsymbol{\alpha}_e = \mathbf{0} \qquad \text{for} \qquad e = 1, \cdots, E \tag{51}$$

where $\mathbf{C}_e$ is the element Bézier extraction operator, Equation (5). The condition of local linear independence in Equation (51) is equivalent to:

$$\operatorname{rank}\left(\mathbf{C}_e\right) = n_e \qquad \text{for} \qquad e = 1, \cdots, E \tag{52}$$

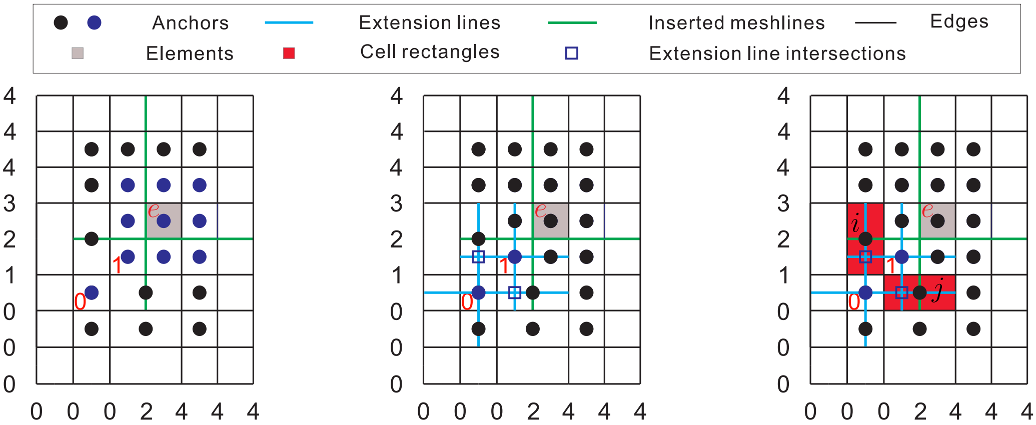
with n_e the number of anchors with support over element e. When an LR T-mesh is locally linearly dependent, the dependencies between the anchors can be detected by transforming Equation (51) into a row echelon form by Gaussian elimination.

DOI: 10.1002/nme



(a) elongation of meshline ε_1 by $\left[\xi^1_{10},\, \xi^1_{11}\right] \times \xi^2_5$ (b) elongation of meshline ε_1 by $\left[\xi^1_{11},\, \xi^1_{12}\right] \times \xi^2_5$

Figure 19. Elongation of meshline ε_1. Figure (a) presents an elongation of meshline ε_1 by a meshline $\left[\xi^1_{10},\, \xi^1_{11}\right] \times \xi^2_5$, which generates $\varepsilon_1 = \left[\xi^1_1,\, \xi^1_{11}\right] \times \xi^2_5$. Figure (b) presents an elongation of meshline ε_1 by a meshline $\left[\xi^1_{11},\, \xi^1_{12}\right] \times \xi^2_5$, which generates $\varepsilon_1 = \left[\xi^1_1,\, \xi^1_{12}\right] \times \xi^2_5$.



(a) (b) (c)

Figure 20. Example of a locally linearly dependent LR mesh in the parameter domain. The linear dependencies between anchors 0 and 1 over element e are: $6N_0 = N_1$, Figure (a). Figure (b) presents the extension lines of anchors 0 and 1. Figure (c) illustrates the cell rectangles with the extension line intersections.

Equation (52) implies Equation (48) – local linear independence yields global linear independence. Conversely, global linear independence does not imply locally linear independence, see Figure 20(a). For the LR T-mesh $\mathcal{T}$ of Figure 20(a), the rank of the Bézier extraction operator $\mathbf{C}$ in Equation (47) is $\text{rank}\,(\mathbf{C}) = 21$, which indicates the LR T-splines defined over $\mathcal{T}$ are globally linearly independent. However, $\mathcal{T}$ is not a locally linearly independent LR T-mesh. For instance, there are 10 anchors defined over element e, Figure 20(a). The rank of the element Bézier extraction operator $\mathbf{C}_e$ is $\text{rank}\,(\mathbf{C}_e) = 9$. The dependencies between anchors 0 and 1 are: $6N_0 = N_1$. To remove this dependency, a new meshline must be inserted. First, we draw extension lines from anchors 0 and 1, the blue lines in Figure 20(b), where the extension line is defined by shooting a ray from the anchor and matching horizontally and vertically until the edges defined by the local knot vectors confronting. These extension lines intersect in the red cell rectangles i and j, Figure 20(c).

DOI: 10.1002/nme

Figure 21 shows the possible meshline insertions for the cell rectangles i and j. In Figure 21(a), the meshline insertion for the cell rectangle i is an elongation of an existing meshline. It yields a globally and locally linearly independent LR T-mesh. This is also the case in Figure 21(b). However, the meshline insertions in Figures 21(c) and 21(d) yield a globally linearly independent LR T-mesh, which is not locally linearly independent.

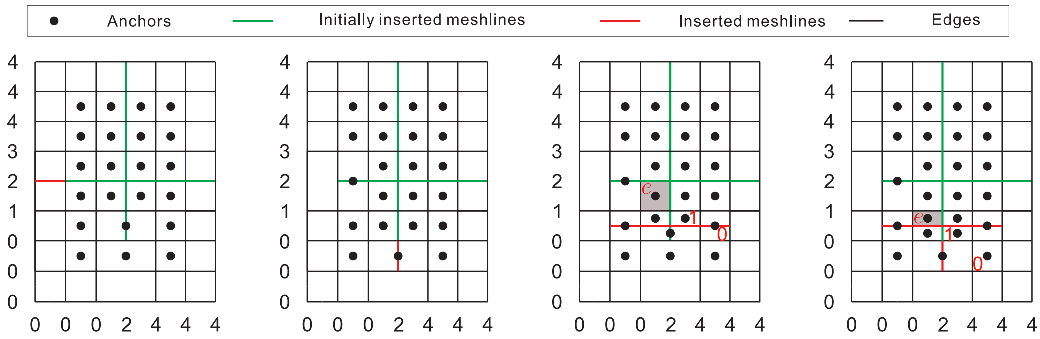
We can repeat this process to remove the linear dependencies. In the analysis, one can impose a strong condition for the linear independence of LR T-splines: locally linear independence, because this also yields global linear independence. The steps to guarantee local linear independence of LR T-splines are as follows:

$\mathcal{S}1$ Check local linear independence for each element. If all elements are equipped with locally linearly independent blending functions, the LR T-mesh is globally linearly independent. Otherwise, mark this element and go to step $\mathcal{S}2$.

$\mathcal{S}2$ Find the locally linearly dependent anchors defined over the marked element by Gaussian elimination and mark these anchors.

$\mathcal{S}3$ Build the extension lines of the marked anchors and find the intersections of these lines.

$\mathcal{S}4$ Locate the cell rectangles with the intersections and insert new meshlines into these cell rectangles. Update anchors, local knot vectors and elements, and go to step $\mathcal{S}1$ until the LR T-mesh is locally linearly independent.



(a) (b) (c) (d)

Figure 21. Possible meshline insertions for the red cell rectangles in Figure 20(c). (a) meshline insertions for the cell rectangle i; (b)–(d) meshline insertions for the cell rectangle j. The LR T-meshes in Figures (a) and (b) are locally linearly independent, while those in Figures (c) and (d) are locally linearly dependent. The element with locally linearly dependent blending functions is indicated in corresponding figures, including the respective anchors.

7. CONCLUDING REMARKS

We have extended the LR B-splines to the LR T-splines by exploiting the Bézier extraction operator concept. The basic idea of LR T-splines is to locally enrich the basis function space by replacing coarse grid T-splines by fine grid T-splines. The properties of LR T-splines have been investigated using the Bézier extraction framework. They include the partition of unity property, the nested nature and linear dependence, both globally and locally. Globally linearly dependent LR T-splines cannot be used in analysis. But even when global linear independence is satisfied, local linear dependence can occur, which necessitates the LR T-mesh to be refined by adding new meshlines in order to split existing elements.

ACKNOWLEDGEMENT

Financial support through ERC Advanced Grant 664734 ”PoroFrac” is gratefully acknowledged.

REFERENCES

[1] Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons: Chichester, 2009.

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

[3] Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T. T-spline simplification and local refinement. *ACM Transactions on Graphics* 2004; **23**:276–283.

[4] Scott M, Li X, Sederberg T, Hughes T. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering* 2012; **213**:206–222.

[5] Evans EJ, Scott MA, Li X, Thomas DC. Hierarchical T-splines: Analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 2015; **284**:1–20.

[6] Wei X, Zhang Y, Liu L, Hughes TJR. Truncated T-splines: Fundamentals and methods. *Computer Methods in Applied Mechanics and Engineering* 2017; **316**:349–372.

[7] Chen L, de Borst R. Adaptive refinement of hierarchical T-splines. *Computer Methods in Applied Mechanics and Engineering* 2017; (submitted).

[8] Dokken T, Lyche T, Pettersen KF. Polynomial splines over locally refined box-partitions. *Computer Aided Geometric Design* 2013; **30**:331–356.

[9] Bressan A. Some properties of LR-splines. *Computer Aided Geometric Design* 2013; **30**:778–794.

[10] Johannessen KA, Kvamsdal T, Dokken T. Isogeometric analysis using LR B-splines. *Computer Methods in Applied Mechanics and Engineering* 2014; **269**:471–514.

[11] Vuong AV, Giannelli C, Jüttler B, Simeon B. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**(49):3554–3567.

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

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

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

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

[16] Deng J, Chen F, Li X, Hu C, Tong W, Yang Z, Feng Y. Polynomial splines over hierarchical T-meshes. *Graphical models* 2008; **70**(4):76–86.

[17] Nguyen-Thanh N, Nguyen-Xuan H, Bordas SPA, Rabczuk T. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Computer Methods in Applied Mechanics and Engineering* 2011; **200**:1892–1908.

[18] Johannessen KA, Kumar M, Kvamsdal T. Divergence-conforming discretization for the Stokes problem on locally refined meshes using LR B-splines. *Computer Methods in Applied Mechanics and Engineering* 2015; **293**:38–70.

[19] Johannessen KA, Remonato F, Kvamsdal T. On the similarities and differences between classical hierarchical, truncated hierarchical and LR B-splines. *Computer Methods in Applied Mechanics and Engineering* 2015; **291**:64–101.

[20] Kumar M, Kvamsdal T, Johannessen KA. Simple a posteriori error estimators in adaptive isogeometric analysis. *Computers & Mathematics with Applications* 2015; **70**:1555–1582.

[21] Kumar M, Kvamsdal T, Johannessen KA. Superconvergent patch recovery and a posteriori error estimation technique in adaptive isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering* 2017; **316**:1086–1156.

[22] Zimmermann C, Sauer RA. Adaptive local surface refinement based on LR NURBS and its application to contact. *Computational Mechanics* 2017; :https://doi.org/10.1007/s00 466–017–1455–7.

[23] Bazilevs Y, Calo VM, Cottrell JA, Evans JA, Hughes TJR, Lipton S, Scott MA, Sederberg TW. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**:229–263.

[24] Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of T-splines. *International Journal for Numerical Methods in Engineering* 2011; **88**:126–156.

[25] May S, Vignollet J, de Borst R. 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* 2015; **103**:547–581.

[26] Cox MG. The numerical evaluation of B-splines. *IMA Journal of Applied Mathematics* 1972; **10**:134–149.

[27] de Boor C. On calculating with B-splines. *Journal of Approximation Theory* 1972; **6**:50–62.

[28] Chen L, Lingen FJ, de Borst R. Adaptive hierarchical refinement of NURBS in cohesive fracture analysis. *International Journal for Numerical Methods in Engineering* 2017; :https://doi.org/10.1002/nme.5600.

[29] de Borst R, Chen L. The role of Bézier extraction in adaptive isogeometric analysis: Local refinement and hierarchical refinement. *International Journal for Numerical Methods in Engineering* 2017; :https://doi.org/10.1002/nme.5696.

[30] Li X, Zheng J, Sederberg TW, Hughes TJR, Scott MA. On linear independence of T-spline blending functions. *Computer Aided Geometric Design* 2012; **29**:63–76.