ICES REPORT 16-02

January 2016

Truncated T-splines: Fundamentals and Methods

by

Xiaodong Wei, Yongjie Zhang, Lei Liu, Thomas J.R. Hughes

Reference: Xiaodong Wei, Yongjie Zhang, Lei Liu, Thomas J.R. Hughes, "Truncated T-splines: Fundamentals and Methods ," ICES REPORT 16-02, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, January 2016.

Truncated T-splines: Fundamentals and Methods

Xiaodong Wei[a], Yongjie Zhang[a,*], Lei Liu[a], Thomas J.R. Hughes[b]

[a]*Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[b]*Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, USA*

Abstract

In this paper, we present *Truncated T-splines* as a new type of T-splines suitable for both geometric design and analysis, supporting highly localized refinement. Truncated T-spline basis functions are piece-wise polynomials that are linearly independent and form a partition of unity. Refinement of truncated T-splines produces nested spline spaces. Furthermore, we study truncated T-splines and local refinement on the general domain (2-manifold) with extraordinary points in the T-mesh. G^1 continuity is attained around extraordinary points by properly capping quartic Bézier patches, where a constrained optimization problem is solved. In the end, we study benchmark problems using truncated T-splines in the context of isogeometric analysis. We also apply truncated T-splines to complex geometries to show the smooth surfaces and simulation results under local refinement.

Keywords: Truncation Mechanism, T-splines, Local Refinement, Analysis-Suitability, Isogeometric Analysis

1. Introduction

To bridge the gap between modern geometric design and engineering simulation, isogeoemtric analysis was first introduced in [10] and later elaborated in [3]. The root idea of isogeometric analysis is to utilize the same basis for both geometric representation and analysis. As the current industrial standard in the CAD (Computer Aided Design) community, NURBS (Non-Uniform Rational B-Spline) was introduced as the first type of basis in isogeometric analysis. However, NURBS does not support local refinement for adaptive analysis. Various spline methods were then developed to support local refinement, including T-splines [19, 18], analysis-suitable T-splines [16], LR-splines [5, 11] and weighted T-splines [14]. Hierarchical meshes were also used for local refinement, such as HB-spines [21, 1], PHT-splines [4], THB-splines [8], hierarchical analysis-suitable T-splines [7] and truncated hierarchical Catmull-Clark subdivision [24, 23].

T-splines [19] break down the global tensor product structure of NURBS, and thus allow so-called *T-junctions*, which are analogous to hanging nodes in the traditional finite element method (FEM). Later, Buffa *et al.* showed that not all T-splines possess linearly independent blending functions [2]. Motivated by this fact, Scott *et al.* developed analysis-suitable T-splines [16], providing a sufficient condition to ensure linear independence. However, additional topological constraints are required such that T-junctions have no mutual influence, which may induce refinement propagation far beyond the region of interest. To improve the refinement locality, weighted T-splines [14] were developed by computing a new weight for each basis function instead of manipulating T-meshes. However, refinement of weighted T-splines slightly changes the geometry and cannot guarantee nested spline spaces.

In a T-spline representation, the local knot vectors are ambiguous when encountering an extraordinary point[1]. T-NURCCs, a generalized version of Catmull-Clark subdivision with non-uniform knot intervals, were used for elements surrounding extraordinary points [19]. Zero-knot-interval edges were inserted around

*Corresponding author: Yongjie Zhang. Tel: (412) 268-5332; Fax: (412) 268-3348; Email: *jessicaz@andrew.cmu.edu*.
[1]An extraordinary point is an inner point (not a T-junction) shared by other than 4 quadrilaterals.

each extraordinary point, and these points are treated as boundaries to obtain knot intervals [22]. In [17], basis functions around extraordinary points are defined as a linear combination of quartic Bernstein polynomials. Moreover, via solving a constrained optimization problem to adjust the coefficients of Bernstein polynomials, G^1 continuity is attained. Local knot vectors can also be defined by repeating knot intervals when encountering an extraordinary point [13]. By minimizing a distortion functional, an alternative G^1 construction with Bézier patches was developed. In [17, 13], no other extraordinary points or T-junctions are allowed within the three-ring neighborhood of a given extraordinary point.

In this paper we propose a new method, namely *truncated T-spline*, for a highly localized refinement while preserving desired properties for geometric design and isogeometric analysis. In the viewpoint of geometric design, truncated T-splines form a non-negative partition of unity, and thus possess the convex hull property. Truncated T-splines also preserve the geometry during refinement. In analysis, truncated T-spline basis functions are piece-wise polynomials. On the regular domain without extraordinary points, they are proved to be (globally) linearly independent and to produce nested spline spaces. Compared to analysis-suitable T-splines [16], these properties are achieved with less refinement propagation. Moreover, we apply truncated T-splines to general 2-manifold domains with extraordinary nodes. We couple the knot interval repeating method [13] with optimization [17], enabling T-junctions or other extraordinary points present in the 2^{nd}-ring boundary of a given extraordinary point. Several benchmark problems are tested to demonstrate the analysis-suitability and refinement locality of the proposed method.

The remainder of this paper is organized as follows. Section 2 reviews the basic concepts of T-splines and their analysis-suitability. We then develop truncated T-splines in Section 3, and generalize to 2-manifold domains with extraordinary points in Section 4. Several numerical examples are studied in Section 5. Section 6 concludes the paper and briefly comments on future directions.

2. T-splines and Analysis-Suitability

We first review related concepts, including T-splines with local refinement and their analysis-suitability. Please refer to [19, 18, 16] for details.

2.1. T-splines and Refinability

T-splines are generalized from B-splines. A uni-variate B-spline is defined on a knot vector $\Xi = \{u_1, u_2, \ldots, u_{m+p+1}\}$, where $u_k \in \mathbb{R}$ is the k-th knot ($k = 1, 2, \ldots, m+p+1$), p is the polynomial degree and m is the number of B-spline basis functions. With the knot vector, B-spline basis functions can be obtained using the Cox-de Boor recursion formula [3]. We denote the i-th basis function of degree p as $N_{i,p}(u)$, where $i = 1, \ldots, m$ and $u \in [u_1, u_{m+p+1}]$. Note that $N_{i,p}(u)$ is non-zero only on the open domain $]u_i, u_{i+p+1}[$, and its support is defined as $suppN_{i,p}(u) := [u_i, u_{i+p+1}]$. Given two knot vectors $\Xi = \{u_1, u_2, \ldots, u_{m+p+1}\}$ and $\mathcal{H} = \{v_1, v_2, \ldots, v_{n+q+1}\}$ in two directions respectively, a bi-variate B-spline basis function is obtained as a tensor product of two uni-variate B-splines, that is, $B_{ij,pq}(u, v) = N_{i,p}(u)N_{j,q}(v)$. The local support of $B_{ij,pq}(u, v)$ is $suppB_{ij,pq}(u, v) := [u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}]$. Together with the given control mesh, the B-spline surface is defined as

$$S(u, v) = \sum_{i=1}^{m} \sum_{j=1}^{n} B_{ij,pq}(u, v)P_{ij}, \tag{1}$$

where P_{ij} are the control points.

A T-spline control mesh or *T-mesh* is a quadrilateral mesh in the physical space where T-junctions are allowed. In the parametric space, the preimage of a control point is called a *node*, whereas the preimage of a quadrilateral is called an *element*. Besides, each edge has a parametric length called the *knot interval*. It is required that the sum of knot intervals on opposite edges of any element must be equal. Fig. 1 shows the preimage of a local T-mesh, and A is a T-junction (the red dot). Unlike B-splines, T-spline basis functions are defined on *local knot vectors* inferred from the T-mesh by shooting rays in two parametric directions. For instance, to extract local knot vectors of A, we shoot a ray (the green horizontal line) and obtain two intersections on the left and two intersections on the right. By collecting these u coordinates in

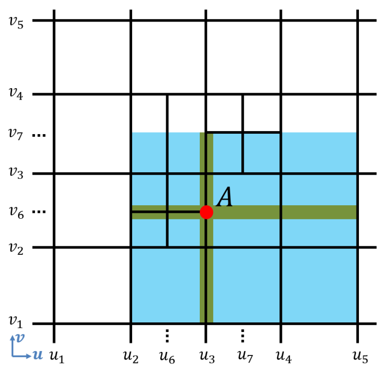


Figure 1: T-mesh, local knot vector extraction of the T-junction A (the red dot) and the support of B_A (the blue region).

a non-decreasing order, we obtain the local knot vector in the u direction, $\Xi_A = \{u_2, u_6, u_3, u_4, u_5\}$. Note that in this paper we only consider bicubic T-splines. Likewise, the local knot vector in the v-direction is $\mathcal{H}_A = \{v_1, v_2, v_6, v_3, v_7\}$. Using the Cox-de Boor formula [3], a uni-variate B-spline basis function can be obtained for each local knot vector, denoted as $N_{\Xi_A}(u)$ and $N_{\mathcal{H}_A}(v)$, respectively. The T-spline basis function associated with A is a tensor-product of $N_{\Xi_A}(u)$ and $N_{\mathcal{H}_A}(v)$, and we have $B_A(u, v) = N_{\Xi_A}(u)N_{\mathcal{H}_A}(v)$ whose support is $[u_2, u_5] \times [v_1, v_7]$; see the blue region in Fig. 1.

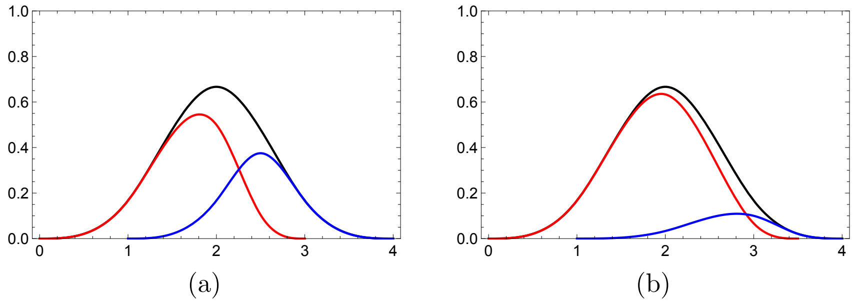
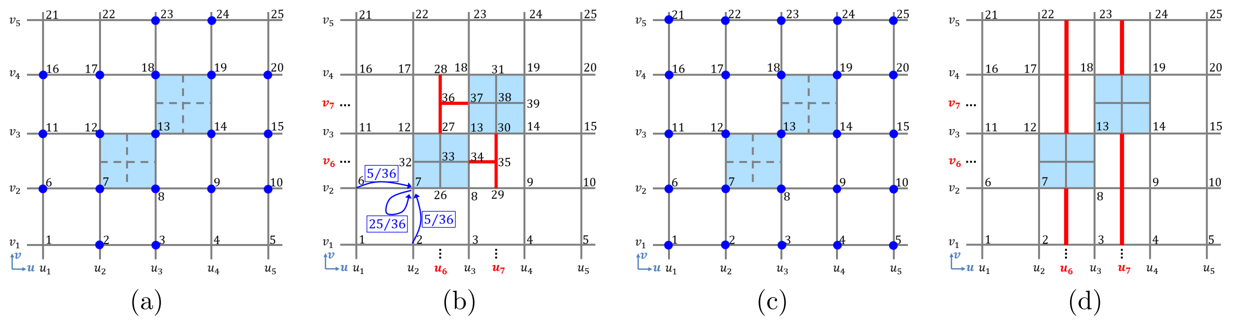


Figure 2: Refinability of a uni-variate B-spline basis function. (a) $N_{\Xi_A}(u)$ (the black curve), $\frac{5}{6}N_{\Xi'_1}(u)$ (the red curve) and $\frac{1}{2}N_{\Xi'_2}(u)$ (the blue curve) where $\Xi_A = \{0, 1, 2, 3, 4\}$, $\Xi'_1 = \{0, 1, 2, \frac{5}{2}, 3\}$ and $\Xi'_2 = \{1, 2, \frac{5}{2}, 3, 4\}$; and (b) $N_{\Xi_A}(u)$ (the black curve), $N_{\Xi'_1}(u)$ (the red curve) and $\frac{1}{6}N_{\Xi'_2}(u)$ (the blue curve) where $\Xi_A = \{0, 1, 2, 3, 4\}$, $\Xi'_1 = \{0, 1, 2, 3, \frac{7}{2}\}$ and $\Xi'_2 = \{1, 2, 3, \frac{7}{2}, 4\}$.

The knot insertion algorithm [19] is employed for T-spline refinement in a local manner. We first study the refinement of a uni-variate basis function, e.g., $N_{\Xi_A}(u)$ where $\Xi_A = \{u_{A,1}, u_{A,2}, u_{A,3}, u_{A,4}, u_{A,5}\}$. By inserting n ($n \in \mathbb{Z}^+$) knots into Ξ_A, we have an enlarged knot vector $\bigcup_{i=1}^{n+1} \Xi'_i = \{u'_{A,1} = u_{A,1}, u'_{A,2}, \ldots, u'_{A,n+5} = u_{A,5}\} \supset \Xi_A$, where $\Xi'_i = \{u'_{A,i}, u'_{A,i+1}, \ldots, u'_{A,i+4}\}$ ($i = 1, \ldots, n+1$) are local knot vectors with enriched knots. Then $N_{\Xi_A}(u)$ can be represented by a linear combination of basis functions defined on Ξ'_i, that is,

$$N_{\Xi_A}(u) = \sum_{i=1}^{n+1} c_i N_{\Xi'_i}(u), \tag{2}$$

where $c_i \in \mathbb{R}$ are the refinement coefficients from knot insertion [19]. Eq. (2) also refers to the *refinability* relationship. For convenience we call $N_{\Xi'_i}(u)$ the *children* of $N_{\Xi_A}(u)$, denoted as $N_{\Xi'_i} \in chd(N_{\Xi_A})$. For instance in Fig. 2(a), a B-spline basis function (the black curve) is defined on a knot vector $\Xi_A = \{0, 1, 2, 3, 4\}$. By inserting a knot $\frac{5}{2}$ we have two children basis functions defined on $\Xi'_1 = \{0, 1, 2, \frac{5}{2}, 3\}$ and $\Xi'_2 = \{1, 2, \frac{5}{2}, 3, 4\}$, respectively; see the red and blue curves. With the refinement coefficients obtained from knot insertion, we have $N_{\Xi_A}(u) = \frac{5}{6}N_{\Xi'_1}(u) + \frac{1}{2}N_{\Xi'_2}(u)$. As another example, we have $N_{\Xi_A}(u) = N_{\Xi'_1}(u) + \frac{1}{6}N_{\Xi'_2}(u)$ by inserting



v_1 v_2 v_3 v_4 v_5 v_6 v_7
u_1 u_2 u_3 u_4 u_5 u_6 u_7
5/36
25/36
5/36
(a)
(b)
(c)
(d)

Figure 3: T-spline refinement (a, b) and analysis-suitable T-spline refinement (c, d). (a, c) A given T-mesh $\mathbb{T}$; and (b, d) the refined T-mesh $\mathbb{T}'$ with additional edges (red edges). The local knot vectors of the blue dots in (a, c) are influenced by the local refinement in (b, d), respectively.

$\frac{7}{2}$ into Ξ_A; see the red and blue curves for the weighted children in Fig. 2(b).

For a bi-variate T-spline basis function, $B_A(u,v) = N_{\Xi_A}(u)N_{\mathcal{H}_A}(v)$, the refinement of $B_A(u,v)$ involves refinement of $N_{\Xi_A}(u)$ and/or $N_{\mathcal{H}_A}(v)$. It is required that each T-spline basis function can be represented as a linear combination of its children basis functions defined on the refined T-mesh. Such requirement ensures nested spline spaces. Given a T-mesh (solid edges denoted as $\mathbb{T}$) shown in Fig. 3(a) with uniform knot interval, when we refine it using the dashed lines, additional refinements are required to ensure nestedness [18]; see the red edges in Fig. 3(b). We denote $B'_i(u,v) = N_{\Xi'_i}(u)N_{\mathcal{H}'_i}(v)$ as a basis function defined on the refined T-mesh $\mathbb{T}'$, where Ξ'_i and $\mathcal{H}'_i$ are local knot vectors inferred from $\mathbb{T}'$. The local knot vectors associated with the blue dots in Fig. 3(a) are all influenced by the refinement, among which we pick $B_6(u,v) = N_{\Xi_6}(u)N_{\mathcal{H}_6}(v)$ as an example to study its refinability. When inserting the knot u_6 into Ξ_6, we have an enlarged knot vector $\Xi'_6 \cup \Xi'_7$. Based on the refinability relationship, we have

$$\begin{aligned}
B_6(u,v) &= \left\{ N_{\Xi'_6}(u) + \frac{1}{6}N_{\Xi'_7}(u) \right\} N_{\mathcal{H}_6}(v) \\
&= N_{\Xi'_6}(u)N_{\mathcal{H}'_6}(v) + \frac{1}{6}N_{\Xi'_7}(u)N_{\mathcal{H}_6}(v) \\
&= B'_6(u,v) + \frac{1}{6}N_{\Xi'_7}(u)N_{\mathcal{H}_7}(v),
\end{aligned} \tag{3}$$

where $\mathcal{H}_6 = \mathcal{H}'_6$ and $\mathcal{H}_6 = \mathcal{H}_7$; see Fig. 3(a, b). Subsequently, we have the enlarged knot vector $\mathcal{H}'_7 \cup \mathcal{H}'_{32}$ when inserting the knot v_6 into $\mathcal{H}_7$. Eq. (3) becomes

$$\begin{aligned}
B_6(u,v) &= B'_6(u,v) + \frac{1}{6}N_{\Xi'_7}(u)\left\{ \frac{5}{6}N_{\mathcal{H}'_7}(v) + \frac{1}{2}N_{\mathcal{H}'_{32}}(v) \right\} \\
&= B'_6(u,v) + \frac{5}{36}N_{\Xi'_7}(u)N_{\mathcal{H}'_7}(v) + \frac{1}{12}N_{\Xi'_{32}}(u)N_{\mathcal{H}'_{32}}(v) \\
&= B'_6(u,v) + \frac{5}{36}B'_7(u,v) + \frac{1}{12}B'_{32}(u,v),
\end{aligned} \tag{4}$$

where $\Xi'_7 = \Xi'_{32}$; see Fig. 3(b). Similarly, we can obtain $B_2 = B'_2 + \frac{5}{36}B'_7 + \frac{1}{12}B'_{26}$, $B_7 = \frac{25}{36}B'_7 + \frac{5}{12}B'_{26} + \frac{5}{12}B'_{32} + \frac{1}{4}B'_{33}$ and the refinability relationships for all the other basis functions associated with the blue dots. We rewrite them in the matrix form as $\mathbf{B} = \mathbf{M}\mathbf{B}'$, where $\mathbf{M}$ is the refinement matrix with elements obtained from knot insertion. The control points ($\mathbf{P}'$) of $\mathbb{T}'$ are updated as a linear combination of the given T-mesh control points ($\mathbf{P}$), and we have $\mathbf{P}' = \mathbf{M}^T\mathbf{P}$. Generally $\mathbf{P}'$ are not a weighted arithmetic mean of $\mathbf{P}$, that is, the summation of all the weights (or the column summation of $\mathbf{M}$) may not be 1. For instance, $P'_7 = \frac{25}{36}P_7 + \frac{5}{36}P_2 + \frac{5}{36}P_6$ and $\frac{25}{36} + \frac{5}{36} + \frac{5}{36} = \frac{35}{36} \neq 1$; see Fig. 3(b). The blue arrow indicates the contribution of a given point (the start of the arrow) to the updated one (the end of the arrow) with the refinement coefficient shown in the blue box.

Given a T-mesh, if B_i form a partition of unity and all B_i are represented by the basis functions defined on the refined T-mesh, we have $B_i = \sum_j c_{ij} B'_j$ and $\sum_i B_i = \sum_i \sum_j c_{ij} B'_j = \sum_j (\sum_i c_{ij}) B'_j = \sum_j w_j B'_j = 1$, where $w_j = \sum_i c_{ij}$. Note that w_j is the column summation of $\mathbf{M}$ and it may not be 1. Therefore, only a weighted summation of B'_j forms a partition of unity. In general T-spline refinement, partition of unity is not guaranteed for polynomial blending functions. To enable a partition of unity, we can simply rationalize the blending functions. However, these rational basis functions are not polynomial splines anymore and it is somewhat more cumbersome to evaluate their derivatives.

2.2. Analysis-Suitable T-splines

The original development of T-splines proves too general in that linear independence may not be satisfied [2]. Analysis-suitable T-splines [16] were then developed to address this issue, as well as partition of unity. A sufficient condition was provided to ensure these two properties by applying additional topological constraints to the T-mesh, that is, no T-junction extension intersects from different parametric directions. As shown in Fig. 4(a), the T-junction extension consists of the face extension and the edge extension. The face extension shoots a ray across the adjacent face until two intersections are encountered; see the green dashed edges. The edge extension is in the opposite direction of the face extension, and it overlaps with the edge connected to the T-junction; see the red dashed edges. There are three types of T-junction intersections as shown in Fig. 4(b): face-face intersection (the green cross), face-edge intersection (the orange cross), and edge-edge intersection (the red cross). Analysis-suitable T-splines do not allow any such intersections. For instance, after the refinement of two target elements marked in blue in Fig. 3(c), analysis-suitable T-splines require to add at least six edges in order to satisfy the topological constraints; see the red edges in Fig. 3(d). It is obvious that the resulting local refinement in Fig. 3(d) is different from the T-spline refinement result in Fig. 3(b), and their influenced local knot vectors are also different; see the blue dots in Fig. 3(a, c).

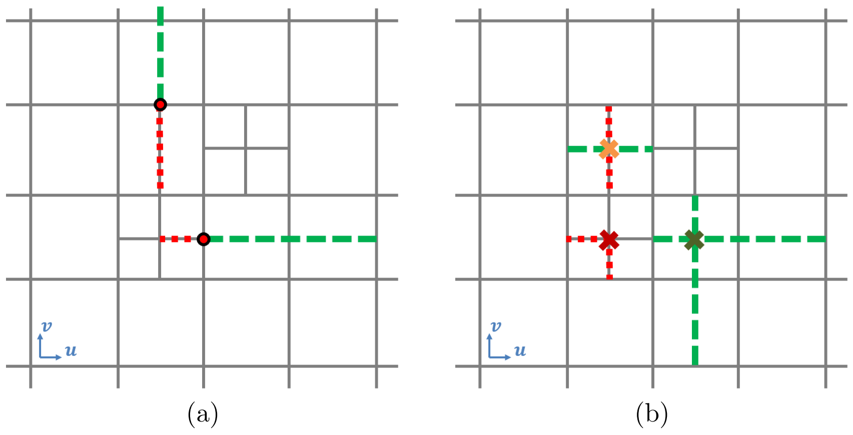


Figure 4: T-junction extensions and intersections. (a) Face extensions (green dashed edges) and edge extensions (red dashed edges) of T-junctions (red dots); and (b) face-face intersection (the green cross), face-edge intersection (the orange cross) and edge-edge intersection (the red cross).

In the following, we will introduce a truncation mechanism into T-splines to release these constraints such that only face-face intersection is not allowed to ensure nested spline spaces. In other words, we allow both face-edge and edge-edge intersections. The resulting basis functions are piece-wise polynomials that form a partition of unity with fewer new edges inserted.

3. Truncated T-splines

In this section, we first briefly review the truncation mechanism in truncated hierarchical B-splines. Then we discuss in detail how we apply it to T-splines, and accordingly how we update the control points.

3.1. *Truncation Mechanism*

The truncation mechanism was first developed for polynomial splines over hierarchical T-meshes [4], and later elaborated for truncated hierarchical B-splines [8]. Recently, it was applied to truncated hierarchical Catmull-Clark subdivision [24, 23]. The truncation mechanism helps build basis functions satisfying partition of unity, preserves the geometry and reduces support overlapping. The truncation of a basis function is to discard active children from its refinability relationship. A child basis function is called *active* if it is selected to form the basis of hierarchical splines.

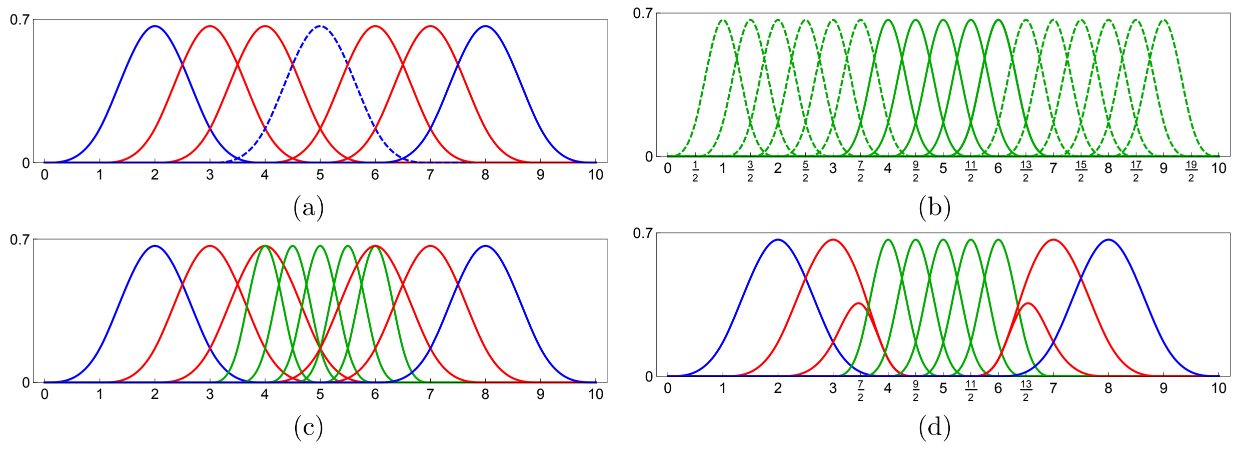


Figure 5: Truncation mechanism in hierarchical B-splines. (a) Level-0 basis functions: a to-be-refined one (the blue dashed curve), to-be-truncated ones (red solid curves), and others (blue solid curves); (b) Level-1 basis functions: active ones (green solid curves) and passive ones (green dashed curves); (c) hierarchical B-spline basis functions; and (d) truncated hierarchical B-spline basis functions (red curves are truncated basis functions).

For example in Fig. 5(a, b), we have two levels (Level 0 and Level 1) of B-spline basis functions. The Level-0 basis functions (blue and red curves) are defined on the knot vector $\Xi^0 = \{0, 1, \ldots, 10\}$, whereas the Level-1 ones (green curves) are defined on the knot vector (Ξ^1) obtained by bisecting Ξ^0. According to refinability, each Level-0 basis function can be represented by a linear combination of its five Level-1 children basis functions. To perform local refinement in hierarchical B-splines [12], certain Level-0 B-spline basis functions are replaced by their children (Level 1). Suppose the blue dashed curve in Fig. 5(a) is such a basis function to be replaced. It has five children defined at Level 1, as shown in green solid curves in Fig. 5(b). By refinement (or replacement), all the solid curves are selected to form the hierarchical basis and they are all active basis functions; see Fig. 5(c). Due to too much overlapping of basis functions from different levels, hierarchical B-spline basis functions do not form a partition of unity. To resolve this issue, the truncation mechanism is introduced to truncate Level-0 basis functions with active children. The red curves in Fig. 5(a) need to be truncated because they have active Level-1 children basis functions. The corresponding truncated basis functions (red curves) are shown in Fig. 5(d), and all the solid curves form the basis of truncated hierarchical B-splines. The essence of the truncation mechanism is to deduct the repeated contribution of active children, ensuring partition of unity as a result. It can also be applied to T-splines to release some topological constraints imposed by analysis-suitable T-splines.

3.2. *Truncated T-spline Basis Functions*

Due to the fundamental structure difference from hierarchical B-splines, applying the truncation mechanism to T-splines is significantly different in determining to-be-discarded children, calculating the corresponding coefficients and updating control points. To facilitate explanation, we first define fully refined and partially refined basis functions in truncated T-splines. During local refinement, the influenced basis functions of a given T-mesh $\mathbb{T}$ are not only the ones whose local knot vectors are changed due to refinement, but also the ones whose certain children are defined on the refined T-mesh $\mathbb{T}'$. An influenced basis function is called *fully refined* if all its children are defined on $\mathbb{T}'$. Otherwise it is called *partially refined*. A fully refined basis function can be updated directly by its children, while a partially refined basis function needs to be truncated. According to this characterization, in the refinement of T-splines and analysis-suitable T-splines, all their influenced basis functions are fully refined. On the other hand, truncated T-splines incorporate a more general case with partially refined basis functions.

Due to its simplicity, $\mathbb{T}'$ is obtained by performing strongly-balanced quadtree subdivision on the target elements of $\mathbb{T}$, together with the additional refinement to ensure no face-face intersection in $\mathbb{T}'$. We call such a refinement scheme the *truncated T-spline quadtree subdivision*. The definition of truncated basis functions is given as follows.

Definition. *Given a T-mesh $\mathbb{T}$ and its refined T-mesh $\mathbb{T}'$ from truncated T-spline quadtree subdivision, $B_i(u,v)$ is a partially refined basis function with certain (but not all) children $B'_j(u,v)$ defined on $\mathbb{T}'$. The truncated basis function of $B_i(u,v)$ is defined as*

$$trunB_i(u,v) = B_i(u,v) - \sum_{j \neq i \wedge B'_j \in chd(B_i)} c_{ij} B'_j(u,v), \tag{5}$$

where c_{ij} are the refinement coefficients obtained from the knot insertion algorithm.

In Eq (5), we call $B_i(u,v)$ the *mother* of $trunB_i(u,v)$, and $B'_j(u,v)$ the *discarded children* of $trunB_i(u,v)$. Note that the discarded children cannot be $B'_i(u,v)$ (when $j = i$). Otherwise, the presence of $B'_i(u,v)$ indicates that $B_i(u,v)$ is fully refined and this contradicts the prerequisite that $B_i(u,v)$ is partially refined. The truncation of $B_i(u,v)$ is performed by deducting the repeated contribution of $B'_j(u,v)$ such that the truncated basis functions form a partition of unity.

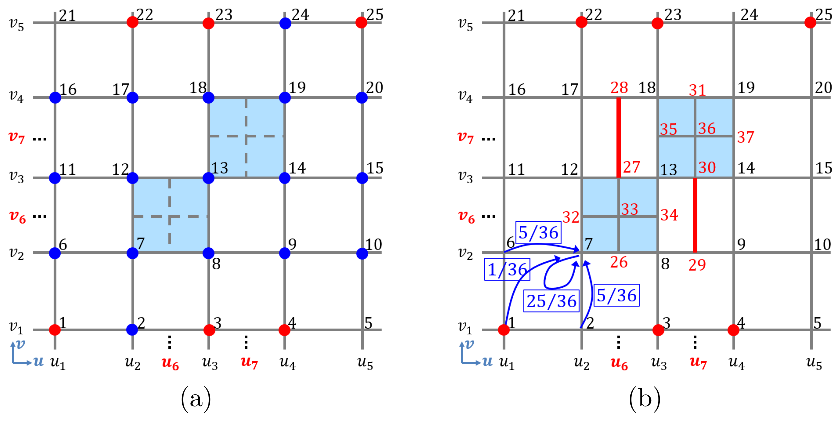


Figure 6: Local refinement of truncated T-splines. (a) An initial local T-mesh ($\mathbb{T}$); and (b) the refined T-mesh ($\mathbb{T}'$) of (a) with two additional edges (red edges). The blue dots represent fully refined basis functions and the red dots represent partially refined basis functions (or truncated basis functions). The blue arrows indicate the contributions of the given points to the updated points with the weights shown in the blue boxes.

Here we use the same example as in Fig. 3(a) to explain the local refinement of truncated T-splines. Fig. 6(a) shows the initial T-mesh $\mathbb{T}$, where the two elements marked in blue are to be refined. However, simply subdividing them would introduce face-face intersections. To eliminate face-face intersections, we bisect two more elements and have a refined T-mesh $\mathbb{T}'$ shown in Fig. 6(b). All the influenced basis functions are marked in blue and red dots whereas the uninfluenced ones are denoted in the index set $\mathcal{I}_0 = \{5, 21\}$; see Fig. 6(a). By performing local knot insertion, we can find that all the basis functions associated with the blue dots in Fig. 6(a) are fully refined (denoted in the index set $\mathcal{I}_F = \{2, 6, \ldots, 20, 24\}$), so they can be updated by their children and no truncation is needed. On the other hand, for any basis function associated with a red dot, not all its children are defined on $\mathbb{T}'$. Therefore, such basis functions (denoted in the index set $\mathcal{I}_P = \{1, 3, 4, 22, 23, 25\}$) are partially refined and need a truncation. Here, let us take $B_1(u,v) = N_{\Xi_1}(u)N_{\mathcal{H}_1}(v)$ as an example. With $\Xi_1 = \Xi_6$ and $\mathcal{H}_1 = \mathcal{H}_2$, we have $B_1(u,v) = N_{\Xi_6}(u)N_{\mathcal{H}_2}(v)$.

After we insert the knots u_6 and v_6 to the local knot vectors Ξ_6 and $\mathcal{H}_2$ as shown in Fig. 6(b), we attain the enlarged knot vectors $\Xi_6' \cup \Xi_7'$ and $\mathcal{H}_2' \cup \mathcal{H}_7'$. Thus $B_1(u,v)$ can be expressed as

$$\begin{aligned} B_1(u,v) &= \left\{ N_{\Xi_6'}(u) + \frac{1}{6} N_{\Xi_7'}(u) \right\} \left\{ N_{\mathcal{H}_2'}(v) + \frac{1}{6} N_{\mathcal{H}_7'}(v) \right\} \\ &= \frac{1}{36} B_7'(u,v) + N_{\Xi_6'}(u) N_{\mathcal{H}_2'}(v) + \frac{1}{6} N_{\Xi_7'}(u) N_{\mathcal{H}_2'}(v) + \frac{1}{6} N_{\Xi_6'}(u) N_{\mathcal{H}_7'}(v). \end{aligned} \tag{6}$$

Note that B_7' is defined on $\mathbb{T}'$ and it is a child of B_1 while the other children (e.g. $N_{\Xi_6'}(u)N_{\mathcal{H}_2'}(v)$) of B_1 are not defined on $\mathbb{T}'$. Therefore, B_1 is partially refined. According to Eq. (5), we need to discard B_7' from Eq. (6) in developing the truncated basis function of B_1, that is,

$$trunB_1(u,v) = B_1(u,v) - \frac{1}{36} B_7'(u,v). \tag{7}$$

On the other hand, according to the definition of truncated basis functions in truncated hierarchical B-splines [8], $trunB_1$ can also be expressed as

$$trunB_1(u,v) = N_{\Xi_6'}(u) N_{\mathcal{H}_2'}(v) + \frac{1}{6} N_{\Xi_7'}(u) N_{\mathcal{H}_2'}(v) + \frac{1}{6} N_{\Xi_6'}(u) N_{\mathcal{H}_7'}(v), \tag{8}$$

where all the basis functions on the right hand side never exist. Eqs. (7, 8) are equivalent to each other in calculating $trunB_1$. Here we employ Eq. (7) because we only need to handle the existing basis functions. Note that the difference in developing truncated basis functions using these two definitions is far more than addition/subtraction of children. This is because the adaptive structure in T-splines is fundamentally different from the hierarchical structure. Similarly, we can verify all the basis functions associated with the red dots in Fig. 6(a) are partially refined, and their truncated basis functions can be developed accordingly. As another representative example, the truncated basis functions of B_3 is

$$trunB_3 = B_3 - \frac{1}{12} B_{26}' - \frac{5}{36} B_8'. \tag{9}$$

After refinement in Fig. 6(b), the basis functions associated with $\mathbb{T}'$ consist of three sets: the children ($\mathcal{I}_C'$) of all the fully refined basis functions, the truncated basis functions ($\mathcal{I}_T'$) developed from the partially refined basis functions, and the uninfluenced basis functions ($\mathcal{I}_0$) of $\mathbb{T}$. We have $\mathcal{I}_T' = \mathcal{I}_P$ and $\mathcal{I}_C' = \mathcal{I}_F \cup \mathcal{I}_N' = \{2, 6, \ldots, 20, 24, 26, \ldots, 37\}$, where $\mathcal{I}_N' = \{26, \ldots, 37\}$ contains the indices of the new points produced by refinement; see Fig. 6(b). The basis functions in $\mathcal{I}_C' \cup \mathcal{I}_0$ are non-truncated and they are defined on the local knot vectors inferred from $\mathbb{T}'$. The truncated basis functions in $\mathcal{I}_T'$ are associated with the local vectors inferred from $\mathbb{T}$ rather than $\mathbb{T}'$ because the mother basis function is used to define a truncated basis function; see Eqs. (7, 9). Beneficial from truncated basis functions, truncated T-spline refinement in Fig. 6(b) allows face-edge and edge-edge intersections, leading to less refinement propagation compared to T-splines (Fig. 3(b)) and analysis-suitable T-splines (Fig. 3(d)).

In truncated T-splines, a valid T-mesh requires no face-face intersection. Otherwise, we cannot use Eq. (5) to develop truncated basis functions. Assume we simply perform the strongly-balanced quadtree subdivision for the target elements (marked in blue) in Fig. 6(a), where the refined T-mesh $\mathbb{T}'$ is represented by the solid and dashed lines. $\mathbb{T}'$ contains two face-face intersections. Let us take a look at $B_{13}(u,v) = N_{\Xi_{13}}(u)N_{\mathcal{H}_{13}}(v)$, which is a partially refined basis function and needs a truncation. By inserting the knots u_6, u_7 and v_6, v_7 into the local knot vectors Ξ_{13} and $\mathcal{H}_{13}$ respectively, we have

$$B_{13}(u,v) = \left\{ \frac{5}{6} N_{\Xi_{13}'}(u) + \frac{1}{2} N_{\Xi_{27}'}(u) + \frac{1}{2} N_{\Xi_{30}'}(u) \right\} \left\{ \frac{5}{6} N_{\mathcal{H}_{13}'}(v) + \frac{1}{2} N_{\mathcal{H}_{34}'}(v) + \frac{1}{2} N_{\mathcal{H}_{35}'}(v) \right\}. \tag{10}$$

We can observe that among all the children of $B_{13}(u,v)$, only $B_{13}'(u,v) = N_{\Xi_{13}'}(u)N_{\mathcal{H}_{13}'}(v)$ is defined on $\mathbb{T}'$. However, according to Eq. (5), we need children other than $B_{13}'(u,v)$ present in $\mathbb{T}'$ to develop $trunB_{13}$. This verifies that a valid T-mesh for truncated T-splines does not allow any face-face intersection. A valid

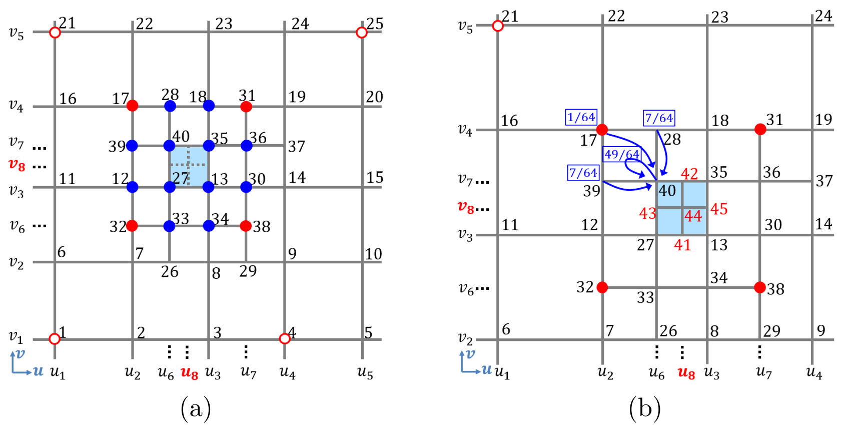


Figure 7: Local refinement of truncated T-splines. (a) An initial local T-mesh $\mathbb{T}$; and (b) the refined T-mesh $\mathbb{T}'$ of (a). The blue dots represent fully refined basis functions and the red dots represent partially refined basis functions or newly developed truncated basis functions. The red circles represent the truncated basis functions associated with $\mathbb{T}$. The blue arrows indicate the contributions of the given points to the updated points with the weights shown in the blue boxes.

T-mesh can be obtained by eliminating face-face intersections, where $B_{13}(u, v)$ becomes a fully refined basis function; see Fig. 6(b).

When multiple levels of refinement are involved, the strongly-balanced T-mesh is employed to simplify the implementation, where the level difference of any two neighboring elements is at most one. The level of an element is determined by the lowest level of its knot intervals. For a to-be-refined element, either two or four of its knot intervals are bisected; see Fig. 6(b). All the knot intervals associated with the initial T-mesh are at Level 0, and bisecting one Level-ℓ ($\ell \in \mathbb{N}$) knot interval results in two Level-($\ell + 1$) knot intervals. For instance, we can also obtain a two-level T-mesh $\mathbb{T}$ in Fig. 7(a) by refining Fig. 6(a). Four truncated basis functions are associated with $\mathbb{T}$, as marked in red circles. Taking $trunB_{21}$ as an example, we have

$$trunB_{21} = B_{21} - \frac{1}{36}B_{17}. \tag{11}$$

We further refine $\mathbb{T}$ to obtain a strongly-balanced three-level T-mesh $\mathbb{T}'$ in Fig. 7(b) by subdividing the blue element. Among the influenced basis functions of the further refinement, we mark the fully refined and partially refined basis functions in blue and red dots, respectively; see Fig. 7(a). We need to develop new truncated basis functions for the red dots following Eq. (5). The truncated basis functions associated with the red circles stay the same during the further refinement. This can be verified using $trunB_{21}$. Note that B'_{40} is a child of both B_{21} and B_{17}. Following the truncation mechanism, we need to further discard B'_{40} from $trunB_{21}$. As a child of B_{17}, the contribution of B'_{40} has been counted in $\frac{1}{36}B_{17}$ and discarded already in developing $trunB_{21}$; see Eq. (11). Therefore, we do not need to discard it again, and thus $trunB_{21}$ is not an influenced basis function for the further refinement. In other words, in the strongly-balanced T-mesh, the influenced basis functions defined on $\mathbb{T}$ are always non-truncated; see Figs. 6(a) and 7(a).

Discussion 3.1. The T-mesh obtained from other than strongly-balanced quadtree subdivision is also allowed in truncated T-splines. Generally, it may involve further truncating a truncated basis function, updating certain discarded children for a truncated basis function, and updating a truncated basis function to a non-truncated one. Given a truncated basis function $trunB_i$ obtained by Eq. (5), all its possible changes can be summarized in

$$trunB'_i = B_i - \sum_{j \neq i \wedge B'_j \in chd(B_i)} c_{ij}B'_j - \sum_{k \neq i \wedge B''_k \in chd(B_i)} t_{ik}B''_k, \tag{12}$$

with

$$t_{ik} = c_{ik} - \sum_{B''_k \in chd(B'_j)} c_{ij}c_{jk}, \tag{13}$$

where B''_k are non-truncated basis functions defined on the further refined T-mesh ($\mathbb{T}''$), and c_{ik}, c_{ij} and c_{jk} are refinement coefficients. Note that the contribution of $\sum_{B''_k \in chd(B'_j)} c_{ij} c_{jk} B''_k$ has been counted in developing $trunB_i$. Therefore, we do not need to count it again in computing t_{ik}. In this manner, truncated T-splines can be recursively constructed.

3.3. Control Point Calculation

When refining a T-mesh $\mathbb{T}$ to $\mathbb{T}'$, we need to calculate the control points associated with the children ($\mathcal{I}'_C$) of the fully refined basis functions. For instance in Fig. 6, we have $\mathcal{I}'_C = \{2, 6, \ldots, 20, 24, 26, \ldots, 37\}$. Recall that in T-spline refinement, only points with influenced local knot vectors are used in calculating control points. However, in truncated T-spline refinement, every control point (P_i, $i \in \mathcal{I}_F \cup \mathcal{I}_P$) associated with an influenced basis function is used. Moreover, in the strongly-balanced T-mesh, the refinement coefficients (c_{ij}) are used as the weights of P_i to calculate the control point P'_j of $\mathbb{T}'$, and we have

$$P'_j = \sum_{B'_j \in chd(B_i)} c_{ij} P_i, \quad \forall j \in \mathcal{I}'_C. \tag{14}$$

On the other hand, for points (P'_j, $j \in \mathcal{I}'_T$) associated with truncated basis functions, their coordinates remain the same as P_j in $\mathbb{T}$.

In Fig. 6, we take P'_7 as an example to explain the updating process. Based on the previous discussion, P_1, P_2, P_6 and P_7 are used to update P'_7 because B'_7 is a child of B_1, B_2, B_6 and B_7; see Fig. 6(a, b). By performing knot insertion, we have $P'_7 = \frac{1}{36}P_1 + \frac{5}{36}P_2 + \frac{5}{36}P_6 + \frac{25}{36}P_7$, which is a weighted arithmetic mean of the points in $\mathbb{T}$. The difference from Fig. 3(b) is the presence of $\frac{1}{36}P_1$. In Fig. 3(b), P_1 is not used to update P'_7 because its knot vectors are not influenced by refinement. On the other hand, $\frac{1}{36}P_1$ will also be used to update P'_7 if we globally refine the T-mesh in a B-spline manner. Therefore when the T-mesh is strongly-balanced, truncated T-splines update control points in an equivalent manner as B-spline refinement, which can be further explained using Fig. 7. Let us consider how to update P'_{40}. P_{40}, P_{28}, P_{39}, and P_{17} are used to update P'_{40} because B'_{40} is a child of all their corresponding basis functions. Recall that although B'_{40} is also a child of B_{21}, B'_{40} actually has no influence on $trunB_{21}$ because its contribution has been counted earlier, so P_{21} (associated with $trunB_{21}$ rather than B_{21}) is not used to update P'_{40}. Then we have $P'_{40} = \frac{49}{64}P_{40} + \frac{7}{64}P_{28} + \frac{7}{64}P_{39} + \frac{1}{64}P_{17}$, which is also a weighted arithmetic mean of the points in $\mathbb{T}$.

Discussion 3.2. In the general refinement of truncated T-splines as introduced in Discussion 3.1, the control point associated with a truncated basis function may also be used to update $\mathbb{T}''$. We call B''_k a child of $trunB_i$ if $t_{ik} \neq 0$ (see Eq. (13)), denoted as $B''_k \in chd(trunB_i)$. Then a control point P''_k of $\mathbb{T}''$ is calculated as

$$P''_k = \sum_{B''_k \in chd(B_i)} c_{ik} P_i + \sum_{B''_k \in chd(B'_j)} c_{jk} P'_j + \sum_{B''_k \in chd(trunB_i)} t_{ik} P_i. \tag{15}$$

Note that we use t_{ik} instead of c_{ik} if P_i is associated with a truncated basis function.

Discussion 3.3. By virtue of the truncation mechanism, truncated T-splines can release the topological constraints in analysis-suitable T-splines [16]. Recall that analysis-suitable T-splines do not allow any type of intersections of T-junction extensions, while in truncated T-splines, we allow face-edge and edge-edge intersections but still do not allow face-face intersection. Compared to T-splines (Fig. 3(b)) and analysis-suitable T-splines (Fig. 3(d)), truncated T-splines introduce less refinement propagation and insert only two edges (red edges) to achieve a topologically valid T-mesh; see Fig. 6(b). Given a tensor-product mesh in Fig. 8(a), we use truncated T-splines to recursively subdivide the elements along one diagonal direction. Then we obtain a locally refined control mesh and the truncated T-spline surface shown in Fig. 8(b, c), respectively. The blue dots are associated with non-truncated T-spline basis functions, whereas the red dots are associated with truncated basis functions. The color in Fig. 8(c) represents the summation of all basis functions, which equals 1 everywhere, showing that they form a partition of unity. We can also observe face-edge intersections existing in Fig. 8(b, c), which are not allowed in analysis-suitable T-splines.

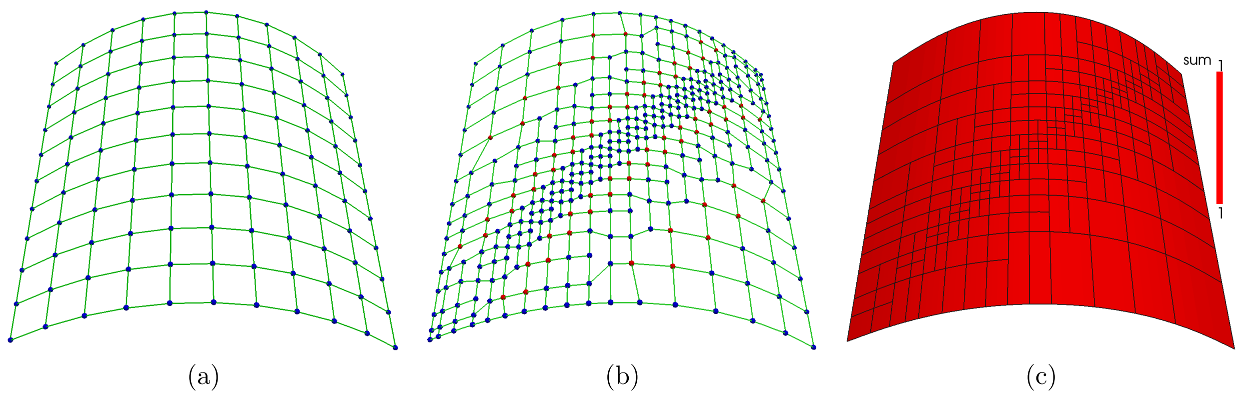


Figure 8: (a) The input control mesh; (b) the T-mesh after refinement with the red and blue points indicating truncated and non-truncated basis functions, respectively; and (c) the truncated T-spline surface with isoparametric lines with the color bar showing the summation of basis functions.

Refinement along the diagonal direction using standard T-splines results in an extensive propagation [6]. Truncated T-splines, on the other hand, release the refinement constraint for less propagation, providing more flexibility in local refinement.

Truncated T-splines have several nice properties such as geometry preservation, partition of unity, and (global) linear independence; see the propositions in Appendix A for details.

4. Extraordinary Points

In this section we discuss truncated T-splines on general 2-manifold domains with extraordinary points. We first define necessary terminologies to facilitate our explanation. Then we introduce how to infer local knot vectors near extraordinary points and obtain a gap-free surface around them. In the end, we use degree elevation and optimization to improve the surrounding surface continuity to G^1.

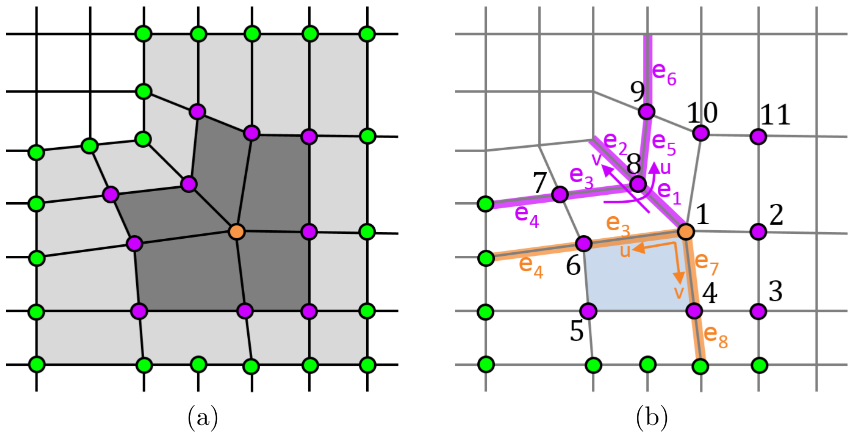


Figure 9: T-mesh with extraordinary points. (a) The 1st-ring neighborhood (dark gray) and the 2nd-ring neighborhood (light gray) of an extraordinary point (the orange dot) with purple dots lying on the 1st-ring boundary and green dots lying on the 2nd-ring boundary; and (b) inferring local knot intervals around an extraordinary point.

Given an extraordinary point P_i, the 1st*-ring neighborhood* of P_i consists of the quadrilaterals touching P_i, and the 2nd*-ring neighborhood* of P_i is formed by quadrilaterals touching the one-ring neighborhood excluding the 1st-ring neighborhood. The n^{th}-ring neighborhood is formed analogously. The n^{th}*-ring boundary* of P_i consists of the edges shared by the n^{th}-ring and $(n+1)^{th}$-ring neighborhoods. In Fig. 9(a), given the extraordinary point marked as an orange dot, its 1st-ring neighborhood is marked in dark gray, and the 2nd-ring neighborhood is marked in light gray. The 1st-ring boundary is formed by edges connecting the purple dots, and the 2nd-ring boundary consists of the edges connecting the green dots. A quadrilateral (or element) is called *irregular* if any of its four corners is an extraordinary point. For an irregular element, as shown in blue in Fig. 9(b), we label the extraordinary node with index 1, and label the nodes on the 1st-ring boundary clockwise from 2 to $2N+1$, where N is the number of elements touching the extraordinary node, or the *valence number*, and e_i $(i = 1, \dots, 8)$ refer to knot intervals.

We briefly explain how the local knot vectors are inferred around extraordinary nodes by shooting rays and repeating knot intervals [13]. In this method, whenever a ray encounters an extraordinary point before it finds two knot intervals, we simply duplicate the previous interval. We take P_8 in Fig. 9(b) as an example. We first shoot a ray in the local $-v$ direction and we find the knot interval e_1. However, the ray encounters the extraordinary point P_1 before it finds the next knot interval. According to [13], this knot interval is duplicated. Similarly in the $+v$ direction, we have both knot intervals as e_2. No duplication is needed in the u direction. In summary, the knot interval vectors of P_8 are $\{e_4, e_3, e_5, e_6\}$ along the u direction and $\{e_1, e_1, e_2, e_2\}$ along the v direction. As another example, we can obtain the two knot interval vectors of P_1 with respect to the blue element, $\{e_3, e_3, e_3, e_4\}$ along the u direction and $\{e_7, e_7, e_7, e_8\}$ along the v direction. Then their corresponding knot vectors can be easily computed. Note that in [13], no T-junctions or other extraordinary points are allowed within the four-ring neighborhood of any given extraordinary point. For uniform knot intervals, this method can be used with two released constraints: 1) no T-junctions or other extraordinary points are allowed in the 1st-ring neighborhood of any given extraordinary point; 2) all the edges touching the 1st-ring boundary have the same non-zero knot interval. These constraints are released compared to [13] such that T-junctions or other extraordinary points are allowed on the 2nd-ring boundary; see Fig. 9(a).

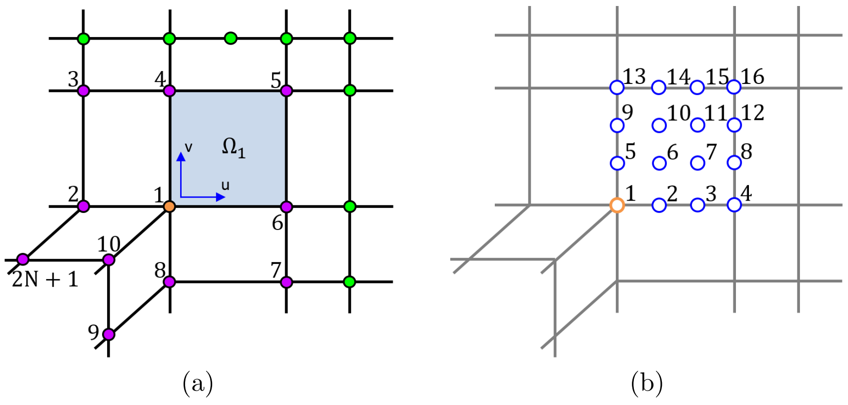


Figure 10: An irregular element Ω_1 (a) and its bicubic Bézier patch (b). The orange and blue circles in (b) schematically represent the Bézier control points.

With the extracted local knot vectors, we need to further modify their defined basis functions to obtain a gap-free or G^1-continuous surface. We use Bézier patches to achieve this goal. To obtain Bézier patches, we perform Bézier extraction for each basis function by knot insertion. Bézier extraction for a non-truncated basis function is the same as that in [15], whereas additional treatment is required for a truncated basis function. According to Eq. (5), Bézier extraction is performed for both the mother $B_i(u,v)$ and the discarded children $B'_j(u,v)$. Then $B_i(u,v)$ and $B'_j(u,v)$ can be represented by 16 bicubic Bernstein polynomials $(B_k^{b,3})$, and we have

$$B_i(u,v) = \sum_{k=1}^{16} b_{ik} B_k^{b,3}(u,v) \tag{16}$$

and

$$B'_j(u,v) = \sum_{k=1}^{16} b_{jk} B_k^{b,3}(u,v), \tag{17}$$

where the Bernstein polynomials $B_k^{b,3}(u,v)$ are labeled as in Fig. 10(b), and b_{ik}, b_{jk} are the Bézier coefficients obtained by knot insertion. Plugging Eqs. (16, 17) into Eq. (5), we have

$$\begin{aligned} trunB_i(u,v) &= \sum_{k=1}^{16} b_{ik} B_k^{b,3}(u,v) - \sum_{j \neq i \wedge B'_j \in chd(B_i)} c_{ij} \sum_{k=1}^{16} b_{jk} B_k^{b,3}(u,v) \\ &= \sum_{k=1}^{16} b_{ik}^t B_k^{b,3}(u,v) \end{aligned} \tag{18}$$

with

$$b_{ik}^t = b_{ik} - \sum_{j \neq i \wedge B'_j \in chd(B_i)} c_{ij} b_{jk}, \tag{19}$$

where b_{ik}^t are the Bézier coefficients for truncated basis functions. For convenience, we use b_{ik}^t as the Bézier coefficients for both $B_i(u,v)$ and $trunB_i(u,v)$, where $b_{ik}^t = b_{ik}$ if $B_i(u,v)$ is non-truncated.

As we intend to improve continuity by changing basis functions, we can adjust the Bézier coefficients after Bézier extraction. Among the basis functions with support on the irregular element Ω_1, we only need to modify the basis functions $B_1 \sim B_{2N+1}$ (Fig. 10(a)). On Ω_1, after expressing $B_1 \sim B_{2N+1}$ in terms of $B_k^{b,3}(u,v)$, certain basis functions are merely a weighted function of $B_1^{b,3}(u,v)$. Such basis functions are denoted in an index set $\mathcal{I}_1$ and the others are denoted in $\mathcal{I}_2$. We have $\mathcal{I}_1 = \{9, \ldots, 2N+1\}$ and $\mathcal{I}_2 = \{1, 2, \ldots, 8\}$ when $N \geq 5$, or $\mathcal{I}_1 = \varnothing$ and $\mathcal{I}_2 = \{1, 2, \ldots, 7\}$ when $N = 3$. For basis functions in $\mathcal{I}_2$, we modify them as

$$B_i = c_{i1}^l B_1^{b,3} + \sum_{j=2}^{16} b_{ij}^t B_j^{b,3}, \quad \forall i \in \mathcal{I}_2, \tag{20}$$

and for the basis functions in $\mathcal{I}_1$, we change the weights as

$$B_i = c_{i1}^l B_1^{b,3}, \quad \forall i \in \mathcal{I}_1, \tag{21}$$

where c_{i1}^l are the coefficients used in Catmull-Clark subdivision to calculate the limit surface point corresponding to the extraordinary point P_1 [20]. This can also be understood from the viewpoint of calculating the Bézier control point Q_1; see the orange circle in Fig. 10(b). Q_1 is calculated from a linear combination of P_1 and its 1st-ring boundary points, $Q_1 = \sum_{i=1}^{2N+1} c_{i1}^l P_i$, which is exactly how the limit surface point (corresponding to P_1) is calculated in Catmull-Clark subdivision. Moreover, Q_1 remains the same no matter how many times the Catmull-Clark subdivision is performed. Since the (truncated) T-spline surface around P_1 is represented by the Bézier surface, and the Bézier surface interpolates its corner control point Q_1, the T-spline surface also interpolates Q_1. Besides, since we employ Catmull-Clark subdivision to locally refine irregular elements, the surface obtained from the refined T-mesh remains the same at Q_1.

With the Bézier patches, we elevate the degree and optimize the Bézier coefficients to obtain G^1-continuity around extraordinary nodes. In Eqs. (20, 21), we have built the T-spline-to-Bézier transformation matrix, written in matrix form as $\mathbf{B} = \mathbf{M}^{T,3}\mathbf{B}^{b,3}$, where $\mathbf{B}$ is the vector of (truncated) T-spline basis functions, $\mathbf{B}^{b,3}$ is the vector of bicubic Bernstein polynomials, and $\mathbf{M}^{T,3}$ is the transformation matrix. Next we elevate the degree of Bernstein polynomials to biquartic, in order to have more degrees of freedom to solve the optimization problem. Thus we have $\mathbf{B}^{b,3} = \mathbf{M}^{3,4}\mathbf{B}^{b,4}$, where $\mathbf{M}^{3,4}$ is the transformation matrix from bicubic Bernstein polynomials to biquartic ones. The (truncated) T-spline basis functions are then expressed by biquartic Bernstein polynomials, $\mathbf{B} = \mathbf{M}^{T,4}\mathbf{B}^{b4}$, where $\mathbf{M}^{T,4} = \mathbf{M}^{T,3}\mathbf{M}^{3,4}$. In [17], G^1 constraints are applied across the edges touching the extraordinary point. Besides these geometric constraints, we add one more constraint that all c_{i1}^l ($i \in \mathcal{I}_1 \cup \mathcal{I}_2$) remain the same. It ensures the surface still interpolates

Q_1 after optimization. Under these geometric constraints, an optimization problem is solved by minimizing the change in the edge length of the Bézier control mesh. All the involved equations are derived in terms of the Bézier coefficients (the elements in $\mathbf{M}^{T,4}$). Then we obtain an optimized transformation matrix $\mathbf{M}^{op}$ such that $\mathbf{B} = \mathbf{M}^{op}\mathbf{B}^{b,4}$. To this end, the G^1 continuity is built into the basis functions.

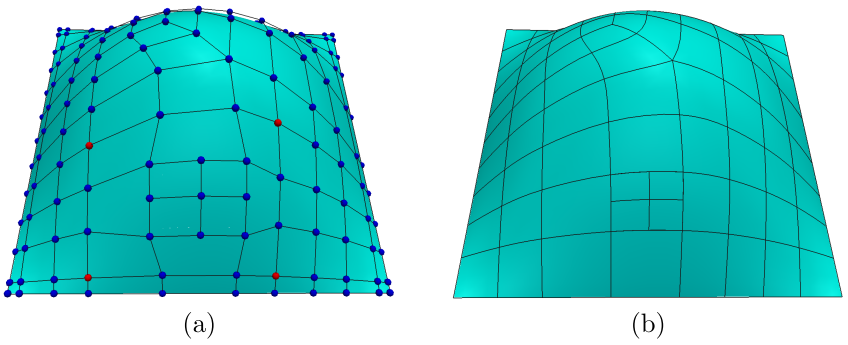


Figure 11: G^1 continuous surface with a T-junction and another extraordinary point present on the 2^{nd}-ring boundary. (a) T-mesh with extraordinary points; and (b) G^1 continuous T-spline surface.

Fig. 11(a) shows the T-mesh where the red dots are associated with truncated basis functions and the blue dots are associated with non-truncated ones. The resulting smooth G^1 surface is shown in Fig. 11(b). The main difference between our G^1 continuity construction and the previous work [17, 13] lies in two aspects: 1) we update Q_1 (Fig. 10(b)) as the Catmull-Clark limit surface point; and 2) we allow T-junctions and other extraordinary points on the 2^{nd}-ring boundary whereas they are only allowed beyond the 3^{rd}-ring neighborhood in [17, 13]. The reason for this improvement is that our T-spline-to-Bézier transformation matrix $\mathbf{M}^{T,3}$ is built by properly inferring local knot vectors around extraordinary points. This method works as long as the knot intervals are uniform around extraordinary points.

5. Examples and Discussion

We first use two patch tests to study the analysis-suitability of truncated T-spline basis functions by solving a 2D linear elasticity problem. A unit square is subject to uniform tension in x-direction while translations are restrained in the y- and x-directions for the bottom and left boundaries, respectively; see Fig. 12(a, d). Young's modulus and Poisson's ratio are given as $E = 1$ and $\nu = 0.3$. For numerical integration, the $(p + 1)$-point Gauss-Legendre quadrature rule is employed, where p is the degree of basis functions. We study two input T-meshes: a T-mesh defined on a regular rectangular domain and a T-mesh with two extraordinary points. The corresponding truncated T-spline surfaces are shown in Fig. 12(a, d). In Fig. 12(d), there exist a T-junction and a valence-3 extraordinary node on the 2^{nd}-ring boundary of the valence-5 extraordinary node. Using truncated T-splines, both meshes pass the patch tests with machine precision. In Fig. 12(b, c, e, f), we show the distribution of the normal strains on the Bézier elements.

We then study a 2D linear elasticity problem as our first benchmark problem: an infinite plate with a circular hole under a far-field constant in-plane tension. As shown in Fig. 13(a), we only need to study a finite quarter plate due to symmetry. The exact solution can be found in [9]. Two input control meshes are used to solve this problem: a regular mesh and an irregular mesh with one valence-5 extraordinary point, as shown in Fig. 13(b, f), respectively. In these two meshes, the circular arc is modeled as a cubic NURBS curve by degree elevating the corresponding quadratic curve. In the regular mesh, the knot vector of the underlying quadratic NURBS curve is $\{0, 0, 0, \frac{1}{2}, \frac{1}{2}, 1, 1, 1\}$. On the other hand, in the irregular mesh the knot vector of the cubic NURBS curve is $\{0, 0, 0, 0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1, 1, 1, 1\}$. The resulting surface is G^1-continuous around the extraordinary point. We perform both uniform and adaptive refinement. In the adaptive case, the elements with larger error than a given threshold are refined. We evaluate the L^2 error of the stress (σ_{xx}) on each element and on the entire domain. The final meshes are shown in Fig. 13(c, g), and the corresponding element-wise error distribution are shown in Fig. 13(d, h). We can observe that

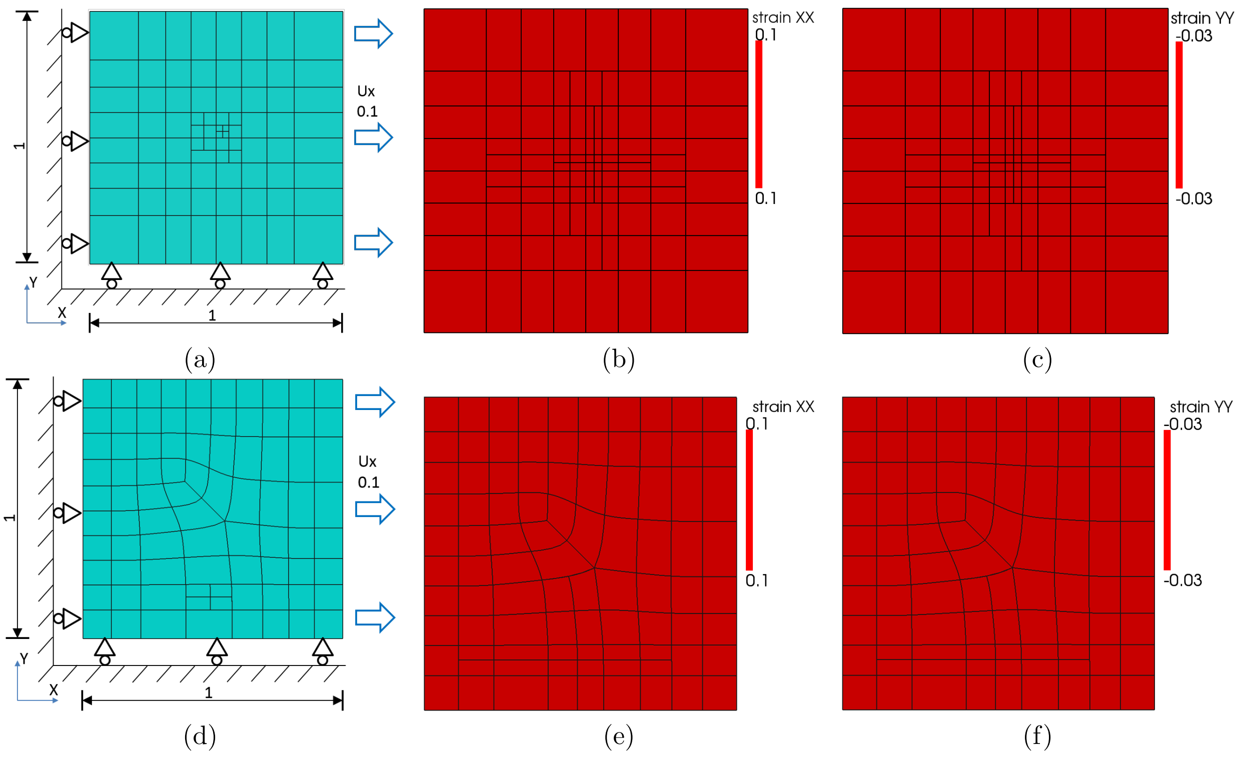


Figure 12: Patch tests by solving a 2D linear elasticity problem. (a, d) Input geometries and Dirichlet boundary conditions; (b, e) normal strain $(X-X)$ distribution on Bézier elements; and (c, f) normal strain $(Y-Y)$ distribution on Bézier elements.

with the regular mesh, the refinement is localized around the hole to decrease the overall error, whereas the refinement is required around the extraordinary point in the irregular mesh. The convergence curves are also plotted with respect to degrees of freedom (DOF) in Fig. 13(e), under both uniform and adaptive refinement. In uniform refinement, the optimal convergence rate of $\frac{3}{2}$ (w.r.t. DOF instead of the element size) is achieved for the regular mesh, whereas the convergence rate using the irregular mesh is only 1. This is because the surrounding splines of an extraordinary point fail to reproduce polynomials exactly. Recovering the optimal convergence rate with presence of extraordinary points is still an open problem. Moreover, adaptive refinement possesses higher efficiency, that is, it needs fewer DOF to achieve the same accuracy as the uniform refinement.

Next we study another widely used benchmark problem: solving the Laplace equation $\Delta u = 0$ on the L-shaped domain $[-1,1]^2\backslash[0,1]^2$; see Fig. 14(a), where Dirichlet boundary conditions (Γ_D) are strongly imposed. The exact solution to this problem is given in polar coordinates (r,θ) as

$$u(r,\theta) = r^{2/3} sin(2\theta/3 - \pi/3), \quad r > 0 \text{ and } \pi/2 \leq \theta \leq 2\pi. \tag{22}$$

Two input control meshes are used: a regular mesh and an irregular mesh with four extraordinary points, shown in Fig. 14(b, f), respectively. We then perform adaptive isogeometric analysis using truncated T-splines. After each step of simulation, the H^1-seminorm error is assessed for each element as well as the entire domain. We locally subdivide the elements with larger error than the given threshold. Since the derivative of the solution field has a singularity at the concave sharp corner, the refinement guided by the H^1-seminorm error is localized at this corner. When the overall H^1-seminorm error reaches a given threshold, the simulation terminates. Fig. 14(c, g) shows truncated T-spline surfaces whereas Fig. 14(d, h) shows the element-wise H^1-seminorm error on Bézier elements. In Fig. 14(e), we plot the convergence curves with respect to DOF. We observe that the two curves have similar convergence rates (around 2), but the irregular mesh needs more DOF to reach the same accuracy. This is because during refinement, if any element is

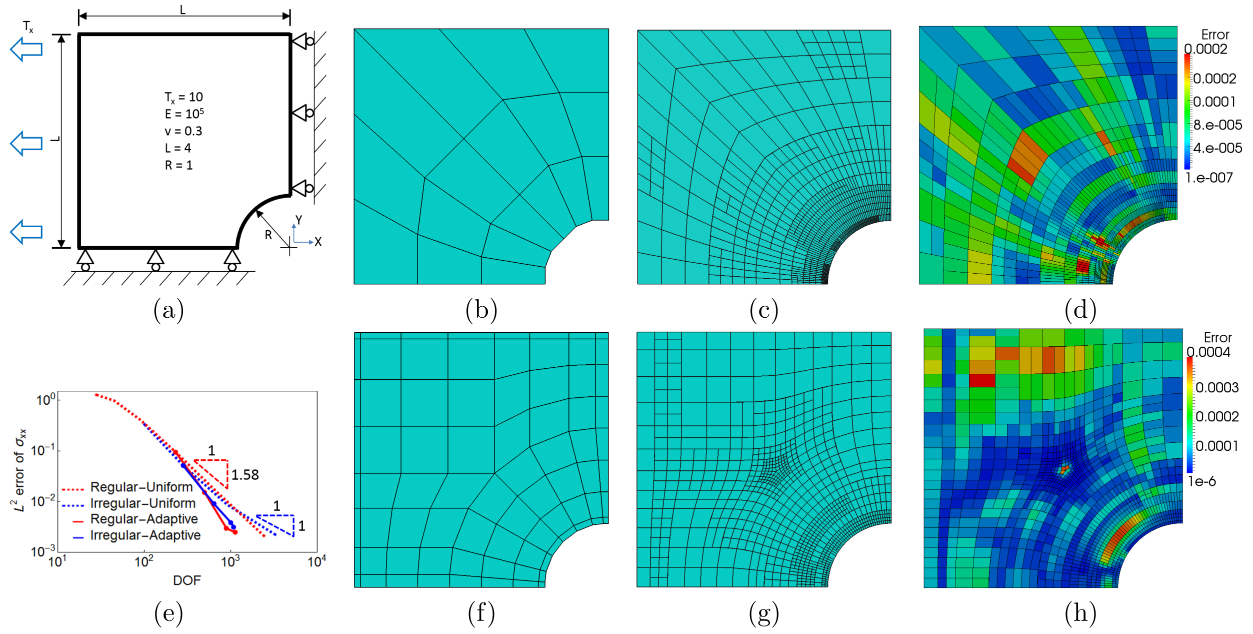


Figure 13: Solving the 2D linear elasticity problem on an infinite plate with a hole under constant in-plane tension. (a) Problem setting; (b, f) input control meshes; (c, g) geometries with parametric lines; (d, h) element-wise L^2 error of stress (σ_{xx}) on Bézier elements; and (e) Convergence curves with respect to DOF.

refined within the two-ring neighborhood of an extraordinary node, all the two-ring neighboring elements of the extraordinary node need to be refined to satisfy the topological constraints on general manifold domains. Our convergence behavior is similar to that using locally refinable B-splines [11].

To show the potential application of truncated T-splines on complex geometries, we also solve the Laplace equation on three locally refined complex models: a fertility model, a bunny model and a hand model. The input is either a T-mesh (Fig. 15(a)) or irregular quadrilateral meshes (Figs. 16(a) and 17(a)). For the fertility model, the input T-mesh is obtained by locally refining an irregular quadrilateral mesh using truncated T-splines. The red dots in Fig. 15(a) indicate points associated with truncated basis functions, whereas blue dots are associated with non-truncated ones. The corresponding truncated T-spline surface is shown in Fig. 15(b) with isoparametric lines. In the input quadrilateral meshes of the bunny and hand models, certain quadrilaterals have more than one extraordinary point; see the red dots in Figs. 16(a) and 17(a). We call such quadrilaterals *invalid* since they violate the constraint that no other extraordinary points are allowed in the 1st-ring neighborhood of any given extraordinary point. Invalid quadrilaterals need to be subdivided before truncated T-splines can be applied. Furthermore, given an extraordinary node, if any element of its two-ring neighborhood (the 1st- and 2nd-ring neighborhoods) is refined, all its two-ring neighboring elements are also refined. Figs. 16(b) and 17(b) show the truncated T-spline surfaces after subdividing invalid elements. Under the Dirichlet boundary conditions as prescribed in Figs. 15(a), 16(a) and 17(a), we solve the Laplace equation on these three surfaces using truncated T-splines. The corresponding solution fields are shown on Bézier elements in Figs. 15(c), 16(c) and 17(c).

6. Conclusion

In this paper, we have presented a new type of T-spline, namely a truncated T-spline, by introducing the truncation mechanism into T-splines to release the topological constraints of T-spline refinement. Truncated T-splines have some nice properties, such as polynomial basis functions, partition of unity, geometry preservation, nested spline spaces, less refinement propagation, and linear independence. Using degree elevation and optimization, we also studied truncated T-splines on general manifold domains with

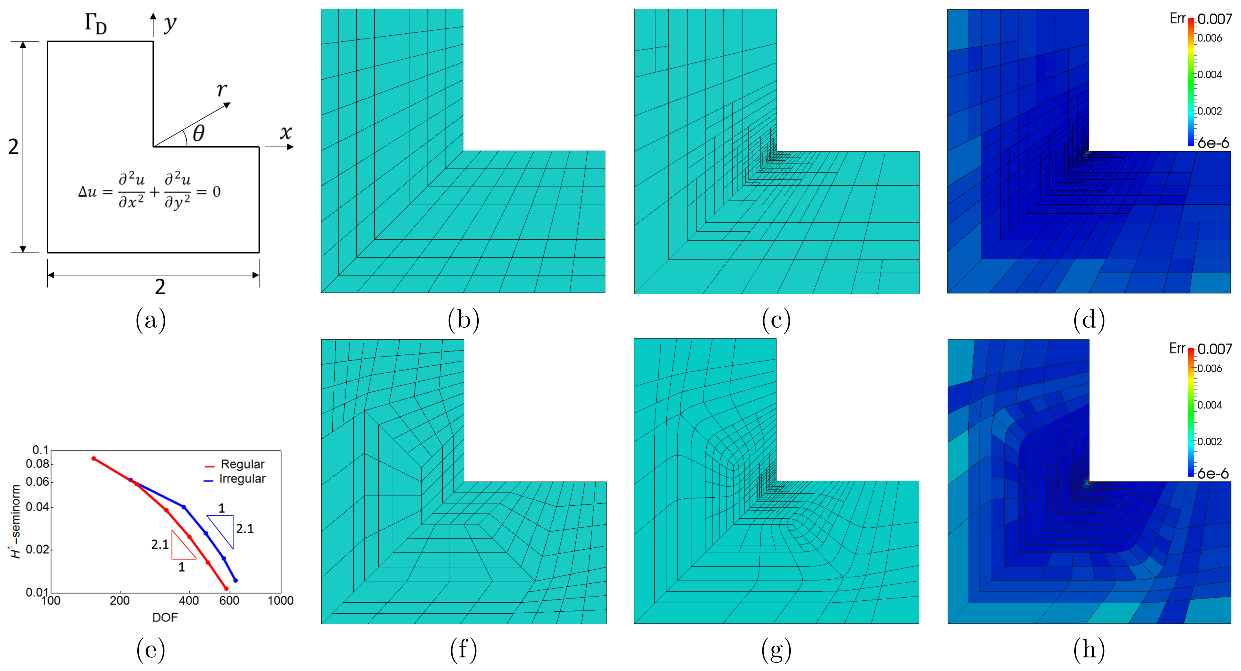


Figure 14: Solving the Laplace equation on the L-shaped domain. (a) Problem setting; (b, f) input control meshes; (c, g) geometries with parametric lines; (d, h) element-wise error (H^1-seminorm) on Bézier elements; and (e) H^1-seminorm convergence curves with respect to DOF.

extraordinary points. T-junctions or other extraordinary points are allowed on the 2^{nd}-ring boundary of a given extraordinary point. In the end, patch tests and a benchmark problem are studied to demonstrate the analysis-suitability and refinement locality of truncated T-splines. We also applied truncated T-splines to several complex models, showing the potential wide application of this method.

In the future, highly localized refinement of invalid elements could be promising in T-splines. In Appendix A, we prove geometry preservation, partition of unity and (global) linear independence of truncated T-splines on regular domains with no extraordinary points. Further fundamental study is also required to address linear independence and nested spline spaces on general manifold domains with extraordinary points. Volumetric truncated T-spline modeling is another interesting topic to investigate. There are still many challenging problems in volumetric spline modeling, such as handling three dimensional extraordinary points and conformal volumetric parameterization from NURBS/T-spline surface representation. Recovering optimal convergence rate also deserves further investigation for T-meshes with extraordinary points.

Acknowledgements

We would like to thank T. Liao for providing the input quadrilateral meshes of the fertility, bunny and hand models. X. Wei, Y. Zhang and L. Liu were supported in part by the PECASE Award N00014-14-1-0234 and NSF CAREER Award OCI-1149591. T. J. R. Hughes was supported in part by grants ONR (N00014-08-1-0992) and SINTEF (UTA10-000374).

Appendix A: Properties of Truncated T-Splines

In this appendix, we prove several important properties of truncated T-splines on regular domains (without extraordinary points): geometry preservation, partition of unity, and (global) linear independence. Here, we restrict the proofs to the truncated T-spline quadtree subdivision, where we perform strongly-balanced

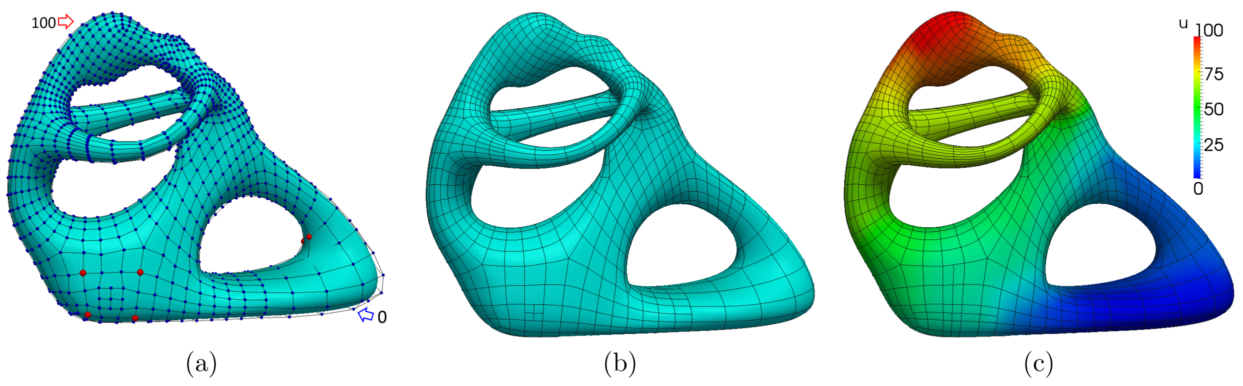


Figure 15: Solving the Laplace equation on a fertility model. (a) Input mesh and Dirichlet boundary condition; (b) truncated T-spline surface; and (c) solution field on Bézier elements.

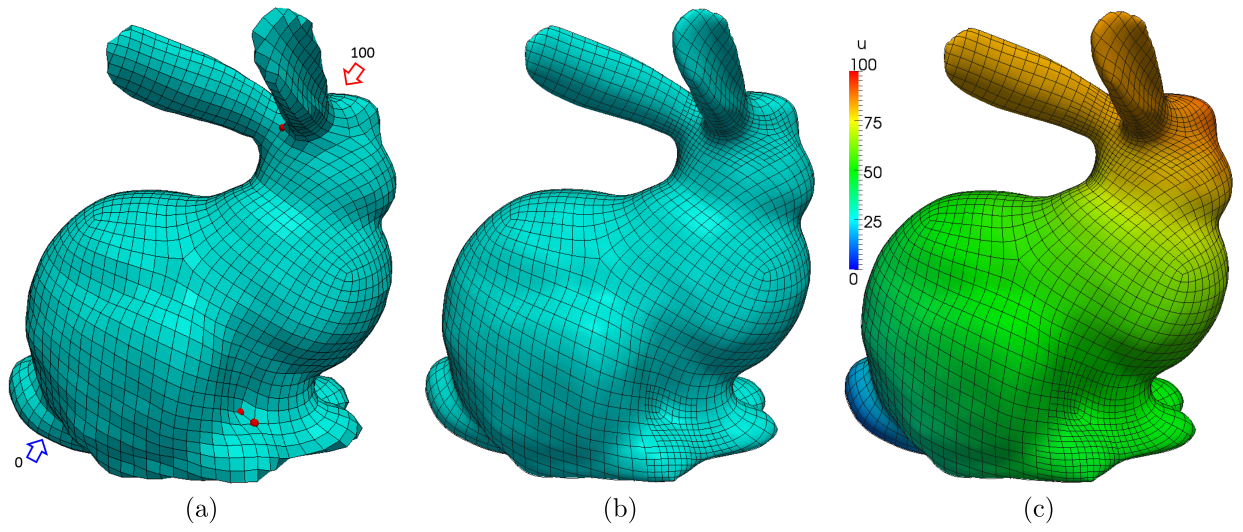


Figure 16: Solving the Laplace equation on a bunny model. (a) Input mesh and Dirichlet boundary condition; (b) truncated T-spline surface; and (c) solution field on Bézier elements.

quadtree subdivision for target elements and also bisect necessary elements to ensure no face-face intersection.

Proposition 1. *In truncated T-splines, the truncated T-spline quadtree subdivision does not change the geometry.*

Proof. We only need to prove that the surface represented by the influenced basis functions does not change during the truncated T-spline quadtree subdivision. Among the influenced basis functions defined on the given T-mesh $\mathbb{T}$, let $\mathcal{I}_F$ denote the index set of fully refined basis functions, and $\mathcal{I}_P$ denote the index set of partially refined ones. Under the truncated T-spline quadtree subdivision, recall that all these influenced basis functions are non-truncated. Prior to refinement, we have the surface represented by the influenced basis functions as

$$S = \sum_{i \in \mathcal{I}_F \cup \mathcal{I}_P} P_i B_i. \tag{23}$$

After refinement, the fully refined basis functions ($\mathcal{I}_F$) are updated by their children. These children are non-truncated basis functions defined on the refined T-mesh $\mathbb{T}'$, whose index set is denoted as $\mathcal{I}'_C$. Moreover,

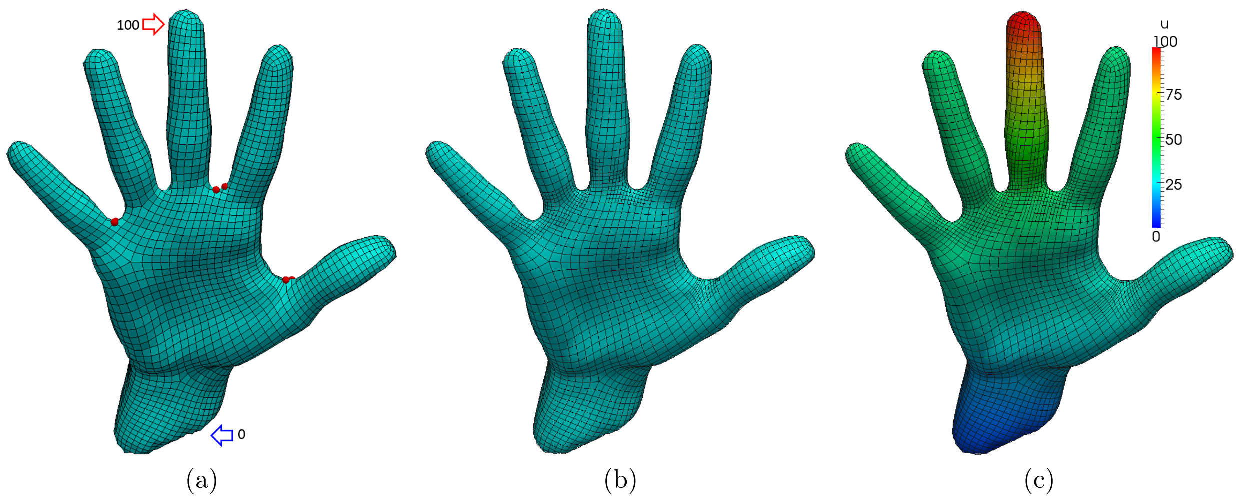


Figure 17: Solving the Laplace equation on a hand model. (a) Input mesh and Dirichlet boundary condition; (b) truncated T-spline surface; and (c) solution field on Bézier elements.

some of these children are also used as the discarded children in developing truncated basis functions ($\mathcal{I}'_T$). The surface after refinement is represented as

$$S' = \sum_{j \in \mathcal{I}'_C} P'_j B'_j + \sum_{i \in \mathcal{I}'_T} P'_i \, trunB_i = \sum_{j \in \mathcal{I}'_C} P'_j B'_j + \sum_{i \in \mathcal{I}_P} P_i \, trunB_i, \tag{24}$$

where $P'_i = P_i$ for all $i \in \mathcal{I}'_T$ and $\mathcal{I}'_T = \mathcal{I}_P$. We can rewrite Eq. (14) as

$$P'_j = \sum_{i \in \mathcal{I}_F \cup \mathcal{I}_P} c_{ij} P_i, \quad \forall j \in \mathcal{I}'_C, \tag{25}$$

where $c_{ij} = 0$ if $B'_j \notin chd(B_i)$. Besides, the truncated basis functions are obtained according to Eq. (5), and we have

$$trunB_i = B_i - \sum_{j \in \mathcal{I}'_C} c_{ij} B'_j, \quad \forall i \in \mathcal{I}'_T. \tag{26}$$

Plugging Eqs. (25, 26) into Eq. (24), we have

$$\begin{aligned} S' &= \sum_{j \in \mathcal{I}'_C} B'_j \sum_{i \in \mathcal{I}_F \cup \mathcal{I}_P} c_{ij} P_i + \sum_{i \in \mathcal{I}_P} P_i \left(B_i - \sum_{j \in \mathcal{I}'_C} c_{ij} B'_j \right) \\ &= \sum_{i \in \mathcal{I}_F} P_i \sum_{j \in \mathcal{I}'_C} c_{ij} B'_j + \sum_{i \in \mathcal{I}_P} P_i B_i. \end{aligned} \tag{27}$$

A fully refined basis functions can be represented by its children, so we have

$$B_i = \sum_{j \in \mathcal{I}'_C} c_{ij} B'_j, \quad \forall i \in \mathcal{I}_F. \tag{28}$$

Plugging Eq. (28) into Eq. (27), we have

$$S' = \sum_{i \in \mathcal{I}_F} P_i B_i + \sum_{i \in \mathcal{I}_P} P_i B_i = S. \tag{29}$$

Therefore, the truncated T-spline surface does not change during the truncated T-spline quadtree subdivision. □

Proposition 2. *Given a tensor-product B-spline mesh, after recursively performing truncated T-spline quadtree subdivision, the underlying (truncated) basis functions form a partition of unity and they are also (globally) linearly independent on the entire parametric domain.*

Proof. We use induction to prove this proposition. It is trivial to verify the starting case since B-spline basis functions form a partition of unity and they are linearly independent. Assume we have a T-mesh $\mathbb{T}$ obtained by recursively performing truncated T-spline quadtree subdivision on the tensor-product B-spline mesh and the (truncated) basis functions of $\mathbb{T}$ form a partition of unity and they are (globally) linearly independent. We next further perform truncated T-spline quadtree subdivision and prove that partition of unity and linear independence still hold for the refined T-mesh $\mathbb{T}'$. Among the basis functions of $\mathbb{T}$, let $\mathcal{I}_F$ denote the index set of fully refined basis functions, $\mathcal{I}_P$ denote the partially refined basis functions, and $\mathcal{I}_0$ denote the uninfluenced (truncated) basis functions.

First, let us prove the partition of unity property. According to assumption, we have

$$\sum_{i\in\mathcal{I}_F} B_i + \sum_{i\in\mathcal{I}_P} B_i + \sum_{i\in\mathcal{I}_0} trunB_i = 1. \tag{30}$$

Note that all the truncated basis functions of $\mathbb{T}$ are included in $\mathcal{I}_0$ due to truncated T-spline quadtree subdivision. For convenience in $\mathcal{I}_0$, we employ $trunB_i$ for truncated and non-truncated basis functions, where *trun*$=B_i$ if B_i is non-truncated. After refinement, the children of fully refined basis functions are denoted as $\mathcal{I}'_C$. They are defined on $\mathbb{T}'$, and some of them are also the discarded children of partially refined basis functions. Eq. (30) becomes

$$\begin{aligned}&\sum_{i\in\mathcal{I}_F}\sum_{j\in\mathcal{I}'_C} c_{ij}B'_j + \sum_{i\in\mathcal{I}_P}\sum_{j\in\mathcal{I}'_C} c_{ij}B'_j + \sum_{i\in\mathcal{I}_P}\left(B_i - \sum_{j\in\mathcal{I}'_C} c_{ij}B'_j\right) + \sum_{i\in\mathcal{I}_0} trunB_i\\ =&\sum_{j\in\mathcal{I}'_C} B'_j \sum_{i\in\mathcal{I}_F\cup\mathcal{I}_P} c_{ij} + \sum_{i\in\mathcal{I}_P} trunB_i + \sum_{i\in\mathcal{I}_0} trunB_i.\end{aligned} \tag{31}$$

Note that via truncated T-spline quadtree subdivision, c_{ij} are exactly the coefficients to update control points in B-spline refinement. For all $i \in \mathcal{I}_F \cup \mathcal{I}_P$, c_{ij} are the coefficients used to calculate new control points, so we have $\sum_{i\in\mathcal{I}_F\cup\mathcal{I}_P} c_{ij} = 1$. Therefore the right hand side of Eq. (31) is the summation of (truncated) basis functions of $\mathbb{T}'$, which means partition of unity still holds after refinement. To this end, we have proved that truncated T-spline basis functions form a partition of unity.

Next we prove the linear independence of truncated T-spline basis functions with the help of Theorem 3.7 in [2]. In other words, we need to prove that

$$\sum_{i\in\mathcal{I}'_C} \alpha_i B'_i + \sum_{i\in\mathcal{I}'_T} \alpha_i trunB_i + \sum_{i\in\mathcal{I}_0} \alpha_i trunB_i = 0 \tag{32}$$

if and only if

$$\alpha_i = 0, \quad \forall i \in \mathcal{I}'_C \cup \mathcal{I}'_T \cup \mathcal{I}_0. \tag{33}$$

We divide the entire domain Ω into two parts: Ω_F and $\Omega\backslash\Omega_F$, where $\Omega_F = \bigcup_{i\in\mathcal{I}_F} suppB_i$ denotes the support of the fully refined basis functions. Note that we also have $\Omega_F = \bigcup_{i\in\mathcal{I}'_C} suppB'_i$ because all the fully refined basis functions can be replaced by their children. Therefore, none of the children basis functions in $\mathcal{I}'_C$ has support on $\Omega\backslash\Omega_F$. Under the truncated T-spline quadtree subdivision, all the uninfluenced basis functions ($\mathcal{I}_0$) and all the truncated basis functions ($\mathcal{I}'_T$) have support on $\Omega\backslash\Omega_F$. Moreover, a truncated basis function is the same as its mother basis function on $\Omega\backslash\Omega_F$ because none of its discarded children has influence on $\Omega\backslash\Omega_F$. Then on $\Omega\backslash\Omega_F$, the uninfluenced basis functions and the truncated basis functions are the same basis functions as in $\mathbb{T}$. They are linearly independent according to the assumption, and we have $\alpha_i = 0$ for all $i \in \mathcal{I}'_T \cup \mathcal{I}_0$. Now we only need to address the 1^{st} term in Eq. (32). The children basis functions ($\mathcal{I}'_C$) form a subset of the T-spline basis functions of $\mathbb{T}'$. According to Theorem 3.7 in [2], given a T-mesh with linearly independent T-spline basis functions and a new T-mesh obtained by adding new nodes[2], the T-spline basis functions defined on the new T-mesh are linearly independent. Therefore, all the T-spline basis functions of $\mathbb{T}'$ are linearly independent, and basis functions in $\mathcal{I}'_C$ are also linear independent. In other words, we have $\alpha_i = 0$ for all $i \in \mathcal{I}'_C$. To this end, we have proved that Eqs. (32, 33) holds. Therefore in addition to the partition of unity property, the truncated T-spline basis functions are also linearly independent.

$\square$

References

[1] P. B. Bornermann and F. Cirak. A subdivision-based implementation of the hierarchical B-spline finite element method. *Computer Methods in Applied Mechanics and Engineering*, 253:584–598, 2013.

[2] A. Buffa, D. Cho, and G. Sangalli. Linear independence of the T-spline blending functions associated with some particular T-meshes. *Computer Methods in Applied Mechanics and Engineering*, 199:1437–1445, 2010.

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

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

[5] T. Dokken, T. Lyche, and K. F. Pettersen. Polynomial splines over locally refined Box-partitions. *Computer Aided Geometric Design*, 30:331–356, 2013.

[6] M. R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199(58):264 – 275, 2010.

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

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

[9] P. L. Gould. *Introduction to Linear Elasticity.* Springer, 3rd edition, 2013.

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

[11] K. A. Johannessen, T. Kvamsdal, and T. Dokken. Isogeometric analysis using LR B-splines. *Computer Methods in Applied Mechanics and Engineering*, 269(0):471–514, 2014.

[12] R. Kraft. Adaptive and linearly independent multilevel B-splines. In A. L. Méhauté, C. Rabut, and L. L. Schumaker, editors, *Surface Fitting and Multiresolution Methods*, pages 209–218. Vanderbilt University Press, 1997.

[13] L. Liu, Y. Zhang, and X. Wei. Handling extraordinary nodes with weighted T-spline basis functions. *Procedia Engineering*, 124:161–173, 2015. 24th International Meshing Roundtable.

[14] L. Liu, Y. Zhang, and X. Wei. Weighted T-splines with application in reparameterizing trimmed NURBS surfaces. *Computer Methods in Applied Mechanics and Engineering*, 295:108–126, 2015.

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

[16] M. A. Scott, X. Li, T. W. Sederberg, and T. J. R. Hughes. Local refinement of analysis-suitable T-splines. *Computer Methods in Applied Mechanics and Engineering*, 213-216:206–222, 2012.

[17] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. A. Bordas, T. J. R. Hughes, and T. W. Sederberg. Isogeometric boundary element analysis using unstructured T-splines. *Computer Methods in Applied Mechanics and Engineering*, 254:197–221, 2013.

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

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

[20] J. Stam. Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values. *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pages 395–404, 1998.

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

[22] W. Wang, Y. Zhang, G. Xu, and T. J. R. Hughes. Converting an unstructured quadrilateral/hexahedral mesh to a rational T-spline. *Computational Mechanics*, 50:65–84, 2012.

[23] X. Wei, Y. Zhang, T. J. R. Hughes, and M. A. Scott. Extended truncated hierarchical Catmull-Clark subdivision. *Computer Methods in Applied Mechanics and Engineering*,, 2015.

[24] X. Wei, Y. Zhang, T. J. R. Hughes, and M. A. Scott. Truncated hierarchical Catmull-Clark subdivision with local refinement. *Computer Methods in Applied Mechanics and Engineering*, 291:1–20, 2015.

[2]The term *anchor* is used in [2], which is equivalent to the definition of node in this paper.