This is a repository copy of *The role of Bézier extraction in adaptive isogeometric analysis: Local refinement and hierarchical refinement*.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/121131/

Version: Published Version

Article:
de Borst, R. orcid.org/0000-0002-3457-3574 and Chen, L. (2018) The role of Bézier extraction in adaptive isogeometric analysis: Local refinement and hierarchical refinement. International Journal for Numerical Methods in Engineering, 113 (6). pp. 999-1019. ISSN 0029-5981

https://doi.org/10.1002/nme.5696

This is the peer reviewed version of the following article: Borst, R., and Chen, L. (2017) The Role of Bézier Extraction in Adaptive Isogeometric Analysis: Local Refinement and Hierarchical Refinement. Int. J. Numer. Meth. Engng, which has been published in final form at https://doi.org/10.1002/nme.5696. This article may be used for non-commercial purposes in accordance with Wiley Terms and Conditions for Self-Archiving.

Reuse
This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can't change the article in any way or use it commercially. More information and the full terms of the licence here: https://creativecommons.org/licenses/

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Received: 27 June 2017 | Accepted: 4 September 2017

DOI: 10.1002/nme.5696

RESEARCH ARTICLE

WILEY

The role of Bézier extraction in adaptive isogeometric analysis: Local refinement and hierarchical refinement

R. de Borst | L. Chen

Department of Civil and Structural Engineering, University of Sheffield, Sheffield, S1 3JD, UK

Correspondence
René de Borst, Department of Civil and Structural Engineering, University of Sheffield, Sheffield, S1 3JD, UK.
Email: r.deborst@sheffield.ac.uk

Funding information
European Research Council, Grant/Award Number: 664734

Summary

We present 2 adaptive refinement techniques, namely, adaptive local refinement and adaptive hierarchical refinement, for isogeometric analysis. An element-wise point of view is adopted, exploiting Bézier extraction, which facilitates the implementation of adaptive refinement in isogeometric analysis. Locally refined and hierarchical T-splines are used for the description of the geometry as well as for the approximation of the solution space in the analysis. The refinement is conducted with the aid of a subdivision operator, which is computed by again exploiting the Bézier extraction operator. The concept and algorithm of an element-based adaptive isogeometric analysis are illustrated. Numerical examples are given to examine the accuracy, the convergence, and the condition number.

KEYWORDS

isogeometric analysis, Bézier extraction, adaptive refinement, LR T-splines, hierarchical T-splines

1 | INTRODUCTION

Isogeometric analysis (IGA) has been used widely because it can result in a seamless integration of the design and analysis processes: the non-uniform rational basis spline (NURBS) functions that are typically used in Computer Aided Geometric Design packages can be reused in the analysis, and, in principle, no (re)meshing is required.[1,2] The ability of the spline functions used in IGA to capture the geometry exactly and thus to reduce the approximation error that stems from the geometry description is another major advantage of the technology. A drawback from the use of NURBS, however, is that only global refinement is possible due to the tensor-product structure of the basis functions in two- and three-dimensional analyses.

Ideally, IGA should pair an exact geometry representation capability with local adaptive mesh refinement to provide a truly local *h* refinement. For this purpose, the concept of T-splines has been investigated.[3-6] T-spline technology breaks the rigid tensor-product structure of NURBS by inserting extra vertices into the tensor product mesh. The mathematical properties of T-splines, such as the linear independence of the bases, have been discussed in Li et al, Morgenstern and Peterseim, and May et al.[7-9] More recently, local refinement of T-splines has been investigated,[10-12] as it further increases the flexibility of T-splines. Alternatively, PHT-spline can also provide local adaptivity since they are defined over hierarchical T-meshes.[13-15]

Adaptive hierarchical refinement, which can be considered as a special type of local mesh refinement, has recently gained considerable attention.[16-26] The basic idea is to locally enrich the approximation space by replacing selected coarse

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.
© 2017 The Authors. *International Journal for Numerical Methods in Engineering* Published by John Wiley & Sons, Ltd.

grid T-splines by fine grid T-splines. The hierarchical and truncated hierarchical bases are considered for the geometry description as well as to span the approximation space for the solution. Locally refined (LR) B-splines have been introduced to achieve a local h refinement.[27-33] Since then, LR B-splines have been extended to LR NURBS[34] and LR T-splines,[35] which enables a more flexible geometry description.

The element-based Bézier extraction operator is nowadays usually used to implement the basis functions used in isogeometic analysis,[6,9,23,36] as it smoothly aligns with existing finite element codes. However, in addition to the implementation of basis functions in a standard finite element framework, the Bézier extraction operator can be applied to hierarchical refinement[12,23,26] and local refinement.[9,35] In this contribution, we will further pursue this role. Adaptive local refinement as well as adaptive hierarchical refinement are considered, and the algorithms developed for the implementation will be detailed.

This contribution starts with a discussion of Bézier extraction of refined T-splines. The construction of the Bézier extraction operator and the subdivision operator for T-splines are given. The implementation of adaptive local refinement is discussed in Section 3, followed by an illustration of the use of the Bézier extraction operator for adaptive hierarchical refinement in Section 4. This section also provides the algorithm for element-based adaptive IGA. Some numerical examples and concluding remarks finalise the manuscript.

2 | BÉZIER EXTRACTION OF REFINED T-SPLINES

We first review the concepts of T-splines and Bézier extraction.[5,6,9] The Bézier extraction framework will be formulated such that it includes refined T-splines, which can be used in adaptive IGA. It is noted that in the sequel, we consider T-splines with the same polynomial degree p in all parametric directions.

2.1 | T-spline and Bézier extraction fundamentals

A T-mesh $\mathcal{T}$ is composed of quadrilateral elements with T-junctions. Elements are defined as nonzero parametric areas which are confined by the edges of the T-mesh and the continuity reduction lines. An example of a quadratic T-spline mesh is given in Figure 1.

In a T-mesh, anchors are prescribed in the index domain and in the parameter domain (Figure 1). A multivariate blending function N is attached to each anchor and is determined by the Cox-de Boor recursion formula.[37,38] We define the union of T-spline blending functions as a T-spline space $\mathcal{N} = \{N_i : \text{supp}N_i \in \mathcal{T}\}$. A local knot vector $\Xi_i\,(i = 1, \cdots, n)$ is prescribed for each anchor to construct N, with n as the number of anchors on $\mathcal{T}$. A T-spline surface $\mathcal{S}\left(\xi^1, \xi^2\right)$ is described by anchors and blending functions:

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

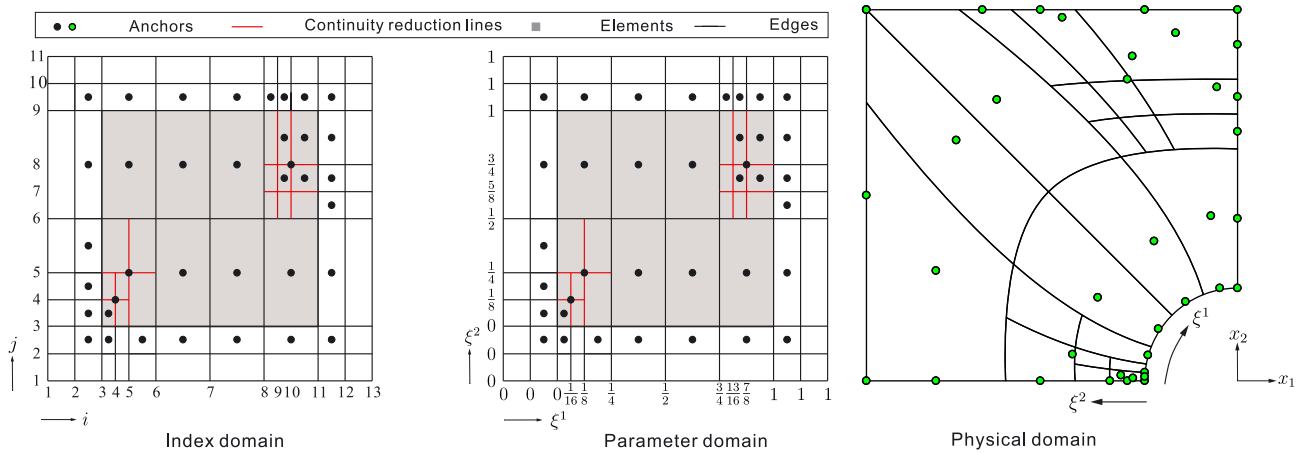


FIGURE 1 Example of a quadratic T-spline mesh. The object is given in the index domain (i, j), in the physical domain (x_1, x_2), and in the parameter domain $\left(\xi^1, \xi^2\right)$

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

where $\mathcal{A}$ is the index set of anchors, $\mathbf{P}_\alpha$ denotes the coordinates of anchors, and γ_α the scaling weight, which enables the T-splines to satisfy the partition of unity property.[35] The construction of the local knot vectors and the blending functions has been described in May et al.[9]

For completeness, we also consider rational T-splines:

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

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

$$S\left(\xi^1,\ \xi^2\right) = \sum_{\alpha\in A} \mathbf{P}_\alpha R_\alpha\left(\xi^1,\ \xi^2\right)\gamma_\alpha, \tag{3}$$

where $\mathbf{P}_\alpha = \left(x_\alpha^1,\ x_\alpha^2,\ w_\alpha\right)$ contains the coordinates of anchor α. The weighted coordinates of anchor α are $\mathbf{P}_\alpha^w = \left(w_\alpha x_\alpha^1,\ w_\alpha x_\alpha^2,\ w_\alpha\right)$.

Generally, blending functions are defined over the entire support of an anchor. It is cumbersome to directly incorporate blending functions in standard finite element data structures. However, Bézier extraction provides an elegant work-around by representing T-splines as element-wise Bernstein shape functions.[6] We consider that $\mathcal{T}$ is divided into E elements with n anchors. For anchor i, the local knot vectors are Ξ_i^1 and Ξ_i^2, and the blending function N_i can be written as

$$N_i^e\left(\xi^1,\ \xi^2\right) = [\mathbf{C}_i^e]^{\mathrm{T}}\mathbf{B}^e\left(\xi^1,\ \xi^2\right) \tag{4}$$

over element e with $(p+1)^2 \times 1$ bivariate Bernstein shape functions $\mathbf{B}^e\left(\xi^1,\ \xi^2\right)$.[9] $\mathbf{C}_i^e$ is the Bézier extraction operator of anchor i over element e.[26] For the Bézier extraction operator of anchor i, which extends over E elements, we have

$$\mathbf{C}_i = \begin{bmatrix} \mathbf{C_i^1} \\ \vdots \\ \mathbf{C_i^E} \end{bmatrix} \tag{5}$$

with the dimension $E(p+1)^2 \times 1$. Writing the Bézier extraction operator for n anchors in a matrix form then leads to

$$\mathbf{N}\left(\xi^1,\ \xi^2\right) = \mathbf{CB}\left(\xi^1,\ \xi^2\right) = \begin{bmatrix} N_1\left(\xi^1,\ \xi^2\right) \\ \vdots \\ N_n\left(\xi^1,\ \xi^2\right) \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{C}_n^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{B}^1 \\ \vdots \\ \mathbf{B}^E \end{bmatrix}. \tag{6}$$

The Bézier extraction operator for anchors with support over element e then gives

$$\mathbf{N}_e\left(\xi^1,\ \xi^2\right) = \mathbf{C}_e\mathbf{B}_e\left(\xi^1,\ \xi^2\right) \tag{7}$$

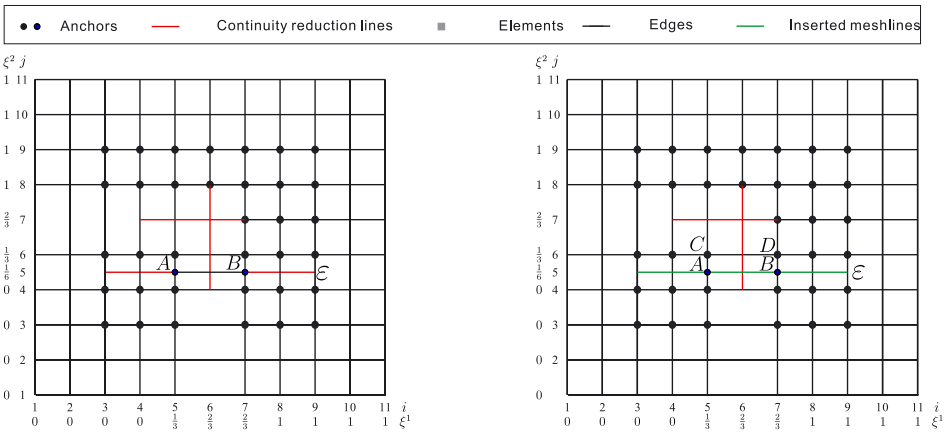
with $\mathbf{C}_e$ the element Bézier extraction operator.

2.2 | Equivalence of anchor insertion and meshline insertion

There are basically 2 approaches for element refinement: anchor insertion[10] and meshline insertion,[29] see Figure 2. We will now show graphically that both approaches are, in fact, equivalent for adaptive refinement. We consider Figure 2A, where the anchors A and B are inserted in the T-mesh $\mathcal{T}$. The local knot vectors of the anchors A and B are $\Xi_A^1 = \left\{\xi_3^1,\ \xi_4^1,\ \xi_5^1,\ \xi_7^1,\ \xi_8^1\right\}$ and $\Xi_A^2 = \left\{\xi_3^2,\ \xi_4^2,\ \xi_5^2,\ \xi_6^2,\ \xi_8^2\right\}$, and $\Xi_B^1 = \left\{\xi_4^1,\ \xi_5^1,\ \xi_7^1,\ \xi_8^1,\ \xi_9^1\right\}$ and $\Xi_B^2 = \left\{\xi_3^2,\ \xi_4^2,\ \xi_5^2,\ \xi_6^2,\ \xi_7^2\right\}$, respectively. Connecting the knots of anchors A and B, we obtain a new horizontal meshline $\varepsilon = \left[\xi_3^1,\ \xi_9^1\right] \times \xi_5^2$. Hence, anchor insertions lead to meshline insertions.

Next, we consider Figure 2B, where we insert a new meshline, $\varepsilon = \left[\xi_3^1,\ \xi_9^1\right] \times \xi_5^2$ in $\mathcal{T}$. As a consequence of this, the blending function of anchor C will be split into 2 separate blending functions that are associated with the anchors C' and A. The local knot vectors of the anchors C, C', and A are $\Xi_C^1 = \left\{\xi_3^1,\ \xi_4^1,\ \xi_5^1,\ \xi_7^1,\ \xi_8^1\right\}$ and $\Xi_C^2 = \left\{\xi_3^2,\ \xi_4^2,\ \xi_6^2,\ \xi_8^2,\ \xi_9^2\right\}$, $\Xi_{C'}^1 = \left\{\xi_3^1,\ \xi_4^1,\ \xi_5^1,\ \xi_7^1,\ \xi_8^1\right\}$ and $\Xi_{C'}^2 = \left\{\xi_4^2,\ \xi_5^2,\ \xi_6^2,\ \xi_8^2,\ \xi_9^2\right\}$, and $\Xi_A^1 = \left\{\xi_3^1,\ \xi_4^1,\ \xi_5^1,\ \xi_7^1,\ \xi_8^1\right\}$ and $\Xi_A^2 = \left\{\xi_3^2,\ \xi_4^2,\ \xi_5^2,\ \xi_6^2,\ \xi_8^2\right\}$, respectively. The positions of the anchors C and C' are the same in the index domain and in the parameter domain, while anchor A is newly inserted in the index domain and the parameter domain. If we consider the blending function splitting for the remaining anchors, we obtain the anchor B in the index domain and the parameter domain, which is also newly inserted. Accordingly, meshline insertions yield anchor insertions.

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License



(A) anchor insertion (B) meshline insertion

FIGURE 2 Example of anchor insertions and meshline insertions in a cubic T-spline mesh. The object is given in the index domain $(i,\ j)$ and in the parameter domain $\left(\xi^1,\ \xi^2\right)$

In sum, anchor insertions and meshline insertions are equivalent techniques for adaptive refinement. Both approaches lead to the same refined T-splines. In the remainder, we will exclusively use the term "meshline insertion."

2.3 | Subdivision operator and control point

Now, we will extend the Bézier extraction framework to T-splines after meshline insertions. As point of departure, we take a T-mesh, $\mathcal{T}$, with n anchors. Inserting a series of single meshlines, $\{\epsilon_i\}_{i=1}^{n}$, in $\mathcal{T}$ results in $\mathcal{T}_r$ with n_r anchors. The T-splines $\mathcal{N}$ that are associated with $\mathcal{T}$ are now described by the T-splines $\mathcal{N}_r$ associated with $\mathcal{T}_r$:

$$\Gamma\mathbf{N}\left(\xi^1,\ \xi^2\right) = \Gamma\mathbf{S}\mathbf{N}_r\left(\xi^1,\ \xi^2\right), \tag{8}$$

where $\mathbf{S}$ is the refinement operator,[26,35,39] $\mathbf{N}$ and $\mathbf{N}_r$ are the blending functions associated with $\mathcal{T}$ and $\mathcal{T}_r$, respectively, and $\boldsymbol{\Gamma}$ is a diagonal matrix with the scaling weights γ of $\mathbf{N}$. Using Equation 6, we can now solve for $\mathbf{S}$:

$$\mathbf{N} = \mathbf{C}\mathbf{B}_r = \mathbf{S}\mathbf{C}_r\mathbf{B}_r, \tag{9}$$

where $\mathbf{C}$ is the Bézier extraction operator of the anchors on $\mathcal{T}$ over the elements on $\mathbf{T}_r$, $\mathbf{C}_r$ denotes the Bézier extraction operator of the anchors on $\mathbf{T}_r$ over the elements on $\mathbf{T}_r$, and $\mathbf{B}_r$ contains the Bernstein polynomials of the elements on $\mathbf{T}_r$.

The row values of $\mathbf{S}$ can subsequently be obtained by expanding the right side of Equation 9, as follows:

$$\begin{bmatrix} \mathbf{C}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{C}_n^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{S}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{S}_n^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{r1}^{\mathrm{T}} \\ \vdots \\ \mathbf{C}_{rn_r}^{\mathrm{T}} \end{bmatrix}, \tag{10}$$

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

$$\mathbf{C}_i = \mathbf{C}_r^{\mathrm{T}}\mathbf{S}_i \qquad \text{for} \qquad i = 1,\ \cdots,\ n \tag{11}$$

Considering Equation 8, the scaling weight γ^r of $\mathbf{N}_r$ is derived as

$$\Upsilon^r = \Upsilon\mathbf{S} \quad \text{with} \quad \Upsilon^r = \left[\gamma_1^r,\ \gamma_2^r,\ \cdots,\ \gamma_{n_r}^r\right] \quad \text{and} \quad \Upsilon = \left[\gamma_1,\ \gamma_2,\ \cdots,\ \gamma_n\right]. \tag{12}$$

From Equation 3, the weighted surface S^w is given as

$$S^w\left(\xi^1,\ \xi^2\right) = \sum_{\alpha=1}^{n} \gamma_\alpha N_\alpha\left(\xi^1,\ \xi^2\right)\mathbf{P}_\alpha^w. \tag{13}$$

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

The weighted surface defined by $\mathbf{T}$ and $\mathbf{T}_r$ should represent the same geometry, so that

$$\mathcal{S}^w\left(\xi^1,\ \xi^2\right) = \mathcal{S}_r^w\left(\xi^1,\ \xi^2\right) \tag{14}$$

Inserting Equation 13 into (14) then yields

$$\sum_{\alpha=1}^{n} \gamma_\alpha N_\alpha\left(\xi^1,\ \xi^2\right)\mathbf{P}_\alpha^w = \sum_{\beta=1}^{n_r} \gamma_\beta^r N_{r\beta}\left(\xi^1,\ \xi^2\right)\mathbf{P}_{r\beta}^w \tag{15}$$

in which γ_α, $N_\alpha\left(\xi^1,\ \xi^2\right)$, and $\mathbf{P}_\alpha^w$ are geometrical properties associated with $\mathcal{T}$, while γ_β^r, $N_{r\beta}\left(\xi^1,\ \xi^2\right)$ and $\mathbf{P}_{r\beta}^w$ are those associated with $\mathbf{T}_r$. Considering the Bézier extraction operator and Equations 9 and 10 results in

$$\mathbf{P}_r^w = \Gamma_r^{-1}\mathbf{S}^{\mathrm{T}}\Gamma\mathbf{P}^w, \tag{16}$$

where Γ_r is a diagonal matrix with the scaling weight γ_β^r of $\mathbf{N}_r$ along the diagonal, see Equation 12. $\mathbf{P}^w$ and $\mathbf{P}_r^w$ are column vectors with the control points $\mathbf{P}_\alpha^w$ and $\mathbf{P}_{r\beta}^w$, respectively.

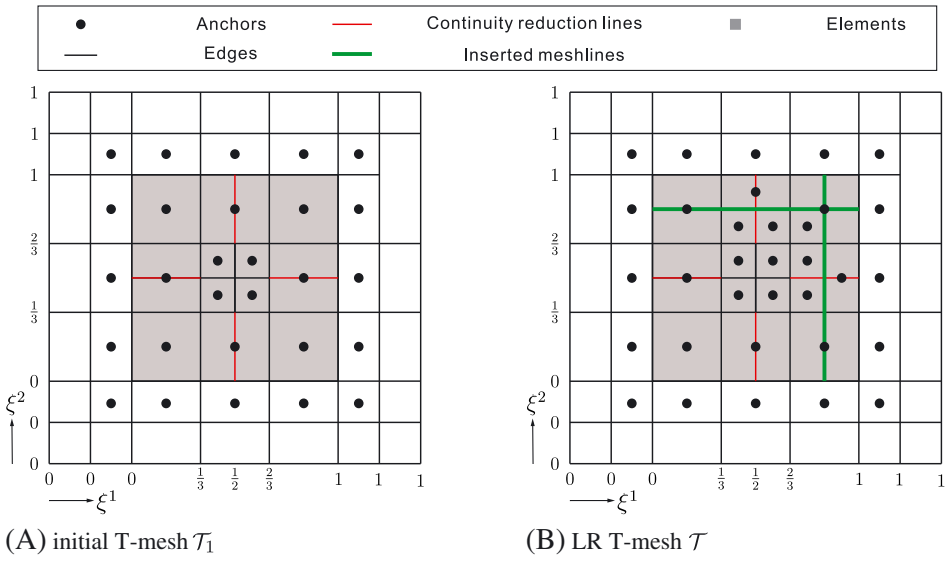
3 | LOCAL REFINEMENT USING BÉZIER EXTRACTION

The most common adaptive refinement technique for T-splines is adaptive local refinement by anchor insertions or meshline insertions in a T-mesh. This technique is normally implemented using the T-spline blending function subdivision approach.[11] Herein, we will use Bézier extraction. First, we review some basic aspects of a recent LR T-spline technology.[35] Then, different refinement strategies will be given for adaptive local refinement on the basis of T-splines, including implementation aspects.

3.1 | LR T-spline fundamentals

Locally refined T-splines are an extension of LR B-splines.[1,35] We consider an initial T-mesh with n anchors. Each anchor is associated with a local knot vector Ξ_i $(i = 1,\ \cdots,\ n)$ and a blending function $N_i\left(\xi^1,\ \xi^2\right)$. When we conduct a sequence of single meshline insertions $\{\varepsilon_i\}_{i=1}^n$ in $\mathcal{T}_1$, we obtain a nested LR T-mesh, $\mathcal{T}_n$, such that $\mathcal{T}_n \supset \mathcal{T}_{n-1} \supset \cdots \supset \mathcal{T}_2 \supset \mathcal{T}_1$ (Figure 3). Intermediate states are denoted by $\mathcal{T}_{i+1} = \{\mathcal{T}_i \cup \varepsilon_i\}$. In an LR T-mesh, elements are nonzero parametric areas defined by the edges of T-mesh, continuity reduction lines and inserted meshlines, as illustrated in Figure 3.

Meshline insertions in an LR T-mesh should (1) pass through an element (knot span), (2) insert one meshline at a time, and (3) span across $p + 2$ knots or more. A meshline insertion ϵ on an LR T-mesh $\mathcal{T}_n$ is then either (1) a new meshline, (2) an elongation of an existing meshline or a continuity reduction line, (3) a joining of 2 existing meshlines or 2 existing



(A) initial T-mesh $\mathcal{T}_1$ (B) LR T-mesh $\mathcal{T}$

FIGURE 3 Example of a locally refined (LR) T-mesh in the parameter domain. The green lines indicate meshline insertions

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

continuity reduction lines, (4) a joining of an existing meshline and a continuity reduction line, or (5) an increase of the multiplicity of an existing meshline or continuity reduction line. 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 conduct the LR T-spline splitting.

For the LR T-mesh $\mathcal{T}$, we can obtain the LR T-spline blending functions N: $\mathbb{R}^2 \rightarrow \mathbb{R}$ 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 minimal support on $\mathcal{T}$, which implies that there is no other meshline traversing the interior space of N.

The union of LR T-spline functions is called an LR T-spline space $\mathcal{N} = \{N_i : \text{supp}N_i \in \mathcal{T}\}$. Locally refined T-splines form a partition of unity and are nested but are not necessarily globally or locally linearly independent.[35]

3.2 | Local refinement strategies

The idea of LR T-splines is to maintain their minimal support property after meshline insertion in an LR T-mesh $\mathcal{T}$. The refinement is realised by separate knot insertions in each parametric direction. As an example, we take the case of a knot insertion in the parametric direction ξ^1 and assume that an LR T-spline blending function N_i is defined by the local knot vectors

$$\Xi_i^1 = \left[\xi_1^1,\ \xi_2^1,\ \cdots,\ \xi_{i-1}^1,\qquad \xi_i^1,\ \cdots,\ \xi_{p+1}^1,\ \xi_{p+2}^1\right]$$

and

$$\Xi_i^2 = \left[\xi_1^2,\ \xi_2^2,\quad \cdots\cdots\cdots\cdots\cdots,\ \xi_{p+1}^2,\ \xi_{p+2}^2\right].$$

A new meshline, $\varepsilon = \hat{\xi} \times \left[\xi_1^2,\ \xi_{p+2}^2\right]$, is now inserted in $\mathcal{T}$. A new knot $\hat{\xi}$ must therefore be inserted in Ξ_i^1, while Ξ_i^2 remains constant. Hence, 2 additional local knot vectors, Ξ_{i1}^1 and Ξ_{i2}^1, result

$$\begin{aligned}\Xi_{i1}^1 &= \left[\xi_1^1,\ \xi_2^1,\ \cdots,\ \xi_{i-1}^1,\ \hat{\xi},\ \xi_i^1,\ \cdots,\ \xi_{p+1}^1\qquad\quad\right],\\ \Xi_{i2}^1 &= \left[\qquad\xi_2^1,\ \cdots,\ \xi_{i-1}^1,\ \hat{\xi},\ \xi_i^1,\ \cdots,\ \xi_{p+1}^1,\ \xi_{p+2}^1\right],\end{aligned} \tag{17}$$

as well as 2 new anchors with respect to local knot vectors Ξ_{i1}^1 and Ξ_i^2, Ξ_{i2}^1 and Ξ_i^2.

Applying this refinement procedure to all anchors on $\mathcal{T}$, we obtain the updated anchors and the updated elements on the refined LR T-mesh $\mathcal{T}_r$. The scaling weights of the updated T-spline blending functions and the updated control points are next obtained using Equations 12 and 16.

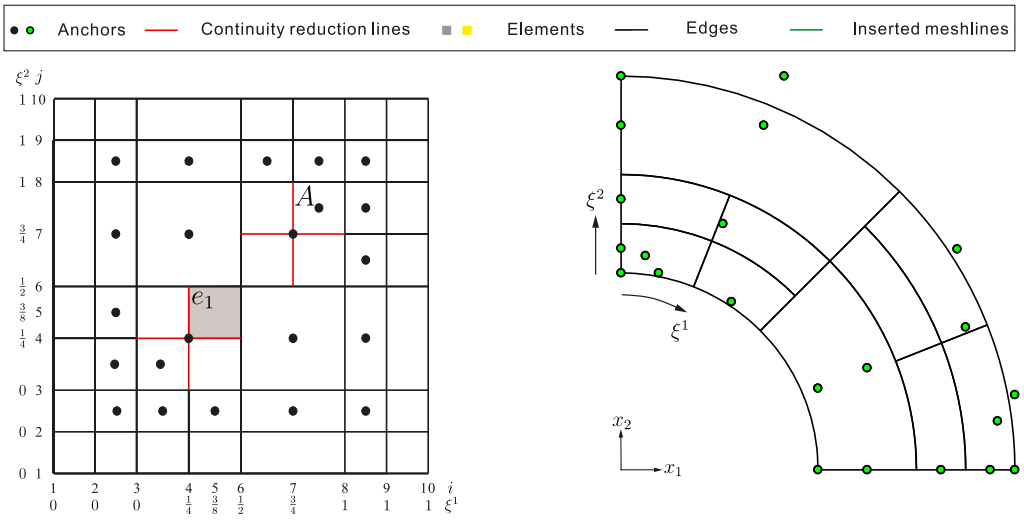


FIGURE 4 Initial LR T-spline surface. The object is given in the index domain $(i,\ j)$, in the physical domain $(x_1,\ x_2)$, and in the parameter domain $\left(\xi^1,\ \xi^2\right)$. The element $e1$ and the blending function of anchor A will be refined

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

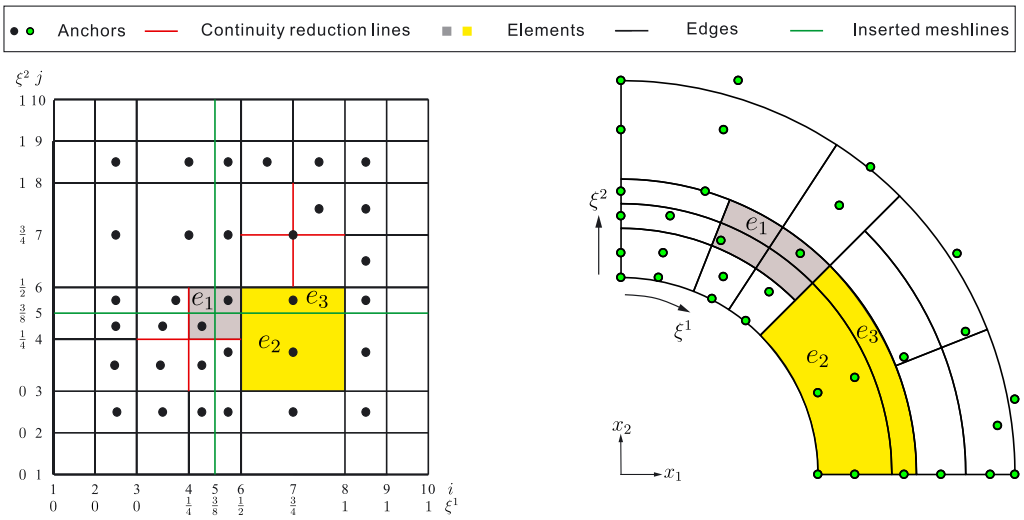


FIGURE 5 LR T-spline surface generated by the full span refinement strategy. It splits all blending functions with support over element $e1$

Different refinement strategies have been proposed for LR T-splines,[29] see also Figures 5 to 7. The figures show the full span and the minimal span refinement strategies which are based on element refinement (Figures 5 and 6). A third option, the structured mesh refinement strategy refines the LR T-spline blending function itself, as illustrated in Figure 7.

The starting point for all refinement strategies is that a certain element or blending function is marked for refinement, and we consider the quadratic LR T-spline surface of Figure 4 as the initial LR T-spline surface. The element $e1$ and the blending function of anchor A are meant to be refined (Figure 4).

Figure 5 illustrates full span refinement. In this approach, every LR T-spline blending function with support on the grey marked element $e1$ is refined. The element marked in grey is subdivided into 4 child elements, and the neighbouring elements are split by a single line. Evidently, this procedure can result in elements which have a poor aspect ratio, in this case, $e3$.

The minimum span refinement strategy inserts a cross through the centre of element $e1$ (Figure 6). The inserted meshline should be as short as possible but must split at least 1 LR T-spline blending function. It is noted that the choice of the blending function, which will be refined, is not unique.[29] Compared with the previous approach, this refinement strategy generally leads to a mesh which is better centred around the marked elements. Clearly, this is an important advantage for adaptive refinement in IGA. However, it can cause relatively poor aspect ratios in neighbouring elements, such as elements $e2$ and $e3$ in the present example.

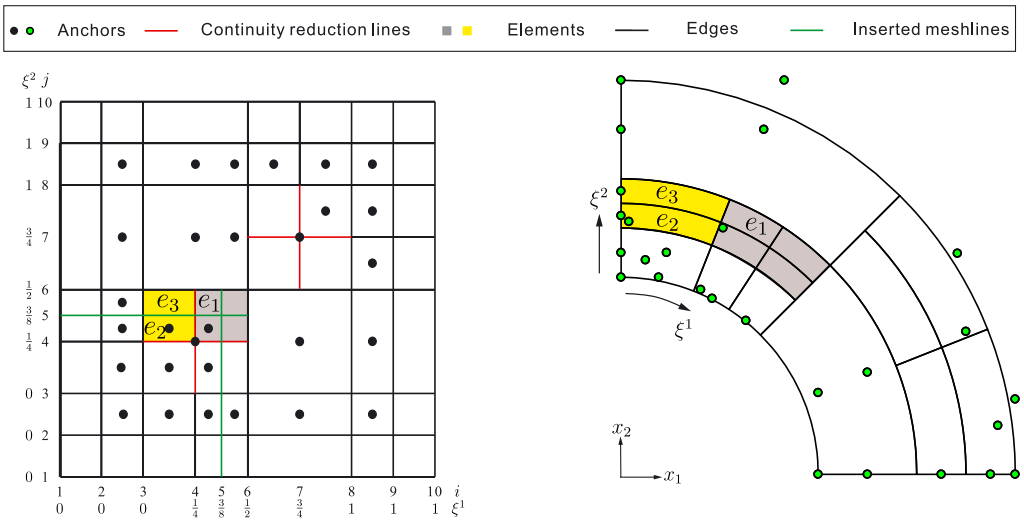


FIGURE 6 LR T-spline surface generated by the minimum span refinement strategy. It splits the blending function with support over element $e1$

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

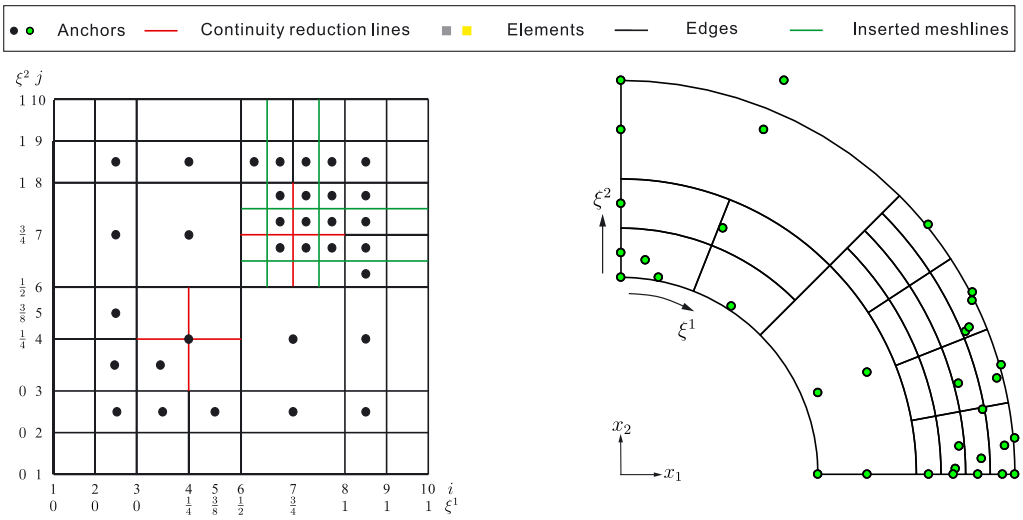


FIGURE 7 Locally reduced T-spline surface generated by the structured mesh refinement strategy. It splits all knot spans of local knot vectors of anchor A

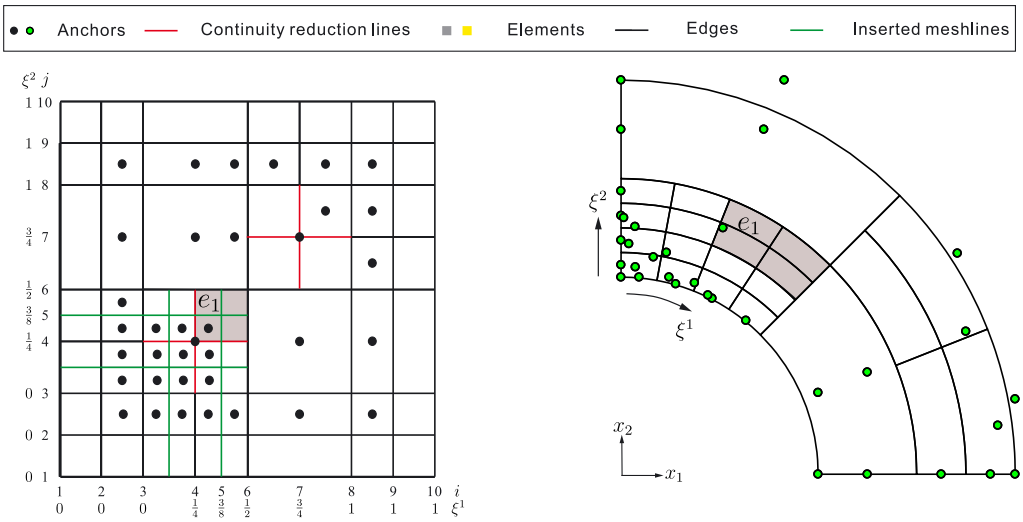


FIGURE 8 Locally refined T-spline surface generated by the element-based structured mesh refinement strategy, which combines the properties of the minimum span and the structured mesh refinement strategies. First, it determines the blending function with the smallest support over element $e1$. Then, the resulting blending function is refined using the structured mesh refinement strategy

Finally, Figure 7 illustrates the concept of structured mesh refinement. In this approach, a net of meshlines is inserted, which halves the largest knot intervals that support the T-spline blending function of anchor A, marked in Figure 4. Different from the previous methods which focus on element refinement, this strategy identifies the blending functions which must be refined and usually does not create elements with a poor aspect ratio.

In view of the advantages of the minimum span and the structured mesh refinement strategies, an element-based structured mesh refinement strategy is proposed, see Figure 8. First, the blending function is determined with a small support over element $e1$ using the minimum span refinement strategy. The blending function is chosen which is around element $e1$ as much as possible. Then, the blending function is refined using the structured mesh refinement strategy, as shown in Figure 8. It is noted that no poor aspect ratios are generated. In general, this refinement strategy not only refines the marked element, in this case, $e1$, but also the neighbouring elements, as shown in Figure 8. It is possible that some "over-refinement" will take place, i.e., refinement where the discrete solution is already accurate and to an almost uniformly refined mesh. This matter will be discussed further in Section 5, where some rules will be suggested to avoid this from happening or at least alleviate the phenomenon.

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

3.3 | Implementation of local refinement in IGA

For an object defined by an LR T-mesh $\mathcal{T}$, we always obtain a global system of equations from the weak form of the equilibrium equation and the Neumann boundary conditions. The steps for doing so are listed in Algorithm 1.

Algorithm 1 Adaptive local refinement

$\mathcal{S}1$ Read the geometry data of the initial T-spline surface to obtain the initial local knot vectors and the initial control points.

$\mathcal{S}2$ Obtain the element Bézier extraction operator $\mathbf{C}_e$, compute the stiffness matrix $\mathbf{K}$ and the force vector $\mathbf{F}$ using Bézier extraction.

$\mathcal{S}3$ Consider the Dirichlet boundary conditions and solve $\mathbf{KU} = \mathbf{F}$ to obtain the displacement vector $\mathbf{U}$.

$\mathcal{S}4$ Check whether elements should be refined and mark them accordingly. If there is no element marked for refinement, stop the calculation for the current load step and go to next load step. Otherwise, obtain the list of meshlines, which must be inserted in the current LR T-mesh $\mathcal{T}$, on the basis of the marked elements and the chosen refinement strategy.

$\mathcal{S}5$ Obtain the local knot vectors of the updated anchors and the updated elements after meshline insertions. This yields a new LR T-mesh $\mathcal{T}_r$.

$\mathcal{S}6$ Compute the subdivision operator $\mathbf{S}$ using Equation 11, obtain the control points of the updated anchors using Equation 16, and return to $\mathcal{S}2$.

4 | HIERARCHICAL REFINEMENT USING BÉZIER EXTRACTION

Refinement using hierarchical basis functions was originally used for the adaptive refinement of a surface[40] but subsequently also used in analysis.[17,19,21,41-44] To further improve the capability of hierarchical refinement truncated hierarchical bases were proposed in Giannelli et al[45] and Buffa and Giannelli.[46] Later, hierarchical and truncated hierarchical T-splines have been developed, which combine the ability of hierarchical B-splines for adaptive refinement with the capability of T-splines to provide an exact geometrical representation.[20,24,26]

4.1 | Hierarchical T-spline fundamentals

A hierarchical T-spline space is constructed from a finite sequence of L nested T-spline spaces $\mathcal{N}^l$ bounded by L parameter domains Ω^l, $l = 1, \cdots, L$. The nested nature of T-spline space defines the nested domains for the hierarchy:

$$\mathcal{N}^1 \subset \mathcal{N}^2 \subset \cdots \subset \mathcal{N}^L, \qquad \Omega^1 \subseteq \Omega^2 \subseteq \cdots \subseteq \Omega^L. \tag{18}$$

To define the L nested T-spline spaces such that $\mathcal{N}^\alpha \subset \mathcal{N}^{\alpha+1}$, $\alpha = 1, \cdots, L-1$, a multilevel T-spline mesh is constructed with a hierarchy of L levels. On the multilevel mesh, the sequence of L T-spline meshes $\mathcal{T}^{\alpha+1}$ is built by subdividing each effective rectangular cell in $\mathcal{T}^\alpha$ into 2 or 4 congruent cells by meshline insertions, where an effective rectangular cell is a cell with nonzero parametric length in at least 1 parametric direction. Examples are the cells A and B in Figure 9A,B. Note that, when defining the cells, the continuity reduction lines are not considered (Figure 9B).

Figure 9 illustrates the algorithm to generate the T-spline mesh $\mathcal{T}^{\alpha+1}$ from $\mathcal{T}^\alpha$. Cell A has a nonzero parametric length in both directions, which leads to 4 congruent cells in $\mathcal{T}^{\alpha+1}$. Cell B has a nonzero parametric length only in the ξ^2 direction. It is divided into 2 congruent cells in $\mathcal{T}^{\alpha+1}$. The cell subdivision is performed by meshline insertions (Figure 9B). The inserted meshline is composed of the middle lines of each effective rectangular cell, and their extensions into the neighbouring cells. Examples are the meshlines ε_1 and ε_2 for cell A, and ε_3 for cell B (Figure 9B). After meshline insertions, the anchors and corresponding local knot vectors are updated using Equation 17. Figure 9C displays the final T-spline mesh $\mathcal{T}^{\alpha+1}$ which is generated from $\mathcal{T}^\alpha$.

We adopt the algorithm of Evans et al[20] and Chen and de Borst[26] to construct the hierarchical T-spline bases. We define N as a T-spline blending function, while $\mathcal{N}$ denotes the T-spline blending function space.[16] The hierarchical T-spline bases $\mathcal{H}$ is built recursively as follows:

(1) Initialisation: $\mathcal{H}^1 = \left\{N \in \mathcal{N}^1 : \text{ supp } N \neq \emptyset\right\}$.

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

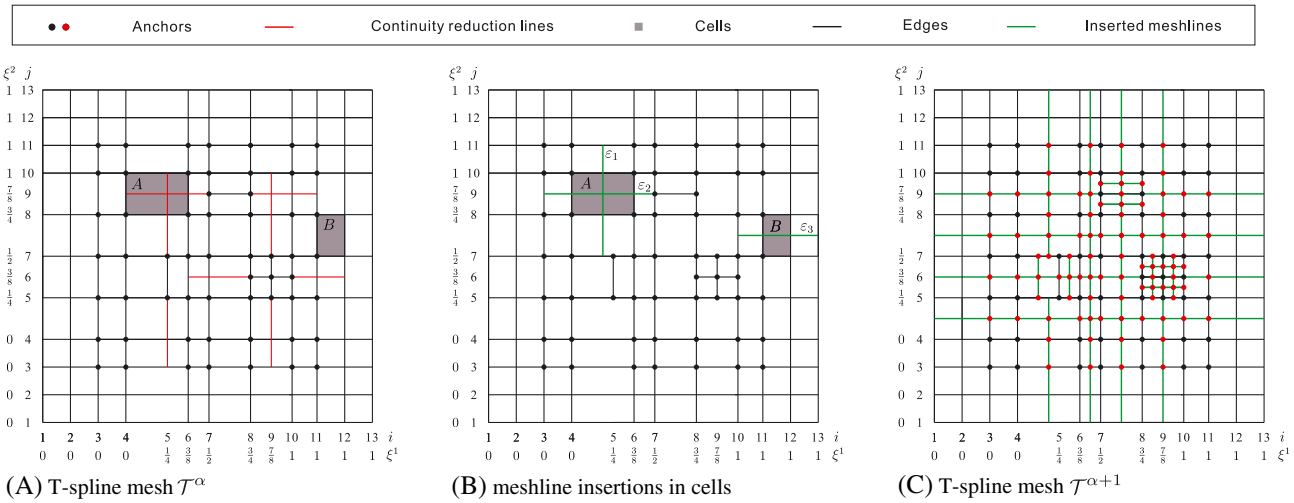


FIGURE 9 Construction of the T-mesh $\mathcal{T}^{\alpha+1}$ from $\mathcal{T}^{\alpha}$. The anchors are indicated by circular dots. Black denotes the anchors on the T-spline mesh $\mathcal{T}^{\alpha}$, while red stands for those generated for the T-spline mesh $\mathcal{T}^{\alpha+1}$

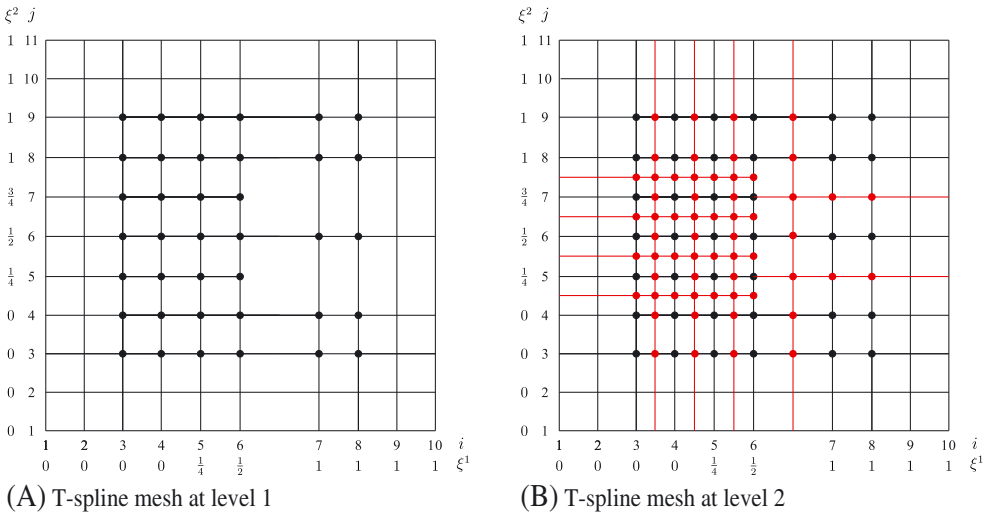


FIGURE 10 Multilevel T-spline mesh with a hierarchy of 3 levels. For a clear representation, only the T-splines meshes at levels 1 and 2 are shown

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

Considering linear combinations between blending functions on hierarchy levels α and $\alpha+1$, we obtain the truncated hierarchical bases.[18,24]

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

To construct these bases, a hierarchy of 3 levels is constructed (Figure 10). An graphical representation of the hierarchical and truncated hierarchical bases for the bivariate case is given in Figures 11 and 12.

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

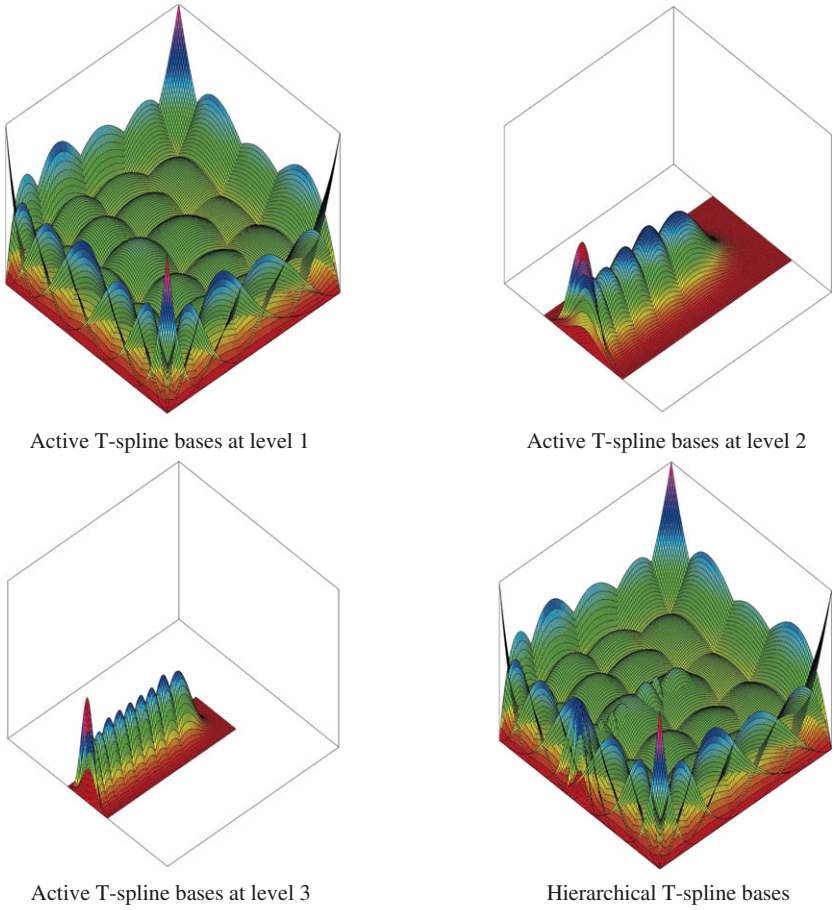


FIGURE 11 Hierarchical T-spline bases

4.2 | Data structure of multilevel mesh

Element-based implementation of hierarchical and truncated hierarchical T-spline bases is a natural choice in adaptive IGA. It consists of T-splines over multiple hierarchical levels with the same polynomial degree. In this subsection, the data structure will be outlined of the multilevel mesh for constructing $\mathcal{H}$ and $\mathcal{H}_T$.

First, we construct a multilevel T-spline mesh with L hierarchy levels. At each hierarchy level, we have n_i anchors, each with a corresponding local knot vector set $\boldsymbol{\Xi}_i = \left\{\boldsymbol{\Xi}_i^j\right\}$ $(i = 1, 2, ...L; j = 1, 2, ...n_i)$. The local knot vector set $\boldsymbol{\Xi}_i$ results from successive uniform cell subdivision within the parameter domain Ω_d, starting from $\boldsymbol{\Xi}_1$. Hence, we obtain nested parameter domains, $\Omega_d^i \subset \Omega_d^{i+1}$ and nested local knot vectors, $\boldsymbol{\Xi}_i \subset \boldsymbol{\Xi}_{i+1}$. Each knot vector set $\boldsymbol{\Xi}_i$ defines a set of T-spline blending functions $\mathbf{N}^i = \left\{N_j^i\right\}_{j=1}^{n_i}$, which in turn forms a nested T-spline approximation space $\mathcal{N}^i$, see Figures 11 and 12.

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

$$\mathbf{N}^i = \mathbf{S}^{i,j}\mathbf{N}^j = \prod_{l=i}^{j-1}\mathbf{S}^{l,l+1}\mathbf{N}^{l+1}, \tag{19}$$

where $\mathbf{S}^{l,l+1}$ is the subdivision or refinement operator.[26] Using Equations 8 and 11, $\mathbf{S}^{l,l+1}$ is obtained using the local knot vector sets $\boldsymbol{\Xi}_l$ and $\boldsymbol{\Xi}_{l+1}$.

Using the subdivision operator, the coordinates and weights of the anchors on the T-spline mesh $\mathcal{T}^i$ at hierarchy level i are computed[26]:

$$\mathbf{P}_w^i = {\mathbf{S}^{i,1}}^{\mathrm{T}}\mathbf{P}_w^1 = \left(\prod_{l=1}^{i}\mathbf{S}^{l,l+1}\right)^{\mathrm{T}}\mathbf{P}_w^1, \tag{20}$$

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

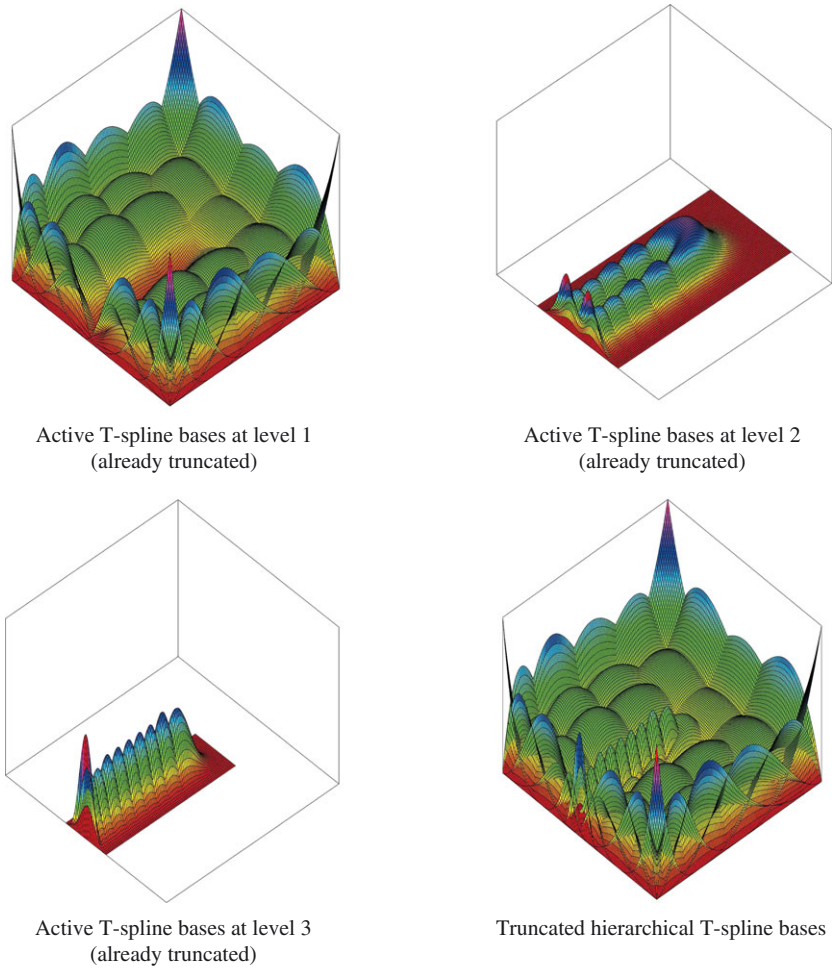


FIGURE 12 Truncated hierarchical T-spline bases

where $\mathbf{P}_w^i$ are weighted control points at level i. Each weighted control point is defined as $\mathbf{P}_{w,j}^i = \left(w_j^i x_{1j}^i,\ w_j^i x_{2j}^i,\ w_j^i\right)$. If we consider a rational T-spline bases, the subdivision operator $\mathbf{S}^{l,l+1}$ in Equation 19 must be modified as[26]

$$\tilde{S}_{IJ}^{l,l+1} = \frac{w_J^l}{w_J^{l+1}} S_{IJ}^{l,l+1}, \tag{21}$$

where w is the weight in Equation 20 and $S_{IJ}^{l,l+1}$ is the term in $\mathbf{S}^{l,l+1}$.

4.3 | Implementation of hierarchical refinement

The hierarchical and truncated hierarchical bases are constructed on the basis of the multilevel mesh, the active elements and the active child elements in the multilevel hierarchy. The active elements are chosen by a marking criterion, such as an a posteriori error estimator.[22] We consider 2 types of hierarchical bases: a standard basis, $\mathcal{H}$, and a truncated basis, $\mathcal{H}_T$, see[26] for details on their construction.

We consider the implementation of adaptive hierarchical refinement in a multilevel adaptive manner. First, using Bézier extraction, the stiffness matrix of active elements at each hierarchy level is obtained, without consideration of multi-level blending function interaction. Having assembled the stiffness matrix at each hierarchy level, the global system of equations is obtained:

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

where $\mathbf{U}$ includes the nodal degrees of freedom at each hierarchy level, $\mathbf{F}$ represents the force vector, $\mathbf{K}$ is a sparse matrix with stiffness matrices $\mathbf{K}^i$ along the diagonal. The submatrix $\mathbf{K}^i$ constitutes the stiffness matrix of the active elements at hierarchy level i, a square sparse matrix of size $2n_c^i \times 2n_c^i$, where n_c^i denotes the number of control points at hierarchy level i.

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

To enforce the interaction between multilevel hierarchical bases $\mathcal{H}$ and $\mathcal{H}_T$ in Equation 22, a hierarchical subdivision operator $\mathbf{M}_h$ is introduced, which yields the following hierarchical system of equations:

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

with $\mathbf{M}_h$ the hierarchical subdivision operator.[26] Solving Equation 23 yields the displacement $\mathbf{U}_h$ for the control points associated with the hierarchical bases. It is noted that in a non-linear solution scheme, the computation of the stiffness matrix $\mathbf{K}$ requires the displacement vector $\mathbf{U}$ rather than $\mathbf{U}_h$ from previous the iteration, see Equation 22.

$$\mathbf{U} = \mathbf{M}_h^{\mathrm{T}}\mathbf{U}_h \tag{24}$$

The full procedure of adaptive hierarchical refinement is provided in Algorithm 2.

Algorithm 2 Adaptive hierarchical refinement

$\mathcal{S}1$ Read the geometry data of the initial T-spline surface to obtain the initial local knot vectors $\left(\Xi_1^1, \Xi_1^2\right)$ and the initial control points $\mathbf{P}_1$.

$\mathcal{S}2$ Conduct successive cell subdivision to generate $\left(\Xi_I^1, \Xi_I^2\right)$ and $\mathbf{P}_I$ for each hierarchy level I from $\left(\Xi_1^1, \Xi_1^2\right)$ and $\mathbf{P}_1$. Meshline insertions are used to obtain $\left(\Xi_I^1, \Xi_I^2\right)$ and $\mathbf{P}_I$.

$\mathcal{S}3$ Compute the subdivision operator $\mathbf{S}^{l,l+1}$ between 2 consecutive hierarchy levels l and $l+1$, according to Equation 19.

$\mathcal{S}4$ Obtain the list of active elements and active child elements. Note that for the first load step, the active elements are directly provided by the initial T-spline mesh.

$\mathcal{S}5$ Compute the subdivision operator $\mathbf{M}_h$, Equation 23.

$\mathcal{S}6$ Solve Equation 23 and use Equation 24 to obtain the displacement vector $\mathbf{U}$.

$\mathcal{S}7$ Check whether elements should be refined and mark them accordingly. If there is no element marked for refinement, stop the calculation for the current load step and go to next load step. Otherwise go to $\mathcal{S}8$.

$\mathcal{S}8$ Obtain the new list of active elements and active child elements on the basis of the marked elements and return to $\mathcal{S}5$.

4.4 | Element based adaptive IGA

Below, we provide a general procedure for adaptive IGA on the basis of these refinement techniques:

- $\mathcal{S}1$ Solve the system of equations to obtain the displacement $\mathbf{U}$. For adaptive local refinement, Equation 22 is to be considered. For adaptive hierarchical refinement, Equations 23 and 24 are used.
- $\mathcal{S}2$ Estimate the approximation error. The H_1 norm of the element-wise residual is used in the examples that follow.
- $\mathcal{S}3$ Mark elements for refinement on the basis of $\mathcal{S}2$.
- $\mathcal{S}4$ Refine the marked elements. The respect refinement procedures are given in Algorithms 1 and 2 for adaptive local refinement and adaptive hierarchical refinement, respectively. If no element needs to be refined, stop the calculation. Otherwise, return to $\mathcal{S}1$.

5 | ELEMENT MARKING

For an admissible mesh $\mathbf{T}$ and E elements, the error per element $\left\{ \varepsilon_Q \middle| Q \in \mathbf{T} \right\} \subset R$ is obtained from step $\mathcal{S}2$. We introduce a marking parameter $\eta \in [0,\ 1]$ to determine whether refinement should be applied, and we define $\Omega = \{Q_1,\ \cdots,\ Q_E\}$ and order $\varepsilon_{Q_1} \geqslant \cdots \geqslant \varepsilon_{Q_E}$. We use quantile marking[12]:

$$\mathcal{M} = \{Q_1,\ \cdots,\ Q_k\} \qquad \text{with} \qquad k = \text{ceil}\,(\eta E)\,, \tag{25}$$

where ceil stands for ceiling function, which rounds up to the nearest integer of ηE. Summing the errors over all the elements gives the domain error, $\varepsilon = \sum_e \varepsilon_e$.

For adaptive hierarchical refinement, element marking continues until the highest hierarchy level is attained. For adaptive local refinement, the element refinement is conducted until a prescribed smallest element size e_m is reached in the parameter domain. To alleviate the effect of over-refinement in the element-based structured mesh refinement strategy, we include the following considerations:

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

- The total area A_m of elements with the smallest element size e_m should satisfy $A_m \leq \eta A$. Here, A denotes the total area of the domain. This relation controls the element marking for refinement in the physical domain.
- The domain error ε^i at refinement step i should satisfy

$$\frac{1}{\zeta} \cdot \frac{n_c^i - n_c^{i-1}}{n_c^{i-1}} \leq \frac{\varepsilon^i - \varepsilon^{i-1}}{\varepsilon^{i-1}}, \tag{26}$$

where ε^{i-1} and ε^i are the domain errors at step $i-1$ and i, respectively, n_c^i and n_c^{i-1} are the corresponding number of control points, and ζ is a constant to control the convergence speed. Equation 26 is an empirical formula to control the convergence speed of refinement. The convergence speed of the domain error should be higher than that of the increase in the number of control points. Furthermore, from our numerical tests it appears that $\zeta = 10$, which reflects that the optimal convergence rate in the error is attained in a logarithmic sense.

Remark 1. To obtain a well-conditioned stiffness matrix $\mathbf{K}_h$ in Equation 23 for the marked elements in adaptive hierarchical refinement, the adjacent elements are forced to be from the same or at most from 2 consecutive hierarchy levels.[23]

Remark 2. To obtain a well-structured mesh in adaptive local refinement, the element-based structured mesh refinement strategy is used for the element refinement, which yields a well-conditioned stiffness matrix $\mathbf{K}$ in Equation 22.

6 | NUMERICAL EXAMPLES

To assess the accuracy of the methodology, we present 2 examples which are considered as benchmark problems in adaptive IGA: a Poisson problem and a linear elasticity with analytical solutions.[16,23] The adaptive refinement techniques (local and hierarchical) are compared numerically, including the condition number of the stiffness matrix. For adaptive local refinement, the element-based structured mesh refinement strategy is used. For adaptive hierarchical refinement, the truncated hierarchical bases $\mathcal{H}_T$ are used to describe the geometry of domain and also to approximate the displacement field.

The error of each element is computed using the H^1 error norm[47]:

$$\|\mathbf{u} - \bar{\mathbf{u}}\|_{H^1(\Omega_e)} = \sqrt{\left(\int_{\Omega_e} (\mathbf{u} - \bar{\mathbf{u}})^{\mathrm{T}} \cdot (\mathbf{u} - \bar{\mathbf{u}}) \; dS + \int_{\Omega_e} (\mathbf{u} - \bar{\mathbf{u}})'^{\mathrm{T}} \cdot (\mathbf{u} - \bar{\mathbf{u}})' \; dS \right)}, \tag{27}$$

where $\mathbf{u}$ stands for the analytical solution, $\bar{\mathbf{u}}$ denotes the approximate solution, and $(\mathbf{u} - \bar{\mathbf{u}})'$ is the derivative of $(\mathbf{u} - \bar{\mathbf{u}})$ with respect to x_1 and x_2, respectively. The domain error is obtained by summing up the element error:

$$\|\mathbf{u} - \bar{\mathbf{u}}\|_{H^1(\Omega)} = \sqrt{\sum\nolimits_e \left(\|\mathbf{u} - \bar{\mathbf{u}}\|_{H^1(\Omega_e)} \right)^2} \tag{28}$$

The relative error of each element, needed for marking elements for refinement, is computed as: ε_e:

$$\varepsilon_e = \frac{\|\mathbf{u} - \bar{\mathbf{u}}\|_{H^1(\Omega_e)}}{\sqrt{\left(\int_{\Omega_e} \mathbf{u}^{\mathrm{T}} \cdot \mathbf{u} \; dS + \int_{\Omega_e} \mathbf{u}'^{\mathrm{T}} \cdot \mathbf{u}' \; dS \right)}}. \tag{29}$$

In general, T-spline meshes are generated by adaptive local refinement of NURBS meshes.[10] Herein, the initial T-spline mesh is directly defined by NURBS meshes. The corresponding initial local knot vectors $\left(\Xi_1^1, \; \Xi_1^2\right)$ and the initial control points $\mathbf{P}_1$ are thus directly obtained from the NURBS meshes. In the examples the, geometry is modelled with the same polynomial degree p in each parametric direction.

We also consider the condition number κ of the stiffness matrix $\mathbf{K}$, which is defined as

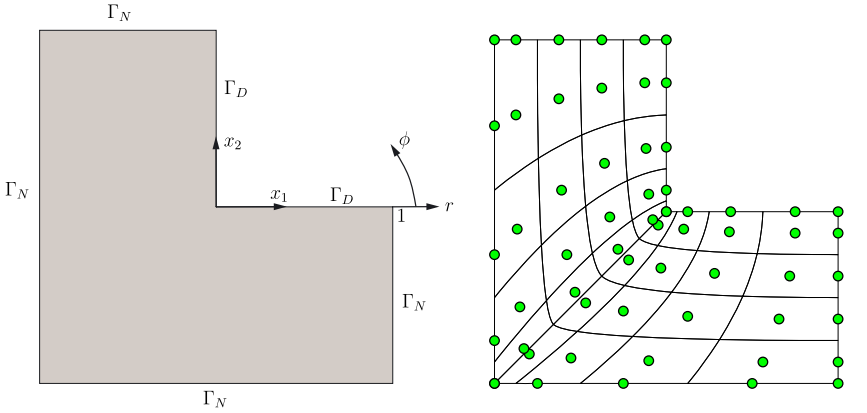
$$\kappa(\mathbf{K}) = \frac{|\lambda_{\max}(\mathbf{K})|}{|\lambda_{\min}(\mathbf{K})|}, \tag{30}$$

where $\lambda_{\max}(\mathbf{K})$ and $\lambda_{\min}(\mathbf{K})$ are the largest and the smallest (by moduli) eigenvalues of $\mathbf{K}$, respectively.

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

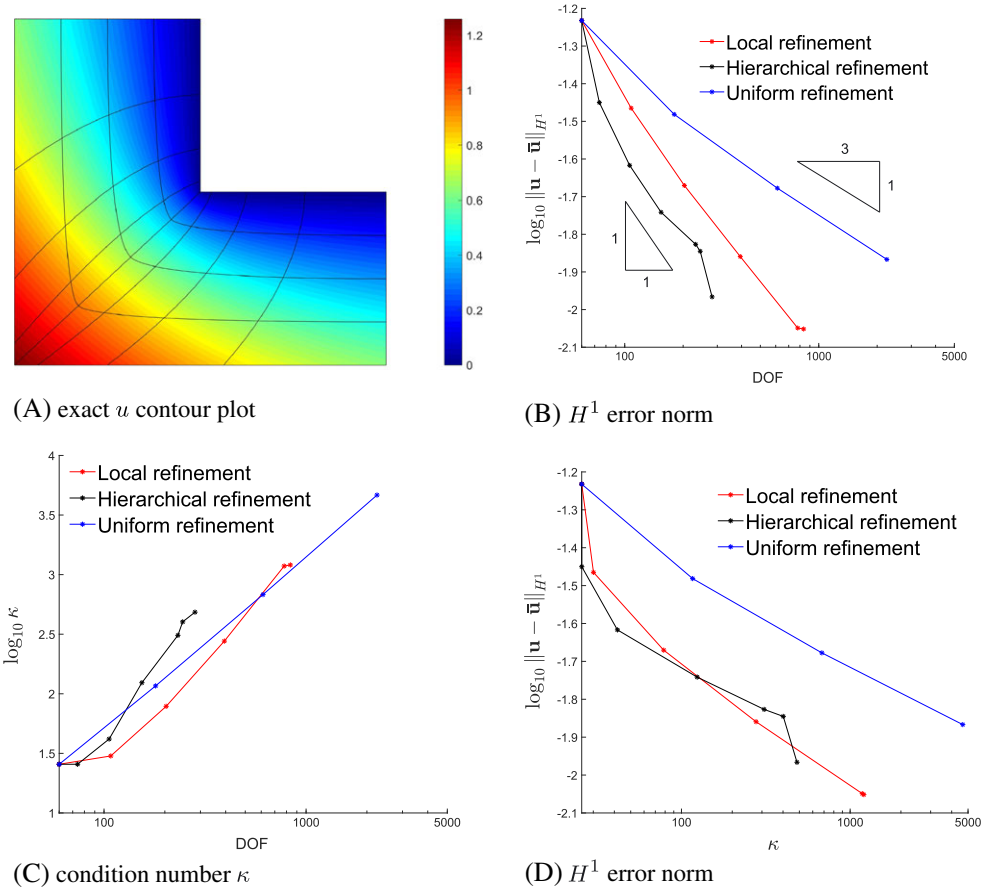
6.1 | Poisson problem on an L-shaped domain

The Poisson problem is now solved for the temperature u on an L-shaped domain (Figure 13A). The L-shaped domain is defined as $\Omega_L = \{(-1,\ 1)\times(-1,\ 1)\}\setminus\{(0,\ 1)\times(0,\ 1)\}$. The Poisson problem is given by the following equation and boundary conditions:



(A) problem definition of the plate (B) Bézier physical mesh and anchors

FIGURE 13 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

(C) condition number κ (D) H^1 error norm

FIGURE 14 Poisson problem on an L-shaped domain: exact solution of u, H^1 error norm, and condition number κ

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$\triangle u = 0, \qquad \frac{\partial \bar{u}}{\partial n} = g \quad \text{on} \quad \Gamma_D, \qquad \bar{u} = 0 \quad \text{on} \quad \Gamma_N, \tag{31}$$

and the exact solution is given by

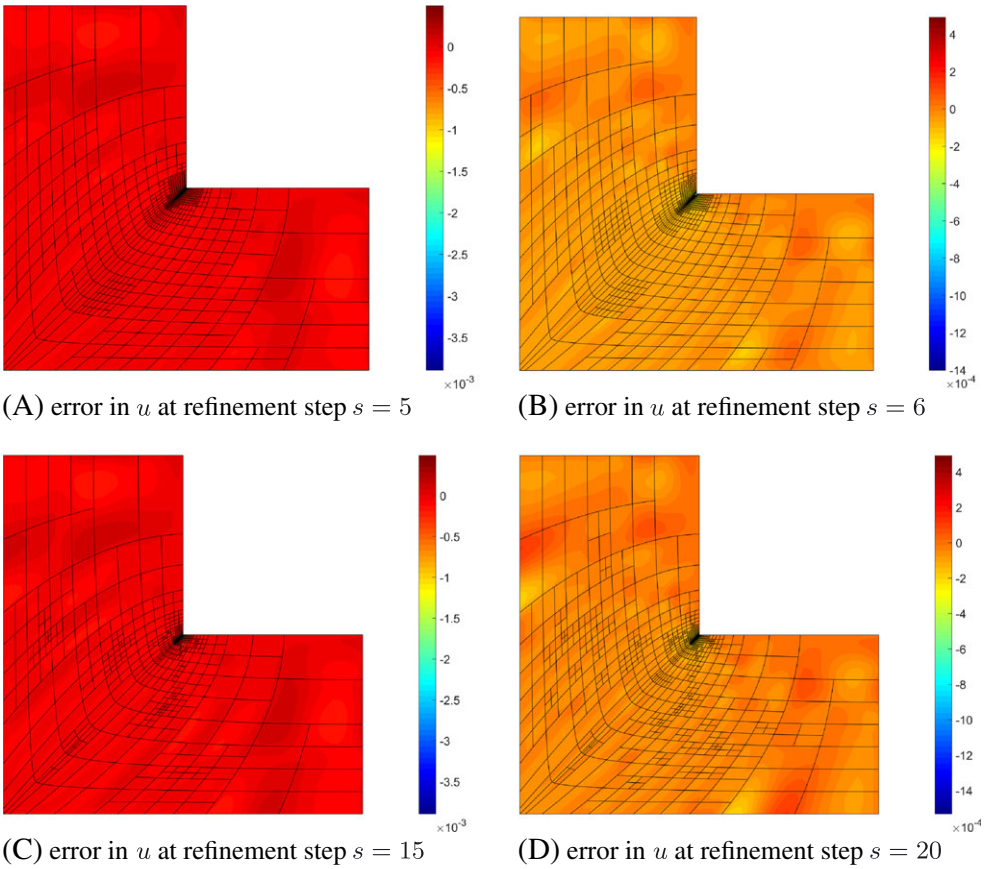
$$\bar{u} = r^{\frac{2}{3}} \sin \frac{2\phi - \pi}{3}, \qquad \phi = (0,\ 2\pi]\,. \tag{32}$$

The L-shaped domain is modelled by a single C^1 continuous B-spline patch (Figure 13B). The domain Ω is discretised by NURBS of polynomial degree $p = 2$ and with knot vectors $\Xi^1 = [0,\ 0,\ 0,\ \frac{1}{8},\ \frac{1}{4},\ \frac{3}{8},\ \frac{1}{2},\ \frac{5}{8},\ \frac{3}{4},\ \frac{7}{8},\ 1,\ 1,\ 1]$ and $\Xi^2 = [0,\ 0,\ 0,\ \frac{1}{4},\ \frac{1}{2},\ \frac{3}{4},\ 1,\ 1,\ 1]$. The corresponding Bézier physical mesh and anchors are shown in Figure 13B. For the construction of the initial T-spline mesh, $\mathcal{T}_1$, the local knot vectors $\left(\Xi_1^1,\ \Xi_1^2\right)$, and the coordinates of anchors are derived from Ξ^1 and Ξ^2. We use adaptive local as well as hierarchical refinement. To provide a good comparison, the smallest element size e_m is prescribed as $e_m = \frac{1}{128}$ for adaptive local refinement, which corresponds to a hierarchy of 5 levels to construct the truncated hierarchical bases in adaptive hierarchical refinement. Elements are refined by adaptive refinement using quantile marking ($\eta = 0.2$).

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

$$k = -\frac{1}{2} \ \min\left(p,\ \frac{\pi}{2\pi - \beta}\right) = -\frac{1}{2} \ \min\left(p,\ \frac{2}{3}\right) = -\frac{1}{3}. \tag{33}$$

For uniform refinement, the corresponding rate of convergence is $k = -1/3$ (Figure 14B). The optimal rate of convergence $k = -1$ could be recovered by adaptive refinement (Figure 14B). It shows that the error level for adaptive refinement is smaller than that for uniform refinement. This is because adaptive refinement smoothens the gradient around the



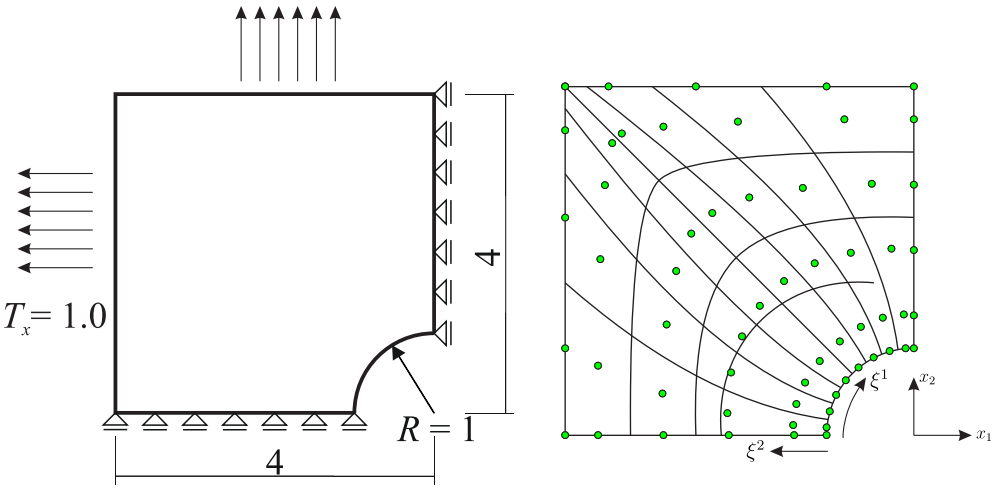
(A) error in u at refinement step $s = 5$ (B) error in u at refinement step $s = 6$

(C) error in u at refinement step $s = 15$ (D) error in u at refinement step $s = 20$

FIGURE 15 Bézier meshes and error in u at each refinement step for quadratic T-spline bases. The Bézier meshes of elements are indicated by solid lines. The error is given as the difference between numerical and exact solutions. A and B are the results for adaptive local refinement, while C and D are those for adaptive hierarchical refinement

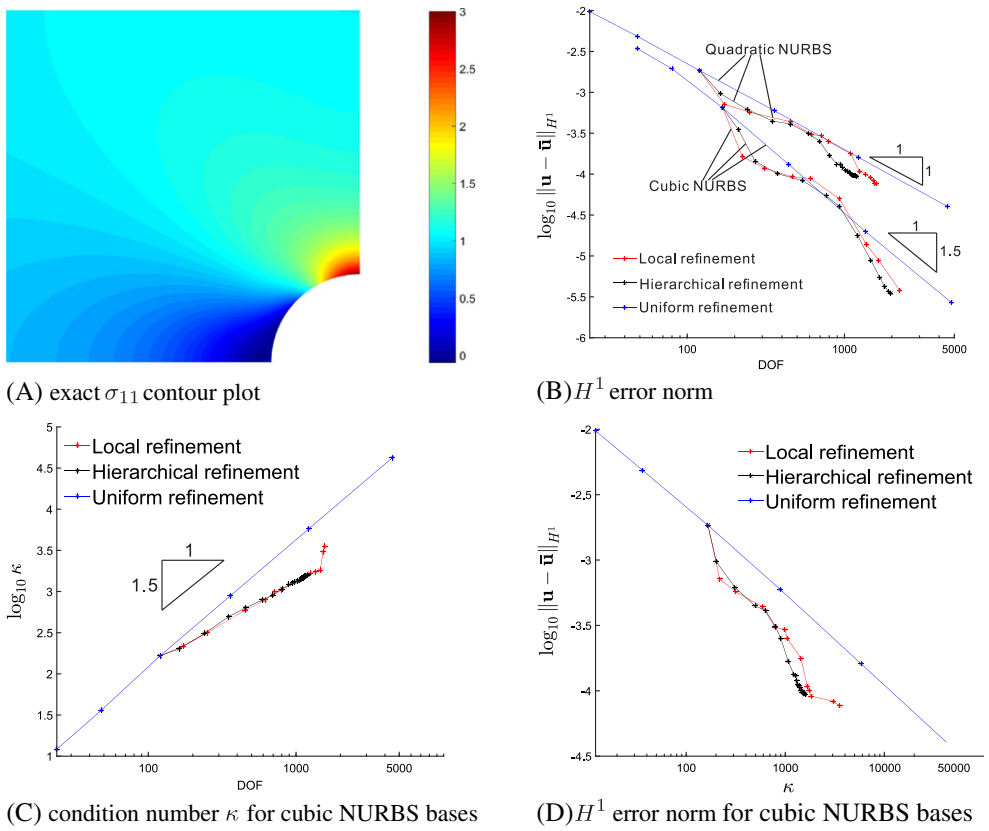
10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

re-entrant corner (Figure 15). From these figures it is observed that the mesh around the re-entrant corner is refined gradually until the smallest element size e_m or the lowest hierarchy level. For adaptive local refinement, the error level is higher than that for adaptive hierarchical refinement, which is due to over-refinement in adaptive local refinement (Figure 14B). If we would have used the full span or the minimum span refinement strategies, the error level would have reduced further.[29]



(A) problem definition of the plate (B) Bézier physical mesh and anchors

FIGURE 16 Linear elasticity: infinite plate with a circular hole—problem definition and initial T-spline mesh $\mathcal{T}_1$ in the physical domain



(A) exact σ_{11} contour plot (B) H^1 error norm

(C) condition number κ for cubic NURBS bases (D) H^1 error norm for cubic NURBS bases

FIGURE 17 Infinite plate with a circular hole: exact solution of σ_{11}, H^1 error norm, and condition number κ

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

As regards the condition number κ, its value is higher for adaptive hierarchical refinement than for uniform refinement, while the value for adaptive local refinement is the smallest (Figure 14C). Figure 14D shows that the level of accuracy for both forms of adaptive refinement increases faster than that for uniform refinement.

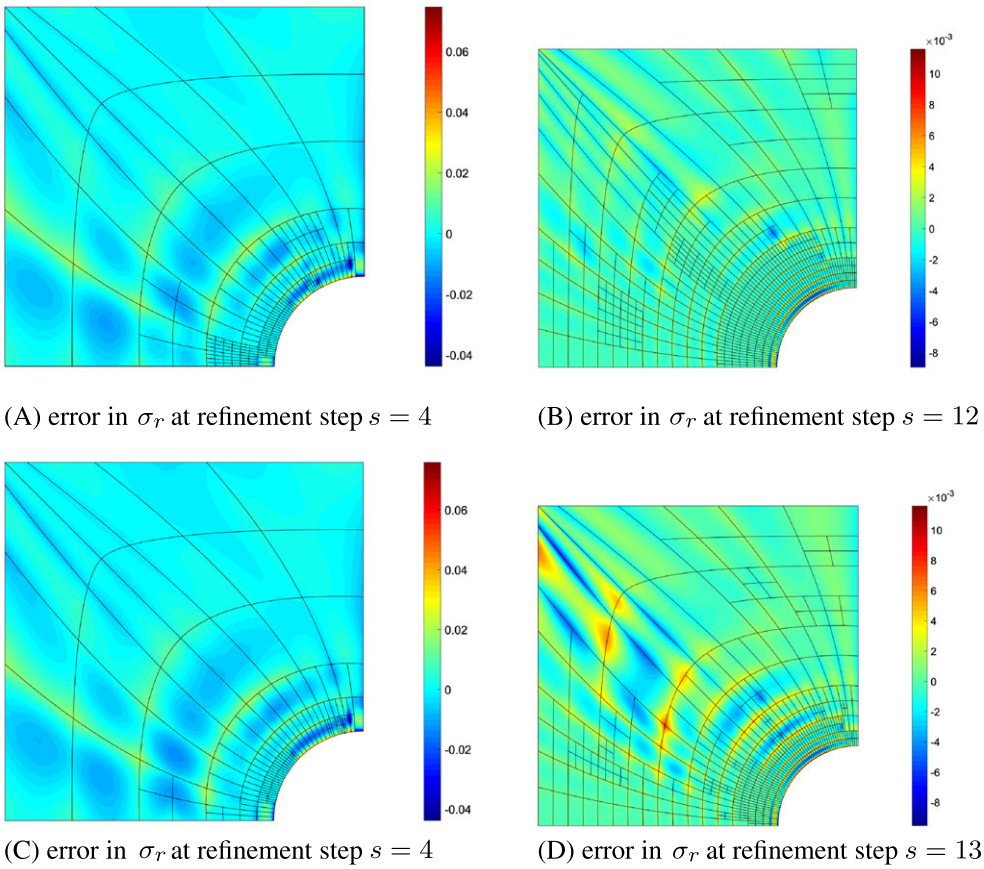
6.2 | Linear elasticity: infinite plate with a circular hole

We next consider an infinite plate with a circular hole (radius $R = 1$ m); see Figure 16A. The material parameters are Young modulus $E = 100\text{N/m}^2$, Poisson ratio $\nu = 0.0$, and the thickness $h = 1$ m. The exact solutions of radial and tangential displacement are

$$\begin{aligned} u_r &= \frac{T_x r\cos(2\theta)}{2E}\left[(1+\nu)+4\frac{R^2}{r^2}-(1+\nu)\frac{R^4}{r^4}\right]+\frac{T_x r}{2E}\left[(1-\nu)+(1+\nu)\frac{R^2}{r^2}\right], \\ u_\theta &= -\frac{T_x r\sin(2\theta)}{2E}\left[(1+\nu)+2(1-\nu)\frac{R^2}{r^2}+(1+\nu)\frac{R^4}{r^4}\right], \end{aligned} \tag{34}$$

where θ is the azimuthal coordinate. From this, the stress components can be derived as

$$\begin{aligned} \sigma_r &= \frac{T_x}{2}\left(1-\frac{R^2}{r^2}\right)+\frac{T_x\cos 2\theta}{2}\left(\frac{3R^4}{r^4}-\frac{4R^2}{r^2}+1\right), \\ \sigma_\theta &= \frac{T_x}{2}\left(1+\frac{R^2}{r^2}\right)-\frac{T_x\cos 2\theta}{2}\left(\frac{3R^4}{r^4}+1\right), \\ \sigma_{r\theta} &= \frac{T_x\sin 2\theta}{2}\left(\frac{3R^4}{r^4}-\frac{2R^2}{r^2}-1\right). \end{aligned} \tag{35}$$



(A) error in σ_r at refinement step $s = 4$ (B) error in σ_r at refinement step $s = 12$

(C) error in σ_r at refinement step $s = 4$ (D) error in σ_r at refinement step $s = 13$

FIGURE 18 Bézier meshes and error in σ_{11} at each refinement step for quadratic NURBS bases. The Bézier meshes of elements are indicated by solid lines. The error is given as the difference between the numerical solution and the exact solution. A and B are the results for adaptive local refinement, while C and D are those for adaptive hierarchical refinement

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

By virtue of symmetry, only a quarter of the plate has to be modelled (Figure 16A). The exact traction from the analytical solution is imposed at the free boundary, e.g., Klinkel et al and Chen et al.[48,49] The domain Ω is discretised by NURBS of a polynomial degree $p = 2$, with knot vectors $\boldsymbol{\Xi}^1 = [0,\ 0,\ 0,\ \frac{1}{8},\ \frac{1}{4},\ \frac{3}{8},\ \frac{1}{2},\ \frac{5}{8},\ \frac{3}{4},\ \frac{7}{8},\ 1,\ 1,\ 1]$ and $\boldsymbol{\Xi}^2 = [0,\ 0,\ 0,\ \frac{1}{4},\ \frac{1}{2},\ \frac{3}{4},\ 1,\ 1,\ 1]$. These knot vectors have been obtained by h refinement of open knot vectors.[1] Accordingly, the number of control points $\mathbf{P}$ has been adapted. The corresponding Bézier physical mesh and anchors are given in Figure 16B. We consider quadratic and cubic NURBS bases to discretise the domain. For cubic NURBS bases, the knot vector and control points are obtained by order elevation from quadratic NURBS bases. For the construction of the initial T-spline mesh, $\mathcal{T}_1$, the local knot vectors $\left(\boldsymbol{\Xi}_1^1,\ \boldsymbol{\Xi}_1^2\right)$ and the coordinates of anchors are derived from $\boldsymbol{\Xi}^1$, $\boldsymbol{\Xi}^2$ and $\mathbf{P}$.

We use both adaptive local refinement and adaptive hierarchical refinement. To provide a fair comparison, the smallest element size e_m is prescribed as $e_m = \frac{1}{64}$ in adaptive local refinement, which corresponds to a hierarchy of 4 levels to construct truncated hierarchical bases in adaptive hierarchical refinement. Elements are refined by adaptive refinement using quantile marking ($\eta = 0.2$).

The solution for this problem has a stress concentration at $(x_1,\ x_2) = (0,\ 1)$, see Figure 17A, but no singularity. Hence, an optimal rate of convergence $k = -p/2$ in the H^1 norm can be attained by uniform refinement; see Figure 17B.

For adaptive refinement, the optimal convergence rate is obtained in the asymptotic limit.[23] In general, the error level for adaptive refinement is lower than that for uniform refinement because adaptive refinement smoothens the stress gradient. This is illustrated in Figure 18, which shows that the mesh around the hole is refined until the smallest element size or the lowest hierarchy level. With adaptive refinement, the error level is reduced for the whole domain, which indicates that adaptive refinement not only efficiently models the local stress concentration but also improve the global accuracy. The figure also shows that the refinement area is almost same for both cases of adaptive refinement. The error level for adaptive local refinement is generally higher than that for adaptive hierarchical refinement, again due to over-refinement in the element-based structured mesh refinement strategy.

The condition number κ increases with a rise in the number of degrees of freedom (Figure 17C). We observe that κ is almost the same for both cases of adaptive refinement. In Figure 17D, the H^1 error norm is plotted versus the condition number κ. For adaptive refinement, the accuracy increases faster than that for uniform refinement without a significant increase in the condition number.

7 | CONCLUDING REMARKS

Two adaptive refinement techniques, local refinement and hierarchical refinement, have been developed for adaptive IGA. The refinement techniques have been cast in the framework of Bézier extraction, which conforms ideally to the element point of view in the traditional finite element method. A detailed description is given how the approach can be implemented in a conventional finite element data structure. Algorithms have been provided for both refinement techniques.

Two examples have been studied numerically and lead to the conclusion that for adaptive refinement, the optimal rate of convergence is obtained in the asymptotic limit, irrespective of potential singularities. Moreover, the error level for adaptive refinement is lower than that for uniform refinement. For adaptive hierarchical refinement, the condition number of the stiffness matrix is slightly higher than that for adaptive local refinement but remains below that for uniform refinement, at least for the linear elasticity problem.

ACKNOWLEDGEMENTS

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

ORCID

R. de Borst http://orcid.org/0000-0002-3457-3574

REFERENCES

1. Hughes TJR, Cottrell JA, Bazilevs Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput Methods Appl Mech Eng*. 2005;194:4135-4195.
2. Cottrell JA, Hughes TJR, Bazilevs Y. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Chichester: John Wiley & Sons; 2009.

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

3. Sederberg TW, Zheng J, Bakenov A, Nasri A. T-splines and T-NURCCs. *ACM Trans Graphics*. 2003;22:477-484.
4. Sederberg TW, Cardon DL, Finnigan GT, North NS, Zheng J, Lyche T. T-spline simplification and local refinement. *ACM Trans Graphics*. 2004;23:276-283.
5. Bazilevs Y, Calo VM, Cottrell JA, et al. Isogeometric analysis using T-splines. *Comput Methods Appl Mech Eng*. 2010;199:229-263.
6. Scott MA, Borden MJ, Verhoosel CV, Sederberg TW, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of T-splines. *Int J Numer Methods Eng*. 2011;88:126-156.
7. Li X, Zheng J, Sederberg TW, Hughes TJR, Scott MA. On linear independence of T-spline blending functions. *Comput Aided Geom Des*. 2012;29:63-76.
8. Morgenstern P, Peterseim D. Analysis-suitable adaptive T-mesh refinement with linear complexity. *Comput Aided Geom Des*. 2015;34:50-66.
9. 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. *Int J Numer Methods Eng*. 2015;103:547-581.
10. Dörfel MR, Jüttler B, Simeon B. Adaptive isogeometric analysis by local h-refinement with T-splines. *Comput Methods Appl Mech Eng*. 2010;199:264-275.
11. Scott MA, Li X, Sederberg TW, Hughes TJR. Local refinement of analysis-suitable T-splines. *Comput Methods Appl Mech Eng*. 2012;213:206-222.
12. Hennig P, Kästner M, Morgenstern P, Peterseim D. Adaptive mesh refinement strategies in isogeometric analysis—a computational comparison. *Comput Methods Appl Mech Eng*. 2017;316:424-448.
13. Li X, Deng J, Chen F. Surface modeling with polynomial splines over hierarchical T-meshes. *The Visual Computer*. 2007;23:1027-1033.
14. Deng J, Chen F, Li X, et al. Polynomial splines over hierarchical T-meshes. *Graphical Models*. 2008;70:76-86.
15. Nguyen-Thanh N, Nguyen-Xuan H, Bordas SPA, Rabczuk T. Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids. *Comput Methods Appl Mech Eng*. 2011;200:1892-1908.
16. Vuong A-V, Giannelli C, Jüttler B, Simeon B. A hierarchical approach to adaptive local refinement in isogeometric analysis. *Comput Methods Appl Mech Eng*. 2011;200:3554-3567.
17. Schillinger D, Dede L, Scott MA, et al. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Comput Methods Appl Mech Eng*. 2012;249:116-150.
18. Giannelli C, Jüttler B, Speleers H. THB-splines: the truncated basis for hierarchical splines. *Comput Aided Geom Des*. 2012;29:485-498.
19. Kuru G, Verhoosel CV, van der Zee KG, van Brummelen EH. Goal-adaptive isogeometric analysis with hierarchical splines. *Comput Methods Appl Mech Eng*. 2014;270:270-292.
20. Evans EJ, Scott MA, Li X, Thomas DC. Hierarchical T-splines: analysis-suitability, Bézier extraction, and application as an adaptive basis for isogeometric analysis. *Comput Methods Appl Mech Eng*. 2015;284:1-20.
21. Verhoosel CV, Van Zwieten GJ, Van Rietbergen B, De Borst R. Image-based goal-oriented adaptive isogeometric analysis with application to the micro-mechanical modeling of trabecular bone. *Comput Methods Appl Mech Eng*. 2015;284:138-164.
22. Buffa A, Giannelli C. Adaptive isogeometric methods with hierarchical splines: error estimator and convergence. *Math Models Methods Appl Sci*. 2016;26:1-25.
23. Hennig P, Müller S, Kästner M. Bézier extraction and adaptive refinement of truncated hierarchical NURBS. *Comput Methods Appl Mech Eng*. 2016;305:316-339.
24. Wei X, Zhang Y, Liu L, Hughes TJR. Truncated T-splines: fundamentals and methods. *Comput Methods Appl Mech Eng*. 2017;316:349-372.
25. Hiemstra R, Calabrò F, Schillinger D, Hughes TJR. Optimal and reduced quadrature rules for tensor product and hierarchically refined splines in isogeometric analysis. *Comput Methods Appl Mech Eng*. 2017;316:966-1004.
26. Chen L, de Borst R. Adaptive refinement of hierarchical T-splines. *Comput Methods Appl Mech Eng*. 2017. submitted.
27. Dokken T, Lyche T, Pettersen KF. Polynomial splines over locally refined box-partitions. *Comput Aided Geom Des*. 2013;30:331-356.
28. Bressan A. Some properties of LR-splines. *Comput Aided Geom Des*. 2013;30:778-794.
29. Johannessen KA, Kvamsdal T, Dokken T. Isogeometric analysis using LR B-splines. *Comput Methods Appl Mech Eng*. 2014;269:471-514.
30. Johannessen KA, Kumar M, Kvamsdal T. Divergence-conforming discretization for Stokes problem on locally refined meshes using LR B-splines. *Comput Methods Appl Mech Eng*. 2015;293:38-70.
31. Johannessen KA, Remonato F, Kvamsdal T. On the similarities and differences between classical hierarchical, truncated hierarchical and LR B-splines. *Comput Methods Appl Mech Eng*. 2015;291:64-101.
32. Kumar M, Kvamsdal T, Johannessen KA. Simple a posteriori error estimators in adaptive isogeometric analysis. *Comput Math Appl*. 2015;70:1555-1582.
33. Kumar M, Kvamsdal T, Johannessen KA. Superconvergent patch recovery and a posteriori error estimation technique in adaptive isogeometric analysis. *Comput Methods Appl Mech Eng*. 2017;316:1086-1156.
34. 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/s00466-017-1455-7
35. Chen L, de Borst R. Locally refined T-splines. *Int J Numer Methods Eng*. 2017. submitted.
36. Borden MJ, Scott MA, Evans JA, Hughes TJR. Isogeometric finite element data structures based on Bézier extraction of NURBS. *Int J Numer Methods Eng*. 2011;87:15-47.
37. Cox MG. The numerical evaluation of B-splines. *IMA J Appl Math*. 1972;10:134-149.

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

38. de Boor C. On calculating with B-splines. *J Approximation Theory*. 1972;6:50-62.
39. Chen L, Lingen FJ, de Borst R. Adaptive hierarchical refinement of NURBS in cohesive fracture analysis. *Int J Numer Methods Eng*. 2017. https://doi.org/10.1002/nme.5600.
40. Forsey DR, Bartels RH. Hierarchical B-spline refinement. *ACM Siggraph Computer Graphics*. 1988;22:205-212.
41. Höllig K, Reif U, Wipper J. Weighted extended B-spline approximation of Dirichlet problems. *SIAM J Numer Anal*. 2001;39:442-462.
42. Bornemann PB, Cirak F. A subdivision-based implementation of the hierarchical B-spline finite element method. *Comput Methods Appl Mech Eng*. 2013;253:584-598.
43. Scott MA, Thomas DC, Evans EJ. Isogeometric spline forests. *Comput Methods Appl Mech Eng*. 2014;269:222-264.
44. Jiang W, Dolbow JE. Adaptive refinement of hierarchical B-spline finite elements with an efficient data transfer algorithm. *Int J Numer Methods Eng*. 2015;102:233-256.
45. Giannelli C, Jüttler B, Speleers H. Strongly stable bases for adaptively refined multilevel spline spaces. *Adv Comput Math*. 2014;40:459-490.
46. Buffa A, Giannelli C. Adaptive isogeometric methods with hierarchical splines: error estimator and convergence. *Math Models Methods Appl Sci*. 2016;26:1-25.
47. Chen L, Dornisch W, Klinkel S. Hybrid collocation—Galerkin approach for the analysis of surface represented 3D-solids employing SB-FEM. *Comput Methods Appl Mech Eng*. 2015;295:268-289.
48. Klinkel S, Chen L, Dornisch W. A NURBS based hybrid collocation—Galerkin method for the analysis of boundary represented solids. *Comput Methods Appl Mech Eng*. 2015;284:689-711.
49. Chen L, Simeon B, Klinkel S. A NURBS based Galerkin approach for the analysis of solids in boundary representation. *Comput Methods Appl Mech Eng*. 2016;305:777-805.

How to cite this article: de Borst R, Chen L. The role of Bézier extraction in adaptive isogeometric analysis: Local refinement and hierarchical refinement. *Int J Numer Meth Engng*. 2018;113:999–1019. https://doi.org/10.1002/nme.5696

10970207, 2018, 6, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/nme.5696 by University Of Sheffield, Wiley Online Library on [09/11/2023]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License