ICES REPORT 15-15

June 2015

Extended Truncated Hierarchical Catmull-Clark Subdivision

by

Xiaodong Wei, Yongjie Jessica Zhang, Thomas J.R. Hughes, and Michael A. Scott

The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: Xiaodong Wei, Yongjie Jessica Zhang, Thomas J.R. Hughes, and Michael A. Scott, "Extended Truncated Hierarchical Catmull-Clark Subdivision," ICES REPORT 15-15, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, June 2015.

Extended Truncated Hierarchical Catmull-Clark Subdivision

Xiaodong Wei[a], Yongjie Jessica Zhang[a,*], Thomas J.R. Hughes[b], Michael A. Scott[c]

[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*
[c]*Department of Civil and Environmental Engineering, Brigham Young University, Provo, UT 84602, USA*

Abstract

In this paper we present an *extended Truncated Hierarchical Catmull-Clark Subdivision* (eTHCCS) method, which improves the efficiency of local refinement in Truncated Hierarchical Catmull-Clark Subdivision (THCCS). We first generalize Stam's Catmull-Clark basis functions for elements with more than one extraordinary node. In this manner we build a set of basis functions over arbitrary quadrilateral meshes and enable isogeometric analysis on such meshes without any preprocessing. Then, a new basis-function-insertion scheme is developed with the aid of the truncation mechanism, which refines one-ring neighboring elements rather than two-ring neighborhoods. Therefore, eTHCCS significantly improves the efficiency of local refinement compared with THCCS, as demonstrated by one benchmark problem and several complex models. Moreover, eTHCCS is also proved to preserve the input geometry and produce nested spaces.

Keywords: Local refinement, truncated hierarchical Catmull-Clark subdivision, arbitrary control meshes, isogeoemtric analysis.

1. Introduction

Local refinement and arbitrary topology have been two key concerns since isogeometric analysis [12, 5] was proposed. Isogeometric analysis aims to integrate engineering design with simulation by employing the basis used for geometric design in analysis. Local refinement was studied in hierarchical B-splines [8, 14, 25, 2], T-splines [23, 22, 1], PHT-splines [6], LR-splines [7, 13], THB-splines [9, 10, 29] and Truncated Hierarchical Catmull-Clark Subdivision (THCCS) [28]. Among them, hierarchical B-splines and THB-splines are restricted to a topologically rectangular domain, and thus they have limited applications in complex geometries. T-splines support arbitrary topologies [27, 26, 16], but the original development of T-splines may violate one prerequisite requirement in analysis, i.e., linear independence [3]. A mildly restricted subset, analysis-suitable T-splines [15, 21], was subsequently developed to address the linear independence issue. However, local refinement of analysis-suitable T-splines may propagate beyond the domain of interest via excessive T-junction extension. PHT-splines support arbitrary topology and element-wise local refinement with a trade-off in almost twice the number of degrees of freedom and reduced continuity (C^1 for cubic splines). THCCS, on the other hand, addresses both local refinement and arbitrary topology without introducing extra degrees of freedom and attains C^2 continuity.

THCCS [28] was developed based on Catmull-Clark subdivision [4, 30, 19]. Catmull-Clark subdivision is a popular quadrilateral-based subdivision scheme that is generalized from mid-knot insertion of bi-cubic B-splines to arbitrary topologies. A Catmull-Clark surface is represented by a quadrilateral mesh, which is obtained by an iterative and global refinement of an initial coarse quadrilateral mesh. The sequence of refined meshes converges to a limit surface that is C^2-continuous everywhere except C^1-continuous at extraordinary vertices (a vertex is called an *extraordinary vertex* if it has other than four quadrilaterals adjacent to it, and it is a regular vertex otherwise). In this paper, *vertex* and *quadrilateral* are used in the physical domain, and correspondingly in the parametric domain, their preimages are referred to as *node* and *element*, respectively. Alternatively, the limit surface can be directly

*Corresponding author: Yongjie Jessica Zhang. Tel: (412) 268-5332; Fax: (412) 268-3348; Email: *jessicaz@andrew.cmu.edu*.

evaluated from the initial quadrilateral mesh by Stam's basis functions [24]. The quadrilateral mesh is called the *control mesh* of the Catmull-Clark surface. Based on Stam's Catmull-Clark basis functions, THCCS takes advantage of the hierarchical structure and truncation mechanism to support local refinement and preserve geometry. THCCS has desired properties for geometric design and analysis such as partition of unity, convex hull, linear independence, and support of arbitrary topology. However, two requirements in THCCS restrict the efficiency of local refinement: (1) at most one extraordinary vertex is allowed in each quadrilateral in the control mesh; and (2) for each to-be-refined basis function, all its two-ring neighboring elements have to be refined. Requirement (1) is inherited from the development of Stam's Catmull-Clark basis functions [24]. In THCCS, this requirement is satisfied by refining all the quadrilaterals that contain more than one extraordinary control point (such quadrilaterals are called *invalid*). The input quadrilateral mesh may contain a large number of quadrilaterals that violate Requirement (1), leading to almost global refinement. Requirement (2) follows the basis-function-refinement manner in hierarchical B-splines [14, 11, 25, 2]. For cubic hierarchical B-splines and Catmull-Clark subdivision, however, such basis-function-refinement needs to refine all the two-ring neighboring elements of each to-be-refined basis function, which is not efficient in capturing abrupt changes in geometric or solution features.

To improve the efficiency of local refinement in THCCS, in this paper we develop the *extended Truncated Hierarchical Catmull-Clark Subdivision* (eTHCCS). We first make a straightforward generalization of Stam's Catmull-Clark basis functions to enable their direct application on arbitrary control meshes, where more than one vertex is allowed in a single quadrilateral. This generalization eliminates the requirement of refining invalid quadrilaterals, and provides a set of basis functions for isogeometric analysis on arbitrary quadrilateral meshes. The analysis-suitability on irregular quadrilateral meshes is also studied for the generalized Catmull-Clark basis functions. The main contribution of this paper is the development of a new *basis-function-insertion* scheme for local refinement to improve its efficiency by releasing the restriction on the to-be-refined region. In THB-splines or THCCS, the to-be-refined region is restricted to be the support of to-be-refined basis functions, which is the union of two-ring neighboring elements of the associated nodes in the case of cubic splines. In addition to preserving all the properties of THCCS, eTHCCS therefore achieves a higher efficiency in local refinement.

The remainder of this paper is organized as follows. Section 2 reviews the basic concepts of THCCS. Section 3 introduces the detailed development of eTHCCS. In Section 4, we study the geometry preservation and nested property of eTHCCS. We then demonstrate the efficiency of the proposed method in Section 6, and conclude the paper in Section 7.

2. A Review of Truncated Hierarchical Catmull-Clark Subdivision

In this section we briefly review the key concepts of THCCS. For details one may refer to related literature [4, 30, 24, 14, 25, 2, 9, 28].

2.1. Stam's Catmull-Clark Basis Functions

THCCS was developed based on Stam's Catmull-Clark basis functions. It is a natural starting point to introduce Catmull-Clark subdivision and its basis functions. Catmull-Clark is one of the most popular subdivision schemes used in the CAD community. The refinement (or subdivision) in Catmull-Clark subdivision is generalized from mid-knot insertion of bi-cubic B-splines [4]. Each new vertex in the refined mesh is calculated via a weighted average of neighboring vertices in the original mesh. This linear relationship can be expressed by a so-called *subdivision matrix*. The repeated global refinement generates a sequence of meshes, $\mathcal{M}^0, \ldots, \mathcal{M}^n$, where $\mathcal{M}^0$ is the initial input quadrilateral mesh, and n is the number of subdivisions. As n goes to infinity, $\mathcal{M}^n$ converges to a *limit surface*. An alternative way to obtain the limit surface takes advantage of the Stam's basis functions [24]. These basis functions are analogous to B-spline basis functions, whereas each mesh $\mathcal{M}^\ell$ $(0 \le \ell \le n)$ behaves as a control mesh. Thus we can express the limit surface $\mathcal{S}_{\text{limit}}$ by a mapping from the parametric domain to the physical domain,

$$\mathcal{S}_{\text{limit}}(\xi, \eta) = \sum_{i=1}^{N^\ell} B_i^\ell(\xi, \eta) P_i^\ell, \tag{1}$$

where $B_i^\ell(\xi,\eta)$ are Stam's Catmull-Clark basis functions, ξ and η are parametric coordinates, P_i^ℓ are control points (or vertices) in the physical domain, and N^ℓ is the number of basis functions of $\mathcal{M}^\ell$. We call ℓ the subdivision *level*. Without loss of generality, we introduce the basis functions at Level ℓ as follows.

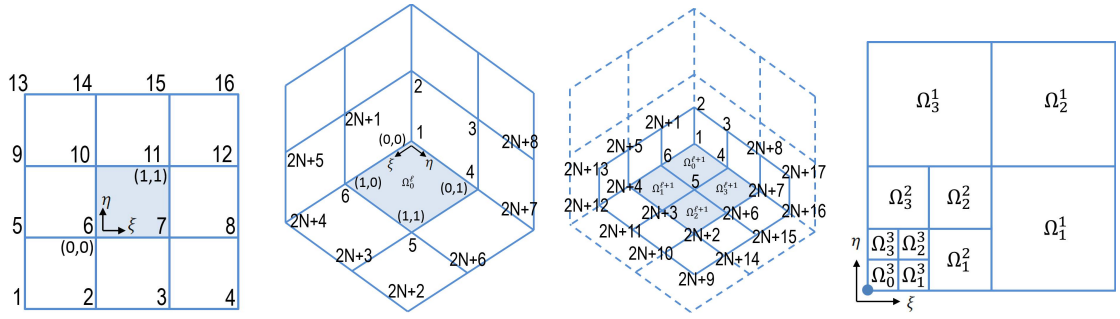

Figure 1: Local parametric elements. (a) A regular element (blue); (b) an irregular element (blue); (c) subdivision of (b); and (d) subdivisions in the parametric domain. The local parametric domain is $[0,1]\times[0,1]$. Nodes are locally numbered with respect to the element marked in blue.

A Catmull-Clark subdivision surface generally does not have a global rectangular parametric domain due to the presence of extraordinary nodes. A parametric *element* is locally associated with each quadrilateral in the control mesh and each control point has a corresponding *node* in the parametric domain. If an element contains any extraordinary node, it is irregular, and otherwise it is regular. Fig. 1(a) shows a local mesh surrounding a regular element (shaded in blue). A total of 16 basis functions have support over this regular element because a Catmull-Clark basis function has local support over its two-ring neighboring elements. They are actually bi-cubic uniform B-spline basis functions,

$$B_i^0(\xi,\eta) = b_{(i-1)\%4}(\xi)b_{(i-1)/4}(\eta), \quad i = 1,2,\ldots,16, \tag{2}$$

where "$\%$" and "/" represent the remainder and division, respectively, and for $t\in[0,1]$ we have

$$b_0(t) = (1-3t+3t^2-t^3)/6, \qquad b_1(t) = (4-6t^2+3t^3)/6, \tag{3}$$

$$b_2(t) = (1+3t+3t^2-3t^3)/6, \qquad b_3(t) = t^3/6. \tag{4}$$

Fig. 1(b) shows a local mesh surrounding a valence-3 vertex[1], where an irregular element (Ω_0^ℓ) is marked in blue. The surrounding $2N+8$ (N is the valence number and here $N=3$) basis functions have support on Ω_0^ℓ, whose associated vertices are locally labeled in the manner shown in Fig. 1(b). The $2N+8$ basis functions over Ω_0^ℓ are derived by infinitely subdividing Ω_0^ℓ [24]. In the first subdivision, Ω_0^ℓ is subdivided into one smaller irregular element $\Omega_0^{\ell+1}$ and three regular elements $\Omega_k^{\ell+1}$ ($k=1,2,3$); see Fig. 1(c). The new $2N+17$ vertices in Fig. 1(c) can be obtained from the $2N+8$ vertices in Fig. 1(b) via a subdivision matrix $\bar{\mathbf{A}}_{(2N+17)\times(2N+8)}$. As the basis functions are well-defined on $\Omega_k^{\ell+1}$ ($k=1,2,3$) as in Fig. 1(a), the limit surface corresponding to the 3/4 parametric domain of Ω_0^ℓ is represented as

$$(\mathbf{B}^\ell)^T\mathbf{P}^\ell = (\mathbf{B}^{\ell+1})^T\mathbf{P}^{\ell+1} = (\mathbf{B}^{\ell+1})^T\bar{\mathbf{A}}\mathbf{P}^\ell = (\bar{\mathbf{A}}^T\mathbf{B}^{\ell+1})^T\mathbf{P}^\ell, \tag{5}$$

where $\mathbf{B}^\ell = [B_1^\ell,\ldots,B_{2N+8}^\ell]^T$, $\mathbf{P}^\ell = [P_1^\ell,\ldots,P_{2N+8}^\ell]^T$, $\mathbf{B}^{\ell+1} = [B_1^{\ell+1},\ldots,B_{2N+17}^{\ell+1}]^T$, and $\mathbf{P}^{\ell+1} = [P_1^{\ell+1},\ldots,P_{2N+17}^{\ell+1}]^T$. Eq. (5) holds for any $\mathbf{P}^\ell$, so we have

$$\mathbf{B}^\ell = \bar{\mathbf{A}}^T\mathbf{B}^{\ell+1}. \tag{6}$$

The remaining 1/4 parametric domain of Ω_0^ℓ is the irregular element $\Omega_0^{\ell+1}$. We need to further subdivide $\Omega_0^{\ell+1}$ to find another 3/4 parametric domain of $\Omega_0^{\ell+1}$ where basis functions are well-defined. By repeating this procedure, the

[1]The valence number of a node is the number of elements adjacent to it.

parametric domain corresponding to the irregular element becomes exponentially smaller, as shown in Fig. 1(d). The repeated subdivision occurs in the irregular element. Note that the subdivision of $\Omega_0^{\ell+1}$ only involves the first $2N+8$ vertices in Fig. 1(c). Therefore the sub-matrix consisting of the first $2N+8$ rows of $\bar{\mathbf{A}}$ will be repeatedly used, denoted as $\mathbf{A}_{(2N+8)\times(2N+8)}$. For computational efficiency, the eigenstructure $(\mathbf{\Lambda}, \mathbf{V})$ of $\mathbf{A}$ ($\mathbf{AV} = \mathbf{V\Lambda}$) is employed such that only diagonal matrix multiplication is required. In this manner, the basis functions at Level ℓ over the irregular element Ω_0^{ℓ} is derived as

$$\mathbf{B}^{\ell}(\xi,\eta) = (\mathbf{V}^{-1})^T \mathbf{\Lambda}^{n-1} (\mathbf{P}_k \bar{\mathbf{A}} \mathbf{V})^T \mathbf{b}(\xi,\eta), \tag{7}$$

where $\mathbf{b}(\xi,\eta)$ are the uniform bi-cubic B-spline basis functions as in Eq. (2). Given parametric coordinates (ξ,η), we perform subdivision n times to restrict (ξ,η) into a regular element $(\Omega_k^n,\ k = 1,2,3)$ as in Fig. 1(d). $\mathbf{P}_k$ $(k = 1,2,3)$ is a selection matrix to locate such regular element. The configuration of $\mathbf{A}$ and $\bar{\mathbf{A}}$ can be found in the appendix A in Stam's work [24], which only depends on the valence of the extraordinary node. When the parametric values (ξ,η) approach zero, Eq. (7) is defined as a limit case where $\mathbf{\Lambda}^{n-1}$ becomes a matrix such that only its first diagonal element is non-zero. This is because in the diagonal matrix $\mathbf{\Lambda}$, all the elements are positive and smaller than 1 except for the first element, which equals to 1. Eq. (6) is also valid over $\Omega_0^{\ell+1}$ [28], where basis functions are defined by Eq. (7). Eq. (6) indicates a general relationship between basis functions at two consecutive levels. We call this relationship *refinability*, and we define high-level basis functions $\mathbf{B}^{\ell+1}$ as the *children* of low-level basis functions $\mathbf{B}^{\ell}$ in Eq. (6). Refinability is fundamental to the construction of THCCS.

However, Eq. (7) does not work for all quadrilateral meshes because its derivation requires that each quadrilateral contains at most one extraordinary node. Usually, an input quadrilateral mesh needs to be globally refined once before Stam's basis functions are applied.

2.2. Truncated Hierarchical Catmull-Clark Subdivision

THCCS [28] generalizes truncated hierarchical B-splines (THB-splines) [9] to control meshes of arbitrary topology. THB-splines were developed to further modify the hierarchical B-spline basis functions to form a partition of unity and to decrease overlapping. However, THB-splines can only be used to represent the geometries with a global parametric domain. Complex geometries unavoidably involve extraordinary nodes and thus cannot be mapped onto a global parametric domain. Based on this fact, THCCS was developed as an attempt to address local refinement on arbitrary topologies.

Starting from a valid input quadrilateral mesh, we recursively construct THCCS. The recursive manner allows us to consider two consecutive levels at one time. We now construct Level $\ell+1$ from Level ℓ. Level-ℓ basis functions are denoted as $\mathcal{B}^{\ell}$. The THCCS space can be enlarged by replacing the identified basis functions ($\mathcal{B}_r^{\ell} \subseteq \mathcal{B}^{\ell}$) with their children ($\mathrm{chd}\mathcal{B}_r^{\ell}$). After the refinement of $\mathcal{B}_r^{\ell}$, we define active Level-ℓ basis functions as $\mathcal{B}^{\ell} \backslash \mathcal{B}_r^{\ell}$, and the children basis functions of $\mathcal{B}_r^{\ell}$ ($\mathrm{chd}\mathcal{B}_r^{\ell}$) are the active Level-$(\ell+1)$ basis functions. Only active basis functions are collected into THCCS basis. Besides, if a Level-ℓ basis function $B_i^{\ell} \in \mathcal{B}^{\ell} \backslash \mathcal{B}_r^{\ell}$ has any children contained in $\mathrm{chd}\mathcal{B}_r^{\ell}$, it has to be truncated in order to form a partition of unity and to preserve the geometry. The truncation is performed by discarding active children from the refinability relationship. Recall that according to refinability, B_i^{ℓ} can be expressed as $B_i^{\ell} = \sum_{B_j^{\ell+1} \in \mathrm{chd}B_i^{\ell}} c_{ij} B_j^{\ell+1}$, where coefficients c_{ij} come from the subdivision matrix $\bar{\mathbf{A}}$. Then the truncation of B_i^{ℓ} is obtained by

$$\mathrm{trun}B_i^{\ell} = \sum_{B_j^{\ell+1} \in \mathrm{chd}B_i^{\ell} \text{ and } B_j^{\ell+1} \notin \mathrm{chd}\mathcal{B}_r^{\ell}} c_{ij} B_j^{\ell+1}. \tag{8}$$

Finally, we collect active basis functions at Level ℓ and Level $\ell+1$ into the THCCS basis. Other high levels can be constructed likewise and the THCCS basis is updated accordingly. The above procedure can be recursively performed until the desired maximum level $\ell_{\max}$ is reached.

THCCS basis functions satisfy partition of unity, convex hull property, linear independence with the support of local refinement, and arbitrary topologies. However, the basis-function-refinement nature requires refinement of two-ring neighborhood of a basis function at each refinement step, leading to the refinement of all the elements within its support. eTHCCS is therefore developed to improve the efficiency of local refinement with a new basis-function-insertion, in which we only need to refine one-ring neighboring elements. This new scheme releases the strong restriction of the minimum number of to-be-refined elements. Details of eTHCCS will be discussed next in Section 3.

3. Development of eTHCCS

In this section, we discuss how to develop eTHCCS. First, we generalize the Catmull-Clark basis functions for elements with arbitrary numbers of extraordinary nodes, eliminating the requirement of refining such elements in order to use Stam's basis functions. In this manner we build the basis suitable for isogeometric analysis over arbitrary quadrilateral meshes. Then we develop a new basis-function-insertion scheme to improve the locality of refinement, releasing the restriction of the to-be-refined region. With this scheme, we can refine even only one element at each refinement step, rather than refining two-ring neighborhoods of elements in THCCS. Finally, we use the generalized Catmull-Clark basis functions and basis-function-insertion scheme to construct eTHCCS.

3.1. Generalized Catmull-Clark Basis Functions

Here we aim to generalize the derivation of Catmull-Clark basis functions to invalid elements. Recall that an invalid element contains more than one extraordinary node. Once an invalid element is subdivided, the resulting four high-level elements are all valid over which basis functions are defined in either Eq. (2) or Eq. (7). Instead of subdividing invalid elements first and then applying the subdivision matrix $\bar{\mathbf{A}}$ to derive Eq. (7), in the following we introduce a more general subdivision matrix, denoted as $\mathbf{S}$, and directly apply it to derive generalized Catmull-Clark basis functions over an invalid element.

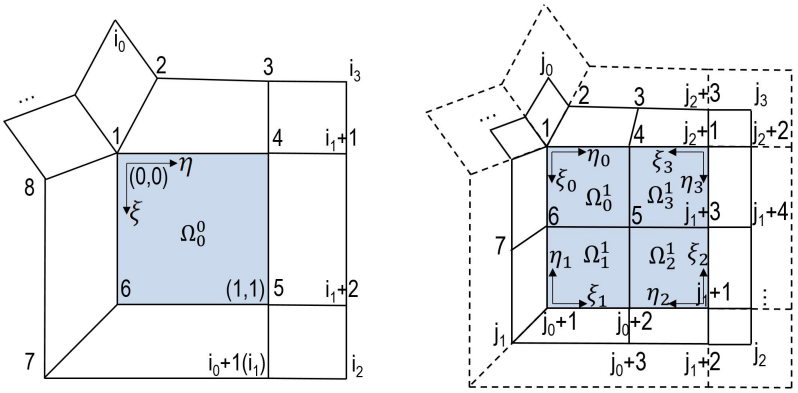

Figure 2: Generalized subdivision. (a) An invalid element Ω_0^0 with its two-ring neighboring nodes labeled; and (b) subdividing Ω_0^0 into four high-level elements Ω_k^1 ($k = 0, \ldots, 3$), whose two-ring neighboring nodes are labeled in the similar manner.

Given an invalid element, let us first locally label its two-ring neighboring nodes. We follow the manner of labeling as in Fig. 1(b), labeling the one-ring neighboring nodes of the extraordinary node clockwise. Note that any corner node of an invalid element can be an extraordinary node. For instance, in Fig. 2(a), Ω_0^0 is the invalid element under study. Let N_0, N_1, N_2 and N_3 be the valence numbers of four corner nodes of Ω_0^0, labeled as Node 1, Node 6, Node 5 and Node 4, respectively. We start labeling the one-ring neighboring nodes of Node 1 in clockwise, from 2, 3, until $i_0 = 2N_0 + 1$. Then we consider the one-ring neighboring nodes of Node 6. Among them only $2N_1 + 1 - 6 = 2N_1 - 5$ nodes remain unlabeled, and we label them from $i_0 + 1$ until i_1, where $i_1 = i_0 + 2N_1 - 5 = 2(N_0 + N_1) - 4$. Here Node 6 is valence-3, so we have $N_1 = 3$ and $i_1 = i_0 + 1$. Next for Node 5, there are $2N_2 + 1 - 6 = 2N_2 - 5$ nodes remaining unlabeled if $N_3 > 3$; otherwise, there are $2N_2 + 1 - 7 = 2N_2 - 6$ unlabeled nodes if $N_3 = 3$, because one more node was labeled already when we labeled the one-ring neighboring nodes for Node 1. We label them from $i_1 + 1$ until i_2 where $i_2 = i_1 + 2N_2 - 5 = 2(N_0 + N_1 + N_2) - 9$ ($N_3 > 3$) or $i_2 = i_1 + 2N_2 - 6 = 2(N_0 + N_1 + N_2) - 10$ ($N_3 = 3$). Similarly for Node 4, the number of unlabeled nodes is $2N_3 + 1 - 7 = 2N_3 - 6$ if $N_3 = 3$, otherwise it is $2N_3 + 1 - 8 = 2N_3 - 7$. We label them from $i_2 + 1$ until i_3 where $i_3 = 2\sum_{i=0}^{3} N_i - 16$. The labels are shown in detail in Fig. 2(a). The associated basis functions are the generalized basis functions to be derived, and we denote them in the vector form,

$$\bar{\mathbf{B}} = [\bar{B}_1, \bar{B}_2, \ldots, \bar{B}_{i_3}]^T. \tag{9}$$

Correspondingly, their control points are denotes as $\bar{\mathbf{P}} = [\bar{P}_1, \bar{P}_2, \ldots, \bar{P}_{i_3}]^T$. As shown in Fig. 2(b), we subdivide Ω_0^0 into four smaller elements Ω_k^1 ($k = 0, \ldots, 3$) and follow in the same manner to label their two-ring neighboring nodes. Note that Node 1, Node $j_0 + 1$, Node $j_1 + 1$ and Node $j_2 + 1$ have the same valence numbers as the four corner nodes of Ω_0^0 in Fig. 2(a). Similarly, we can derive the following relationships, $j_0 = 2N_0 + 1$, $j_1 = 2(N_0 + N_1) - 1$, $j_2 = 2(N_0+N_1+N_2)-3$ and $j_3 = 2(N_0+N_1+N_2+N_3)-7$. The corresponding basis functions are Stam's Catmull-Clark basis functions defined by either Eq. (2) or Eq. (7), denoted as

$$\mathbf{B} = [B_1, B_2, \ldots, B_{j_3}]^T. \tag{10}$$

Their control points are $\mathbf{P} = [P_1, P_2, \ldots, P_{j_3}]^T$, which are calculated by

$$\mathbf{P} = \mathbf{S}\bar{\mathbf{P}}, \tag{11}$$

where $\mathbf{S}$ can be directly obtained from the Catmull-Clark subdivision rule. For instance, Vertex 1 is relocated to a weighted average of its neighboring vertices, whose indices are $I = \{1, 2, 3, \ldots, i_0\}$. Then the 1st row, j-th ($j \in I$) column element of $\mathbf{S}$ is filled with the corresponding Catmull-Clark subdivision coefficient. Other elements in $\mathbf{S}$ can be filled in the same manner. The configuration of $\mathbf{S}$ is given in Appendix A. Assume that the evaluation of Ω_0^0 using $\bar{\mathbf{B}}$ yields the same limit surface as that of $\Omega_0^1 \sim \Omega_3^1$ using $\mathbf{B}$, and we have

$$\bar{\mathbf{B}}^T\bar{\mathbf{P}} = \mathbf{B}^T\mathbf{P} = \mathbf{B}^T\mathbf{S}\bar{\mathbf{P}} = (\mathbf{S}^T\mathbf{B})^T\bar{\mathbf{P}}. \tag{12}$$

Eq. (12) holds no matter what the values of $\bar{\mathbf{P}}$ are. Therefore we have

$$\bar{\mathbf{B}} = \mathbf{S}^T\mathbf{B}. \tag{13}$$

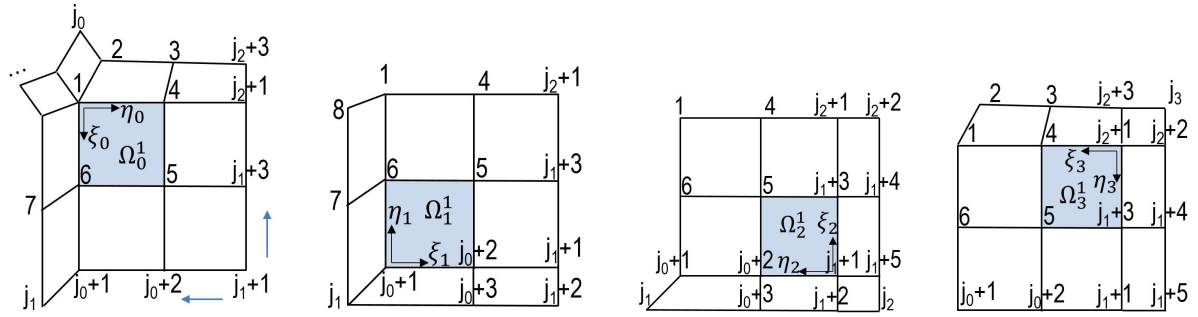

Figure 3: Generalized selections of basis functions for high-level elements Ω_k^0 ($k = 0, \ldots, 3$).

We next find the relationship between $\mathbf{B}$ and Eq. (2) or Eq. (7) so that we can obtain an explicit expression for $\bar{\mathbf{B}}$. Given a pair of parametric coordinates (ξ, η) in Ω_0^0, we can locate it in one of the four elements (Ω_k^1, where $k = 0, \ldots, 3$) in Fig. 2(b), and only the two-ring basis functions of that element are non-zero at (ξ, η). Thus $\bar{\mathbf{B}}$ is defined piecewise. For instance, if $0 \le \xi < 1/2$ and $0 \le \eta < 1/2$, (ξ, η) is located in the irregular element Ω_0^1 and only the two-ring basis functions are non-zero over Ω_0^1, as shown in Fig. 3(a). These $2N_0 + 8$ basis functions (denoted as $\mathbf{B}_0$) are selected from $\mathbf{B}$ in Eq. (13). To directly use Eq. (7), we need to sort $\mathbf{B}_0$ such that the basis functions in $\mathbf{B}_0$ have the same order as in Fig. 1(b), that is,

$$\mathbf{B}_0 = [B_1, B_2, \ldots, B_{j_0}, B_{j_1+1}, B_{j_0+2}, B_{j_0+1}, B_{j_1}, B_{j_1+3}, B_{j_2+1}, B_{j_2+3}]^T. \tag{14}$$

Note that in $\mathbf{B}_0$, the first $2N_0 + 1$ basis functions correspond to the extraordinary node and its one-ring neighboring nodes (from 1 to j_0), and the next 4 basis functions are sorted along the opposite η_0 direction, and the last 3 basis functions follow the opposite ξ_0 direction. This specific manner of sorting follows Stam's [24]. Then we can directly obtain $\mathbf{B}_0$ using Eq. (7) by replacing $\mathbf{B}^\ell$ ($\ell = 0$) with $\mathbf{B}_0$. We define a set of pairs as

$$\begin{aligned}\mathcal{P}_0 =&\{(1,1),(2,2),\ldots,(2N_0+1,j_0),(2N_0+2,j_1+1),(2N_0+3,j_0+2),(2N_0+4,j_0+1),(2N_0+5,j_1),\\ &(2N_0+6,j_1+3),(2N_0+7,j_2+1),(2N_0+8,j_2+3)\},\end{aligned} \tag{15}$$

which represents the element correspondence between $\mathbf{B}_0$ and $\mathbf{B}$. For $(i, j) \in \mathcal{P}_0$, the i-th element in $\mathbf{B}_0$ is the j-th element in $\mathbf{B}$. For instance in $(2N_0 + 2, j_1 + 1)$, the $(2N_0 + 2)$-th element in $\mathbf{B}_0$ is B_{j_1+1}, which is also the $(j_1 + 1)$-th element in $\mathbf{B}$. Moreover since $0 \le \xi < 1/2$ and $0 \le \eta < 1/2$, the basis functions in $\mathbf{B}$ other than those in $\mathbf{B}_0$ are all zero. Then we can obtain $\mathbf{B}$ by a linear transformation, $\mathbf{B} = \bar{\mathbf{P}}_0^T \mathbf{B}_0$, where $\bar{\mathbf{P}}_0$ is the so-called selection matrix with respect to Ω_0^1 and it maps the selected basis functions to the correct positions in $\mathbf{B}$. $\bar{\mathbf{P}}_0$ is a matrix of dimension $(2N_0 + 8) \times j_3$ and all its elements are zero except that the i-th row, j-th column element is 1, where $(i, j) \in \mathcal{P}_0$. Plugging $\mathbf{B} = \bar{\mathbf{P}}_0^T \mathbf{B}_0$ into Eq. (13), we have

$$\bar{\mathbf{B}} = (\bar{\mathbf{P}}_0 \mathbf{S})^T \mathbf{B}_0, \tag{16}$$

when $0 \le \xi < 1/2$ and $0 \le \eta < 1/2$.

Likewise, other selections are shown in Fig. 3(b–d) for $(\xi, \eta) \in [1/2, 1] \times [0, 1/2)$, $(\xi, \eta) \in [1/2, 1] \times [1/2, 1]$ and $(\xi, \eta) \in [0, 1/2) \times [1/2, 1]$, respectively. The corresponding selection matrix $\bar{\mathbf{P}}_k$ has the dimension $(2N_k + 8) \times j_3$ $(k = 1, 2, 3)$ and the selected basis functions $\mathbf{B}_k$ are sorted with the aid of the corresponding set of pairs $\mathcal{P}_k$. Note that Ω_1^1 in Fig. 3(b) is also an irregular element, so $\bar{\mathbf{P}}_1$ and $\mathbf{B}_1$ are similar to $\bar{\mathbf{P}}_0$ and $\mathbf{B}_0$, respectively. However in Fig. 3(c, d), Ω_2^1 and Ω_3^1 are regular elements, and thus the selected basis functions $\mathbf{B}_2$ and $\mathbf{B}_3$ can be directly obtained from Eq. (2). $\mathbf{B}_2$ and $\mathbf{B}_3$ are sorted following Fig. 1(a). For example, the set of pairs of $\mathbf{B}_2$ is defined as

$$\begin{aligned} \mathcal{P}_2 = \{&(1, j_2), (2, j_1 + 5), (3, j_1 + 4), (4, j_2 + 2), (5, j_1 + 2), (6, j_1 + 1), (7, j_1 + 3), (8, j_2 + 1), \\ &(9, j_0 + 3), (10, j_0 + 2), (11, 5), (12, 4), (13, j_1), (14, j_0 + 1), (15, 6), (16, 1)\}. \end{aligned} \tag{17}$$

In summary, we derive the basis functions over an invalid element as

$$\bar{\mathbf{B}}(\xi, \eta) = (\bar{\mathbf{P}}_k \mathbf{S})^T \mathbf{B}_k(\xi_k, \eta_k), \quad k = 0, 1, 2, 3, \tag{18}$$

where (ξ_k, η_k) are the parametric values defined in the local coordinate system of Ω_k^1 $(k = 0, \ldots, 3)$. Note that in Fig. 3, the local coordinate systems of four elements $\Omega_k^1 = [0, 1] \times [0, 1]$ $(k = 0, \ldots, 3)$ do not coincide with that of Ω_0^0. Given the parametric coordinates (ξ, η) in Ω_0^0, we need to transform them to parametric values consistent with the local coordinate system of Ω_k^1, where $k = 0, \ldots, 3$. We have

$$\begin{array}{lllllll} \Omega_0^1: & (\xi_0, \eta_0) & = (2\xi, 2\eta) & \text{if} & 0 \le \xi < \frac{1}{2} & \text{and} & 0 \le \eta < \frac{1}{2}, \\ \Omega_1^1: & (\xi_1, \eta_1) & = (2\eta, 2 - 2\xi) & \text{if} & \frac{1}{2} \le \xi \le 1 & \text{and} & 0 \le \eta < \frac{1}{2}, \\ \Omega_2^1: & (\xi_2, \eta_2) & = (2 - 2\xi, 2 - 2\eta) & \text{if} & \frac{1}{2} \le \xi \le 1 & \text{and} & \frac{1}{2} \le \eta \le 1, \\ \Omega_3^1: & (\xi_3, \eta_3) & = (2 - 2\eta, 2\xi) & \text{if} & 0 \le \xi < \frac{1}{2} & \text{and} & \frac{1}{2} \le \eta \le 1. \end{array} \tag{19}$$

Remark 3.1. In classification, we have a total of three types of elements in the control mesh: regular elements, irregular elements and invalid elements. The first two types are all valid. The basis functions over a regular element can be directly obtained from Eq.(2). For an irregular element, Eq. (7) can be used to calculate the basis functions with support on it. Eq. (18) defines the generalized Catmull-Clark basis functions over an invalid element, any node of which can be an extraordinary node. The generalized Catmull-Clark basis functions also satisfy partition of unity, the proof of which is the same as that in [28] except that generalized subdivision matrix $\mathbf{S}$ is involved. However, we do not allow all its four corners to be valence-3, in which case the basis functions are linearly dependent on the invalid element [18]. Moreover, Eq. (13) indicates refinability is also valid for generalized Catmull-Clark basis functions. Therefore we can use them to construct eTHCCS. With the generalized Catmull-Clark basis functions, preprocessing of input control meshes is no longer required to refine invalid quadrilaterals, which is a significant improvement on efficient local refinement, especially for complex quadrilateral meshes with many extraordinary points.

3.2. Basis-Function-Insertion Scheme

The original development of THCCS (or HB-splines, THB-splines) employs a basis-function-refinement scheme, which replaces a basis function with its children to enlarge the spline space. As a result, we need to refine all the elements within the support of to-be-refined basis functions. In cubic splines and Catmull-Clark subdivision, this leads to the refinement of all the two-ring neighboring elements, which is not efficient to capture abrupt change in geometric or solution features. Instead, in the following we develop a new basis-function-insertion scheme. With this scheme, we only need to select the support of inserted high-level basis functions as the to-be-refined region. In

contrast, the to-be-refined region in THB-splines and THCCS is the support of low-level basis functions. For example we consider cubic splines in Fig. 4. A basis function at Level ℓ has support over the two-ring neighboring Level-ℓ elements, as shown in blue in Fig. 4(a). A Level-(ℓ + 1) basis function has support over the one-ring neighboring Level-ℓ elements (Fig. 4(b)), and a Level-(ℓ + 2) basis function has support only on one Level-ℓ element (Fig. 4(c)). In the following, we study how to insert Level-(ℓ + 1) basis functions, which requires refinement of their one-ring neighboring elements only.

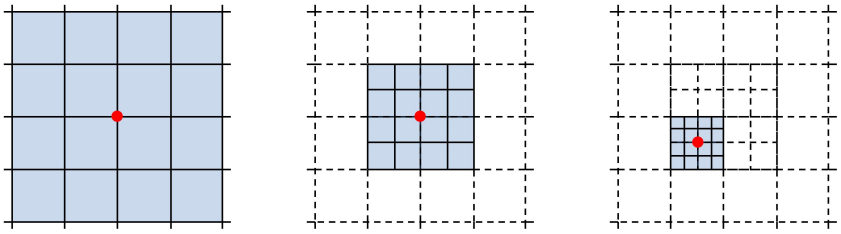

Figure 4: The support of basis functions.

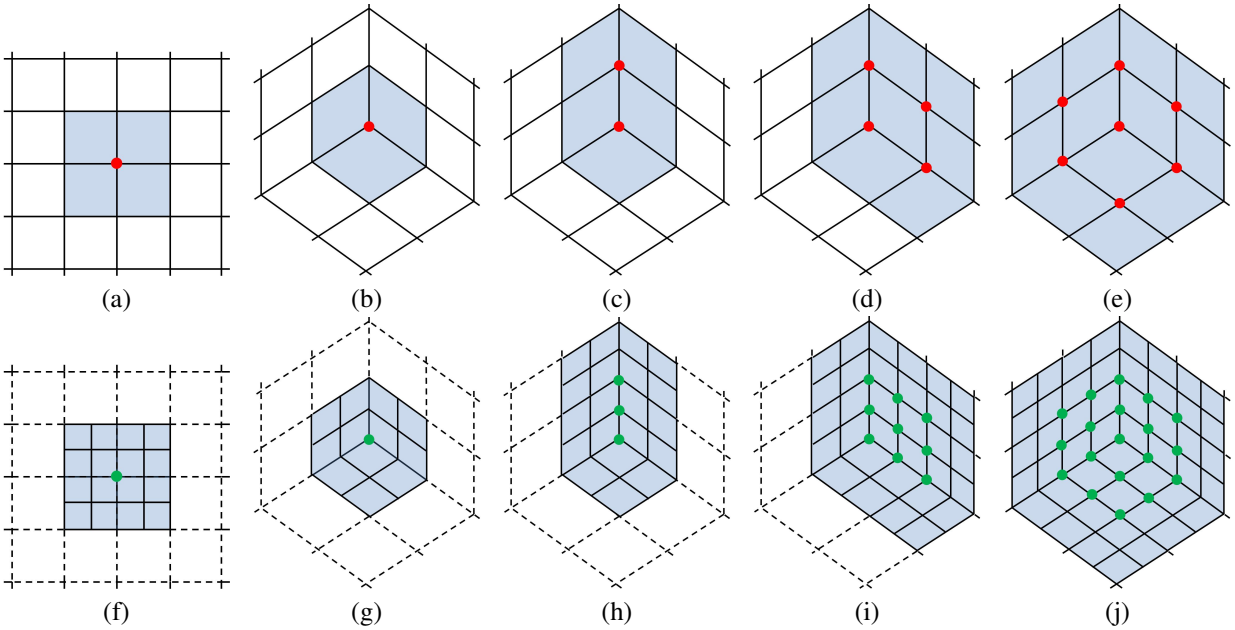

Figure 5: The identification of to-be-refined one-ring neighboring elements of a regular node (a), a valence-3 extraordinary node (b), two nodes (c), and four nodes (d). Refinement of one-ring neighboring elements of a regular node (f), a valence-3 extraordinary node (g), two nodes (h), and four nodes (i). (e) and (j) are equivalent case of the basis-function-refinement scheme.

In the basis-function-insertion scheme, at each refinement step we need to identify and refine the one-ring neighboring elements of one or multiple nodes. Local refinement can be triggered by a user-defined region or simulation error. In geometric design, designers locally refine regions of interest to add more features. In analysis, elements with large error need to be refined to improve numerical performance. Assume that we use simulation error to identify to-be-refined elements. We group all the one-ring neighboring elements for each Level-ℓ node and compare their error with a given threshold. Then a set of elements is identified as to-be-refined. In the control mesh we have both regular nodes and extraordinary nodes. Let us first study the refinement of one-ring neighboring elements of a regular node. Assume we have a local mesh at Level ℓ ($\ell \geq 0$) shown in Fig. 5(a), where a regular node is marked with a red dot and its four one-ring neighboring elements are to be refined, as marked in blue. After refinement, we obtain 16

Level-($\ell+1$) elements as shown in Fig. 5(f). Next we define the basis-function-insertion rule to insert a Level-($\ell+1$) basis function.

Basis-Function-Insertion Rule. *During local refinement around a node at Level ℓ, a Level-($\ell+1$) basis function has to be inserted associated with this node if all its two-ring Level-($\ell+1$) neighboring elements are generated.*

The basis-function-insertion rule is straightforward since a (generalized) Catmull-Clark basis function has support over its two-ring neighboring elements. Back to Fig. 5(f), the newly generated 16 Level-($\ell+1$) elements are the two-ring neighborhood of the green dot. Therefore according to the basis-function-insertion rule, there exists one basis function associated with this node. In this manner, the refinement of the one-ring neighboring elements of a regular node at Level ℓ leads to the insertion of a Level-($\ell+1$) basis function.

The same idea can be used to handle extraordinary nodes. In Fig. 5(b), the elements in blue are the one-ring neighboring elements of a valence-3 extraordinary node (red dot). They are refined and 12 Level-($\ell+1$) elements are generated, as shown in Fig. 5(g). According to the basis-function-insertion rule, the Level-($\ell+1$) basis function associated with the green dot is inserted. In general, for the extraordinary node of any valence N, by refining its one-ring neighboring elements we obtain $4N$ Level-($\ell+1$) elements together with one inserted Level-($\ell+1$) basis function.

The number of inserted high-level basis functions varies when the one-ring neighboring elements of multiple nodes are refined. For instance, Fig. 5(c) shows the one-ring neighboring elements (in blue) of two nodes of an edge. The refinement of these elements is shown in Fig. 5(h), where we observe that 3 basis functions associated with the green dots have to be inserted according to the basis-function-insertion rule. Moreover, in the case of four corner nodes of an element as shown in Fig. 5(d), 9 Level-($\ell+1$) basis functions are inserted; see Fig. 5(i). When we have the case in Fig. 5(e), the one-ring neighboring elements of those red dots are actually the two-ring neighboring elements of the valence-3 extraordinary node. Thus the refinement leads to the equivalent case of the basis-function-refinement scheme. The basis function of the extraordinary node is replaced by its children associated with green dots in Fig. 5(f). In general, the basis-function-insertion scheme yields less refinement than the basis-function-refinement scheme. Practical cases can be more complicated, but we can always follow the basis-function-insertion rule to determine where the basis functions need to be inserted. The insertion of the basis function enlarges the spline space, leading to the nested property, which will be proved later in Appendix B. The basis-function-refinement scheme can also be directly applied to truncated hierarchical B-splines.

3.3. Truncation

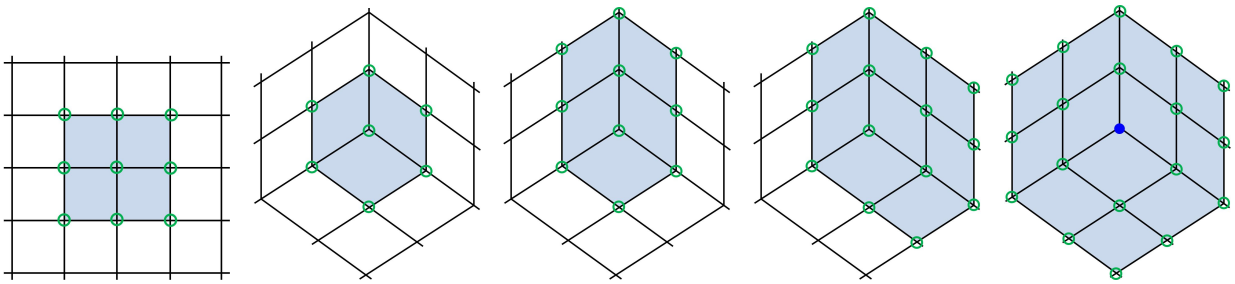

Figure 6: Five examples of Level-ℓ to-be-truncated basis functions (green circles).

Inserting high-level basis functions destroys partition of unity and changes the geometry. To resolve this issue, we apply a truncation mechanism to the neighboring low-level basis functions. A basis function needs to be truncated if any of its children is added in the spline space. Let $\mathcal{B}^\ell$ be the set of Level-ℓ basis functions. The inserted basis functions are at Level $\ell+1$ and they are added into the spline space, denoted as $\mathcal{B}^{\ell+1}$. Then some Level-ℓ basis functions ($\mathcal{B}_t^\ell$) need to be truncated if any of their children is an inserted Level-($\ell+1$) basis function, and we identify these basis functions as

$$\mathcal{B}_t^\ell = \{B_i^\ell \in \mathcal{B}^\ell \mid \mathrm{chd}B_i^\ell \cap \mathcal{B}^{\ell+1} \neq \varnothing\}. \tag{20}$$

For the five cases studied in Fig. 5, Fig. 6 illustrates how Eq. (20) is used to identify to-be-truncated basis functions. After refining the blue elements, we need to truncate basis functions associated with the corner nodes of all the refined elements, as marked with green circles. According to the refinability property, a Level-ℓ basis function B_i^ℓ can be expressed as a linear combination of its children, and we have

$$B_i^\ell = \sum_{B_j^{\ell+1} \in \mathrm{chd} B_i^\ell} c_{ij} B_j^{\ell+1}, \tag{21}$$

where c_{ij} comes from the general subdivision matrix $\mathbf{S}$ or $\bar{\mathbf{A}}$. The truncation is performed for basis functions $B_i^\ell \in \mathcal{B}_t^\ell$ by removing the children contained in $\mathcal{B}^\ell$ from the summation in Eq. (21), that is,

$$\mathrm{trun} B_i^\ell = \sum_{B_j^{\ell+1} \in \mathrm{chd} B_i^\ell \text{ and } B_j^{\ell+1} \notin \mathcal{B}^{\ell+1}} c_{ij} B_j^{\ell+1}. \tag{22}$$

Note that in Fig. 6(e), all the children of the basis function associated with the blue dot are inserted Level-$(\ell+1)$ basis functions; see Fig. 5(j). Therefore this basis function becomes a zero function after truncation. In this case we call such basis functions passive and they no longer exist in the spline space. The other non-zero basis functions are called active. The passive basis functions are actually the refined basis functions.

Truncating basis functions can help us remove the repeated contribution of high-level basis functions. Consider a Level-ℓ element Ω_k^ℓ and let I_k^ℓ be the index set of all Level-ℓ basis functions B_i^ℓ with support on it. Suppose before inserting any new Level-$(\ell+1)$ basis functions, B_i^ℓ satisfy partition of unity and we have $\sum_{i \in I_k^\ell} B_i^\ell = 1$. According to refinability, we further obtain $\sum_{i \in I_k^\ell} B_i^\ell = \sum_{i \in I_k^\ell} \sum_{j \in C_i^\ell} c_{ij} B_j^{\ell+1} = 1$, where C_i^ℓ is the index set of children basis functions of B_i^ℓ. Now we insert a Level-$(\ell+1)$ basis function ($B_{j'}^{\ell+1}$) by refinement and $B_{j'}^{\ell+1}$ has support over Ω_k^ℓ. The summation of basis functions over Ω_k^ℓ becomes $\sum_{i \in I_k^\ell} \sum_{j \in C_i^\ell} c_{ij} B_j^{\ell+1} + B_{j'}^{\ell+1} = 1 + B_{j'}^{\ell+1}$. This can be rewritten as

$$\sum_{i \in I_k^\ell} \sum_{j \in C_i^\ell} c_{ij} B_j^{\ell+1} + B_{j'}^{\ell+1} = \sum_{i \in I_k^\ell} \sum_{j \in C_i^\ell, j \neq j'} c_{ij} B_j^{\ell+1} + \sum_{i \in I_k^\ell} c_{ij'} B_{j'}^{\ell+1} + B_{j'}^{\ell+1} = \sum_{i \in I_k^\ell} \sum_{j \in C_i^\ell, j \neq j'} c_{ij} B_j^{\ell+1} + 2 B_{j'}^{\ell+1}, \tag{23}$$

where $\sum_{i \in I_k^\ell} c_{ij'} B_{j'}^{\ell+1} = B_{j'}^{\ell+1} \sum_{i \in I_k^\ell} c_{ij'} = B_{j'}^{\ell+1}$ because $\sum_{i \in I_k^\ell} c_{ij'} = 1$ holds for Catmull-Clark subdivision and knot insertion algorithm [28]. From Eq. (23) we can observe that the summation of basis functions over Ω_k^ℓ counts the inserted basis function $B_{j'}^{\ell+1}$ twice. By removing $c_{ij'} B_{j'}^{\ell+1}$ from the refinability relationship, we can achieve partition of unity. Furthermore, the geometry is preserved during local refinement, which will be proved in Appendix B.

3.4. Construction of eTHCCS

Similar to the construction of THCCS [28], we follow three steps to construct eTHCCS: identification of to-be-refined elements and to-be-truncated basis functions (Step 1), refinement of identified elements and truncation of identified basis functions (Step 2), and collection of all the hierarchical basis functions and elements (Step 3).

We start with an initial control mesh that can be any quadrilateral mesh, except the one with all its vertices that are valence-3[2]. Given the input control mesh $\mathcal{M}^0$, the initial set of eTHCCS basis functions is defined by Level-0 basis functions $\mathcal{B}^0$ associated with $\mathcal{M}^0$, and the entire domain is defined as $\mathrm{supp}\mathcal{B}^0$. The initial elements of all the quadrilaterals in $\mathcal{M}^0$ are denoted as $\mathcal{E}^0$. eTHCCS is recursively constructed up to a desired maximum level, $\ell_{\max}$ ($\ell_{\max} > 0$), which allows us to study two consecutive levels (Level ℓ and Level $\ell+1$) at one time. Suppose we have constructed Level-ℓ elements and basis functions, and now we want to construct Level $\ell+1$. Let $\mathcal{B}^\ell$ be the set of Level-ℓ basis functions and $\mathcal{E}^\ell$ be the set of Level-ℓ elements. $\mathcal{E}^\ell$ defines the sub-domain (Ω^ℓ) at Level ℓ. The eTHCCS basis functions of ℓ levels are collected in the set $\mathcal{B}^\ell_{\mathrm{eTHCCS}}$, whereas the eTHCCS elements are in $\mathcal{E}^\ell_{\mathrm{eTHCCS}}$.

Identification (Step 1). As discussed in Section 3.2 we use simulation error to identify to-be-refined elements. Then a set of elements is identified as to-be-refined if their error is larger than a given threshold, denoted as $\mathcal{E}_r^\ell$. Besides, all the basis functions associated with the corner nodes of these elements are identified as to-be-truncated basis functions ($\mathcal{B}_t^\ell$); see Fig. 6(a–e).

[2]A mesh with all valence-3 vertices produces linearly dependent blending functions [18] and thus cannot be used in analysis.

Refinement and Truncation (Step 2). Refinement of elements in $\mathcal{E}_r^\ell$ can be easily obtained by quadtree subdivision. After refinement, elements in $\mathcal{E}_r^\ell$ are defined as passive whereas the newly generated Level-$(\ell+1)$ elements ($\mathcal{E}^{\ell+1}$) are defined as active. Only active elements will be used in eTHCCS construction. All the Level-$(\ell+1)$ elements define the sub-domain at Level $\ell+1$, denoted as $\Omega^{\ell+1}$. It is obvious that $\Omega^{\ell+1} \subseteq \Omega^\ell$. We check which node has all its two-ring neighboring elements generated. According to the basis-function-insertion rule, Level-$(\ell+1)$ basis functions ($\mathcal{B}^{\ell+1}$) are inserted for such nodes and they will be added in the eTHCCS space. The corresponding Level-$(\ell+1)$ control points are calculated by the Catmull-Clark subdivision rule. Next, the Level-ℓ basis functions in $\mathcal{B}_t^\ell$ are truncated according to Eq. (22). With all its children inserted, a basis function becomes zero and it is passive, so it no longer exists in eTHCCS.

Collection (Step 3). On one hand, by refinement elements in $\mathcal{E}_r^\ell$ become passive and basis functions in $\mathcal{B}_t^\ell$ are truncated, some of which are even passive ones. Therefore, we remove $\mathcal{E}_r^\ell$ elements from $\mathcal{E}_{\text{eTHCCS}}^\ell$ and update the basis functions of $\mathcal{B}_t^\ell$ in $\mathcal{B}_{\text{eTHCCS}}^\ell$. On the other hand, we obtain new active Level-$(\ell+1)$ elements and basis functions, which are used to construct eTHCCS of $(\ell+1)$ levels. We have

$$\mathcal{E}_{\text{eTHCCS}}^{\ell+1} = \mathcal{E}_{\text{eTHCCS}}^{\ell} \cup \mathcal{E}^{\ell+1} \tag{24}$$

and

$$\mathcal{B}_{\text{eTHCCS}}^{\ell+1} = \mathcal{B}_{\text{eTHCCS}}^{\ell} \cup \mathcal{B}^{\ell+1}. \tag{25}$$

We can recursively perform Step 1 to Step 3 until the maximum level $\ell_{\max}$ is reached. The construction enlarges the spline space of eTHCCS with nested sub-domains as the level increases, that is,

$$\Omega^0 \supseteq \Omega^1 \supseteq \cdots \supseteq \Omega^{\ell_{\max}} \tag{26}$$

and

$$\text{span}\mathcal{B}_{\text{eTHCCS}}^0 \subseteq \text{span}\mathcal{B}_{\text{eTHCCS}}^1 \subseteq \cdots \subseteq \text{span}\mathcal{B}_{\text{eTHCCS}}^{\ell_{\max}}. \tag{27}$$

We will prove this property in Appendix B.

Remark 3.2. Given a local region at Level ℓ, in this paper we focus on inserting Level-$(\ell+1)$ basis functions such that we can recursively construct eTHCCS level by level. However, inserting basis functions at higher levels is also supported in our algorithm. For instance, inserting a Level-$(\ell+2)$ basis function requires refinement of a Level-ℓ element into 16 Level-$(\ell+2)$ elements, and the truncation of a Level-ℓ basis function needs its Level-$(\ell+2)$ children basis functions. The construction procedure is the same as discussed in Sections 3.2 and 3.3.

4. Examples and Discussion

In this section, we first study the analysis-suitability of generalized Catmull-Clark basis functions via three patch tests. Then we solve a benchmark problem: the Laplace equation on the *L*-shaped domain, where both THCCS [28] and eTHCCS basis functions are used to study the convergence behavior. In the end, we solve the Laplace equation over four complex models. Table 1 summarizes the statistics of the tested models.

4.1. Patch Tests

For patch tests, we solve a 2D linear elasticity problem by applying uniform tension to a square (Young's modulus $E = 1$, Poisson's ratio $\mu = 0.3$). We use three input irregular quadrilateral meshes with different numbers of extraordinary nodes, as shown in Figs. 7(a) ~ 9(a). The meshes in 8(a) and 9(a) have elements with more than one extraordinary node. In particular, all the nodes of the central element are valence-3 in 9(a). Recall that according to Peters [18], subdivision functions are not linearly independent on such elements, which, however, does not violate the global linear independence condition unless all the nodes in the mesh are valence-3. Our application only requires global linear independence, and generalized Catmull-Clark basis functions are used here. To accurately integrate an element with extraordinary nodes, we subdivide the elements into a sequence of sub-elements, as indicated in Fig. 1(d), and in each sub-element, we adopt 4×4 Gaussian integration. Two strain components in each patch test are

Table 1: Statistics of all the tested models

Models	# Input Nodes	# Input Elements	# Invalid Elements	ε	# Levels THCCS	# Levels eTHCCS	DOF-THCCS	DOF-eTHCCS	DOF Ratio
L-Regular	153	128	0	2E-4	5	7	773	391	50.6%
L-Irregular	221	192	0	2.5E-3	4	6	807	549	68.0%
Genus-3	3,068	3,072	0	0.1	5	5	4,016	3,427	85.3%
Bunny	3,023	3,021	4	0.1	5	5	4,682	3,482	74.4%
Venus	1,559	1,552	899	0.5	4	5	6,646	1,822	27.4%
Head	2,909	2,907	1,572	0.5	5	4	11,318	3,207	28.3%

Note: # stands for number and ε is the given threshold. DOF Ratio = (eTHCCS DOF)/(THCCS DOF).

shown in Figs. 7(b, c) ~ 9(b, c), where the black curves are the isoparametric lines projected onto the physical domain. We calculate the error in the L^2 norm and H^1 norm, and display them with respect to subdivision levels used for Gauss integration of irregular elements; see Figs. 7(d) ~ 9(d). The error decreases as the subdivision level increases. However, it remains of the same order when we use more than 15 levels. As discussed in [20, 17, 28], Catmull-Clark basis functions have limitations in analysis. Directly applying Gaussian quadrature over elements with extraordinary nodes introduces numerical error, because Catmull-Clark basis functions are infinite piecewise polynomials. Further study is needed to develop efficient and accurate quadrature schemes.

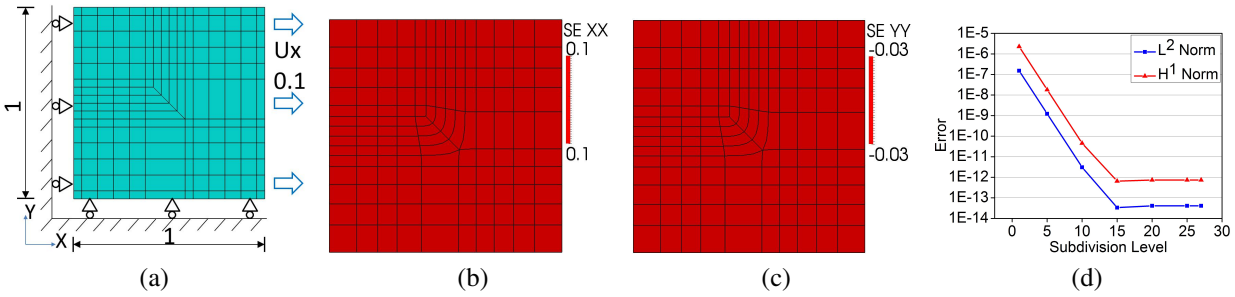

Figure 7: Patch test 1 on an irregular quadrilateral mesh with 2 extraordinary nodes using generalized Catmull-Clark basis functions. (a) The input mesh and boundary conditions; (b, c) two strain components in $X-X$ and $Y-Y$ directions, respectively; and (d) error with respect to subdivision levels used for Gauss integration.

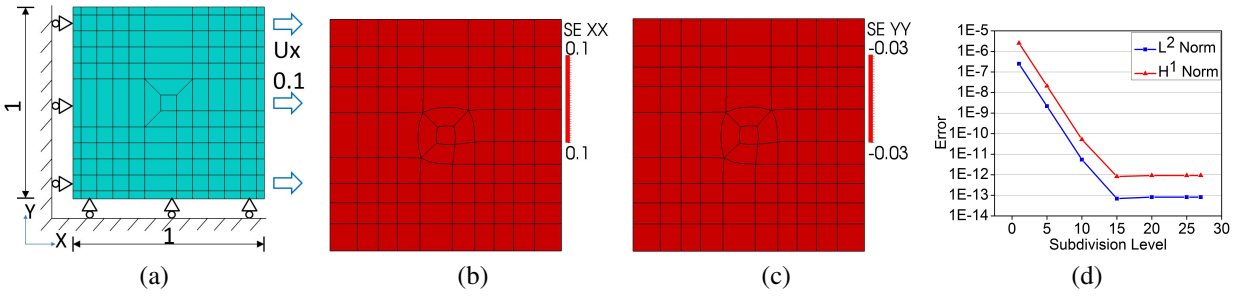

Figure 8: Patch test 2 on an irregular quadrilateral mesh with 6 extraordinary nodes using generalized Catmull-Clark basis functions. (a) The input mesh and boundary conditions; (b, c) two strain components in $X-X$ and $Y-Y$ directions, respectively; and (d) error with respect to subdivision levels used for Gauss integration.

4.2. L-shaped Problem

Fig. 10(a) shows the problem setting of the Laplace equation $\Delta u = 0$ over the L-shaped domain $[-1,1]^2 \backslash [0,1]^2$, where the Dirichlet boundary conditions (Γ_D) are strongly imposed. The analytical solution is available in polar

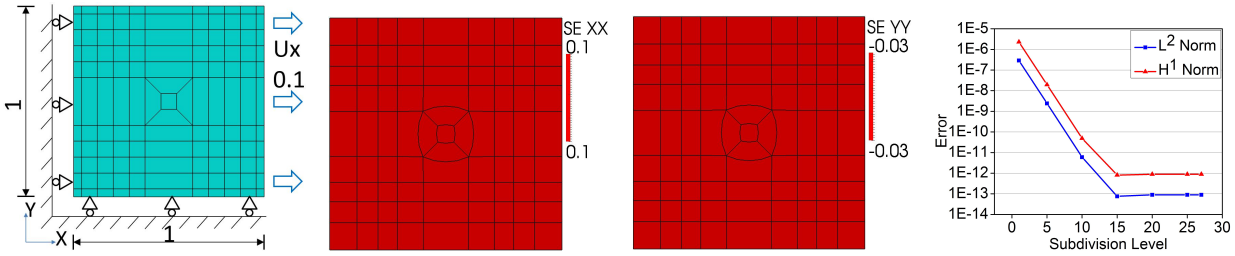

Figure 9: Patch test 3 on an irregular quadrilateral mesh with 8 extraordinary nodes using generalized Catmull-Clark basis functions. (a) The input mesh and boundary conditions; (b, c) two strain components in $X-X$ and $Y-Y$ directions, respectively; and (d) error with respect to subdivision levels used for Gauss integration.

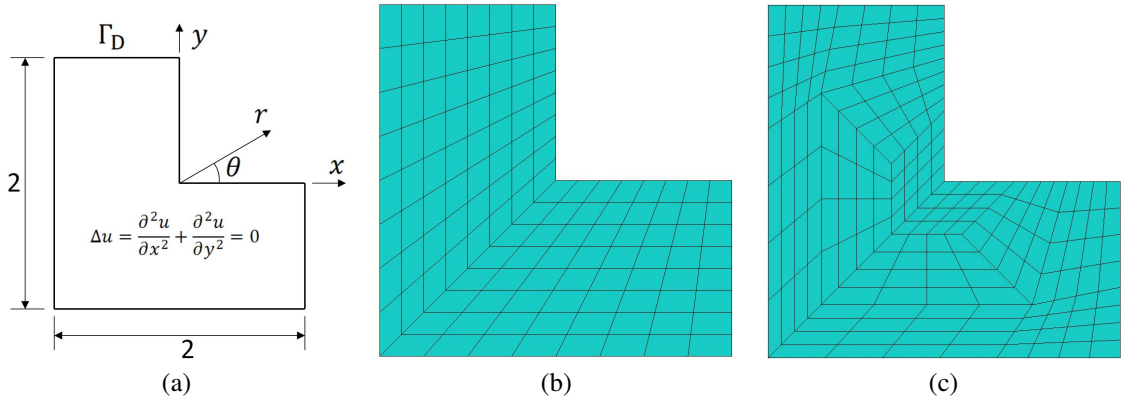

Figure 10: Laplace equation on the L-shaped domain. (a) Geometry and problem settings; (b) input regular control mesh; and (c) input irregular control mesh.

coordinates (r, θ),

$$u(r,\theta) = r^{2/3}\sin(2\theta/3 - \pi/3), \text{ where } r > 0 \text{ and } \pi/2 \le \theta \le 2\pi. \tag{28}$$

We use two input control meshes: a regular mesh and an irregular mesh, shown in Fig. 10(b, c), respectively. For each mesh, three refinement schemes are studied: uniform refinement, THCCS [28] and eTHCCS. The uniform refinement simply subdivides all the elements at each refinement step. The error is assessed in the L^2 norm and H^1 semi-norm for each element, as well as the entire domain. In THCCS or eTHCCS, to-be-refined elements are identified in terms of two-ring or one-ring neighborhood of a node. Therefore, we convert the element-wise error to the node-wise error by summing the error of two-ring (in THCCS) or one-ring (in eTHCCS) neighboring elements of the node. At each refinement step, we refine elements with the node-wise error larger than $\eta \cdot e_{\max}$, where η ($0\% < \eta \le 100\%$) refers to a prescribed percentage and $e_{\max}$ is the maximum node-wise error. In this problem, we set $\eta = 100\%$ to refine the elements with maximum node-wise error. To improve the accuracy of numerical integration surrounding an extraordinary node, if any element within its two-ring neighborhood is to be refined, we refine all its two-ring neighboring elements. The adaptive analysis terminates when the L^2 error (or H^1 semi-norm) over the entire domain is smaller than a given threshold ε. Fig. 11(a, c) shows the distribution of element-wise L^2 error using THCCS at the final step, and Fig. 11(b, d) shows this result using eTHCCS. We observe that the refinement on both regular and irregular meshes is more localized at the sharp corner, where there is a singularity in the solution. In Fig. 12, the L^2 error and H^1 semi-norm are plotted with respect to degrees of freedom (DOF). eTHCCS achieves the same accuracy with only 50.6% DOF of THCCS in the regular mesh and 68% of THCCS in the irregular mesh.

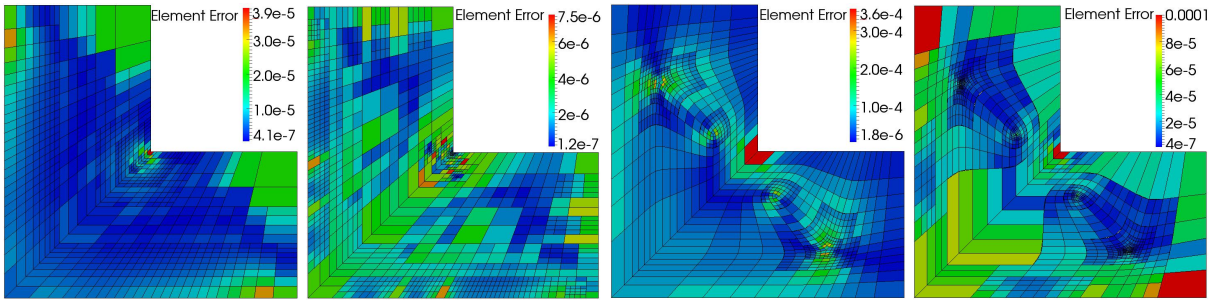

Figure 11: L^2 error distribution of numerical solutions on the L-shaped domain at the final refinement step: (a) the regular mesh with THCCS; (b) the regular mesh with eTHCCS; (c) the irregular mesh with THCCS; and (d) the irregular mesh with eTHCCS.

4.3. Complex Models

We also solve the Laplace equation on four complex models: the genus-3 model (Fig. 13), the bunny model (Fig. 14), the Venus model (Fig. 15) and the head model (Fig. 16). The input quadrilateral meshes have no elements with nodes that are all valence-3. However, the bunny model, the Venus model and the head model have invalid elements, where generalized Catmull-Clark basis functions are used. To create a solution field with abrupt local change, we strongly prescribe the solution to have a certain value, say 100, on some elements, and then set a very different value, say 1, over some elements nearby. And we study the performance of THCCS and eTHCCS. Figs. 13(a) – 16(a) show the input meshes with boundary conditions, where the elements marked in red are set at 100 and the elements in blue are set at 1. Due to the lack of analytical solutions, the L^2 error for each element is approximated by using the so-called *bubble function* [25]. Then following the same procedure as in solving the L-shaped problem, we perform adaptive analysis until the relative maximum error is smaller than a given threshold ε. The error is defined as e^n/e^0, where e^n is the maximum element-wise error after n refinement steps and e^0 is the maximum element-wise error of THCCS at the initial step. We set $\varepsilon = 0.1$ for the genus-3 and bunny models, and $\varepsilon = 0.5$ for the Venus and head models. We also set $\eta = 70\%$ to refine more elements than for the L-shaped problem at each refinement step.

At the final step, the solutions using THCCS are shown in Figs. 13(b) – 16(b) whereas the solutions using eTHCCS are shown in Figs. 13(c) – 16(c). From Table 1, eTHCCS is the most efficient method, especially in the Venus and head models, where at the final step, the degrees of freedom are only 27.4% and 28.3% of those using

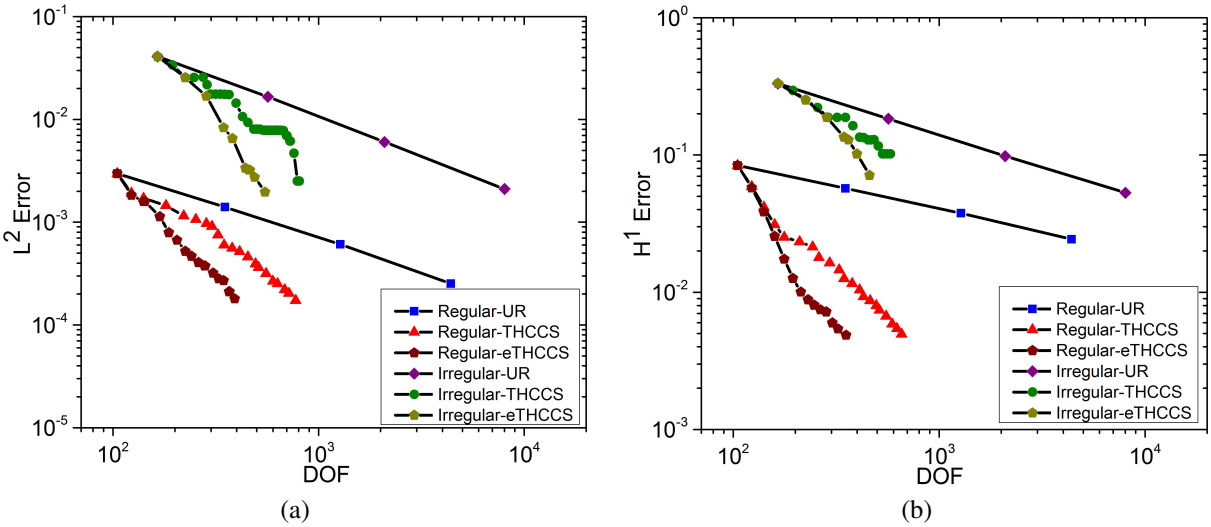

Figure 12: Convergence curves with respect to L^2 error (a) and H^1 error (b). The legend "UR" represents uniform refinement.

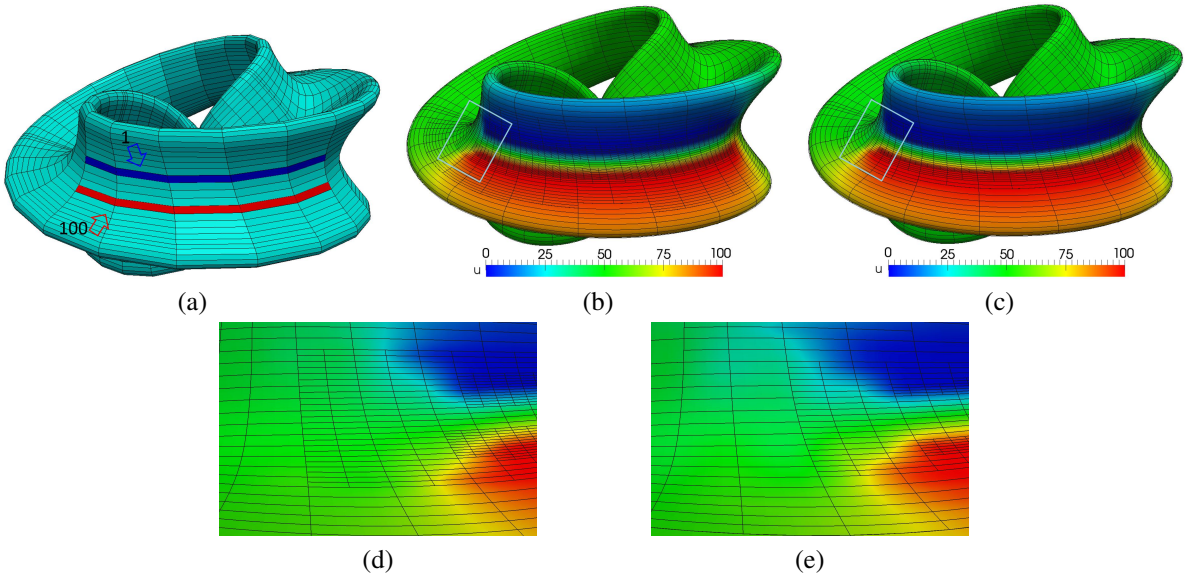

Figure 13: Solving Laplace equation over a genus-3 model. (a) Input quadrilateral mesh with boundary conditions; (b) numerical solution using THCCS; (c) numerical solution using eTHCCS; (d, e) zoom-in picture of the window in (b, c) respectively.

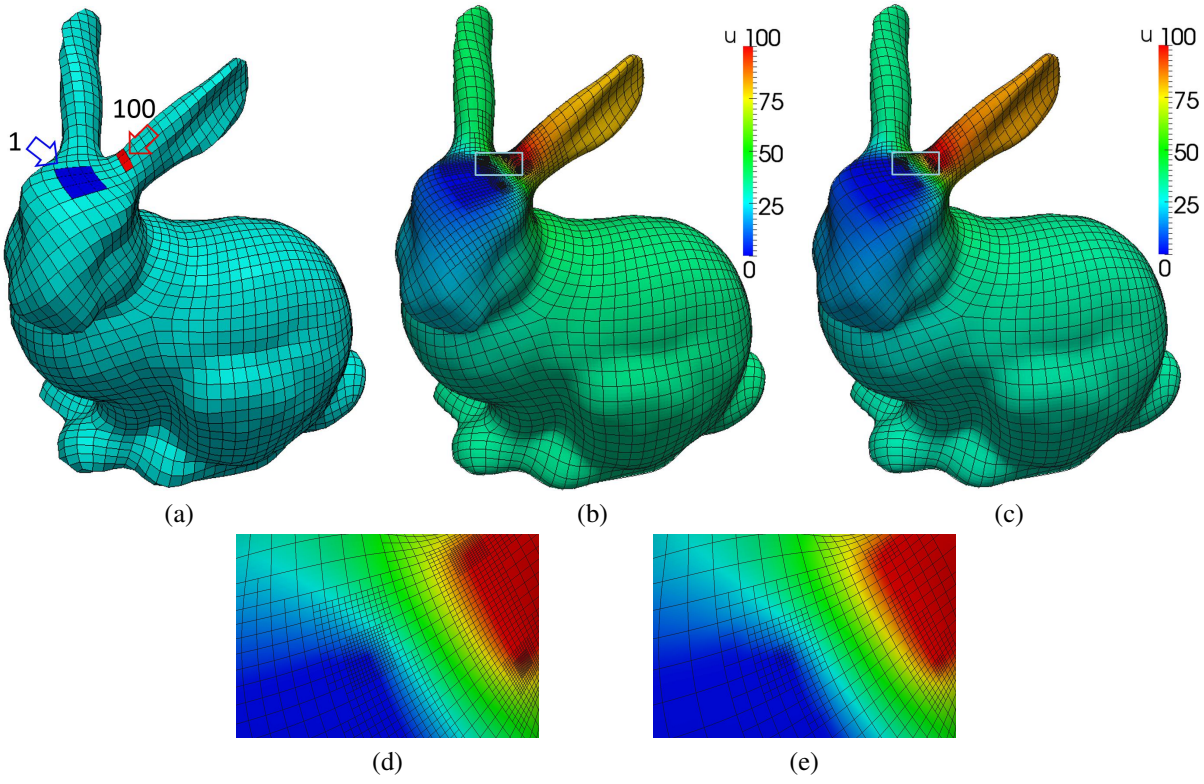

Figure 14: Solving Laplace equation over a bunny model. (a) Input quadrilateral mesh with boundary conditions; (b) numerical solution using THCCS; (c) numerical solution using eTHCCS; (d, e) zoom-in picture of the window in (b, c) respectively.

THCCS. The significant improvement mainly benefits from the generalized Catmull-Clark basis functions. As the input quadrilateral meshes of the Venus model and the head model have a large number of invalid elements, THCCS needs to refine all such elements, resulting in an almost uniform refinement; see Figs. 15(b) and 16(b). In contrast, generalized Catmull-Clark basis functions can be directly applied to the input meshes and the refinement is only performed for elements with large error, as shown in Figs. 15(c) and 16(c). On the other hand, the genus-3 model and the bunny model only have a few invalid elements. The improvement of efficiency mainly benefits from the basis-function-insertion scheme. eTHCCS only uses 85.3% DOF of THCCS in the genus-3 model and 74.4% DOF of THCCS in the bunny model. The basis-function-insertion scheme works well especially when the solution field has significant local features.

5. Conclusion and Future Work

In conclusion, we develop eTHCCS with generalized Catmull-Clark basis functions and a new basis-function-insertion scheme, aiming to improve the efficiency of local refinement. The generalized Catmull-Clark basis functions directly work on invalid elements with more than one extraordinary node, providing a basis for isogeometric analysis of arbitrary quadrilateral meshes. The basis-function-insertion scheme releases the selection restriction of the to-be-refined region in THB-splines and THCCS. In cubic splines, it allows refinement of a single element at each refinement step, while THB-splines or THCCS need to refine at least two-ring neighboring elements. In practice, as we construct

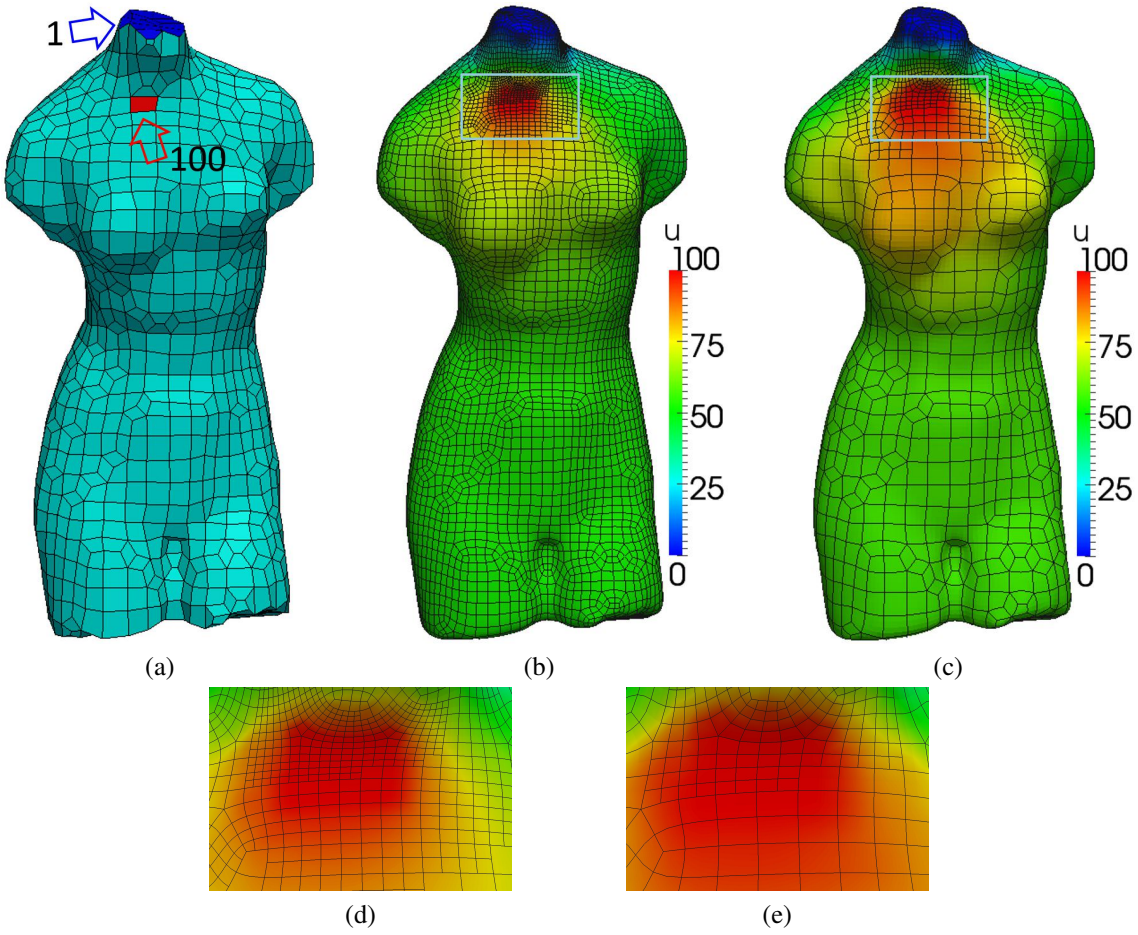

Figure 15: Solving Laplace equation over a Venus model. (a) Input quadrilateral mesh with boundary conditions; (b) numerical solution using THCCS; (c) numerical solution using eTHCCS; (d, e) zoom-in picture of the window in (b, c) respectively.

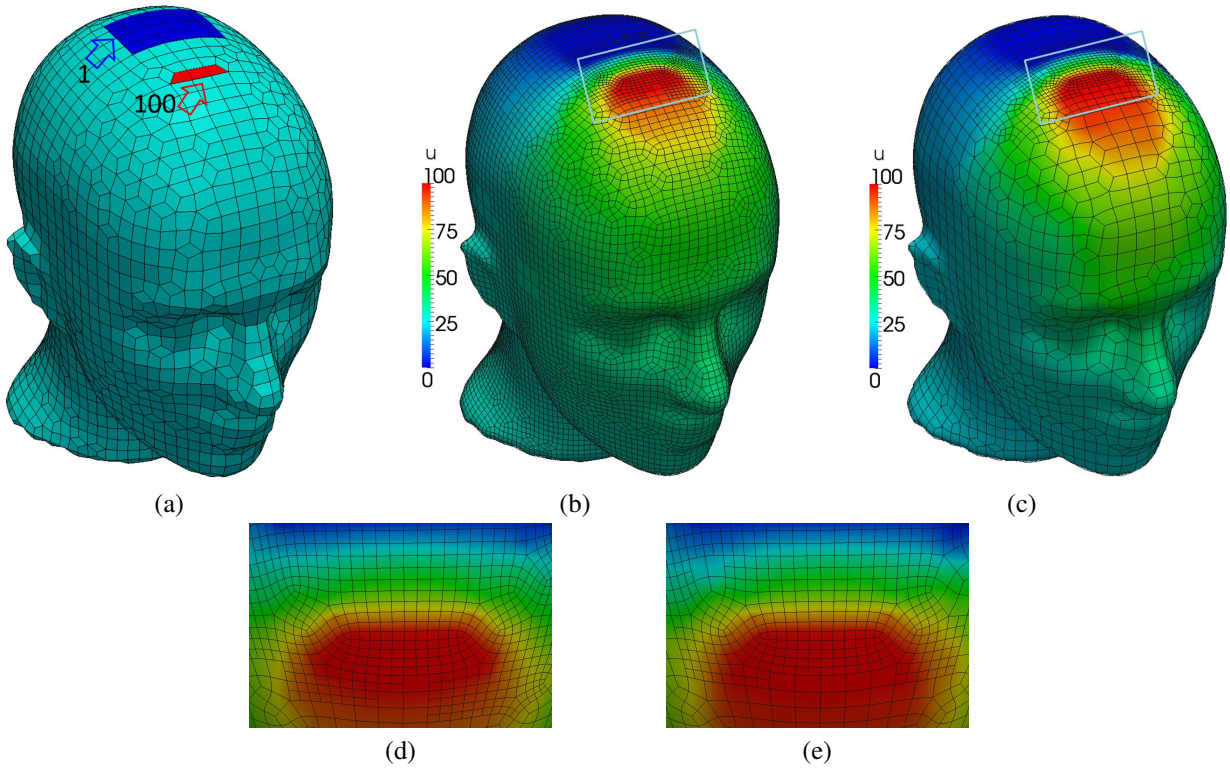

Figure 16: Solving Laplace equation over a head model. (a) Input quadrilateral mesh with boundary conditions; (b) numerical solution using THCCS; (c) numerical solution using eTHCCS; (d, e) zoom-in picture of the window in (b, c) respectively.

eTHCCS level by level, we refine one-ring neighboring elements. The basis-function-insertion scheme can also be applied to truncated hierarchical B-splines. Like THCCS, eTHCCS preserves geometry and produces nested spline spaces. eTHCCS basis functions also satisfy partition of unity, convex hull and global linear independence. We use five numerical models to test the proposed method. From the results, we can observe that eTHCCS achieves the same accuracy with fewer degrees of freedom than THCCS. In the future, it may be promising to employ the proposed method on hierarchical T-splines, where the patch test can be successfully passed without introducing huge numbers of quadrature points.

Acknowledgements

X. Wei and Y. Zhang 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: Generalized Subdivision Matrix and Selection Matrices

For an element with more than one extraordinary nodes, we develop the generalized subdivision matrix to perform local Catmull-Clark subdivision. The generalized subdivision matrix $\mathbf{S}_{j_3 \times i_3}$ is used to calculate the new j_3 vertices at Level $\ell + 1$ (Fig. 2(b)) from the neighboring i_3 vertices at Level ℓ (Fig. 2(a)). $\mathbf{S}$ is constructed directly by the Catmull-Clark subdivision rule. Following the indices locally labeled as in Fig. 2, we have

	1	2	3	4	5	6	$\cdots$	i_0	+1	+2	$\cdots$	i_1	+1	+2	$\cdots$	i_2	+1	$\cdots$	i_3
1	a_{N_1}	b_{N_1}	c_{N_1}	b_{N_1}	c_{N_1}	b_{N_1}	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	0	$\cdots$	0
2	d	d	e	e	0	0	$\cdots$	e	0	0	$\cdots$	0	0	0	$\cdots$	0	0	$\cdots$	0
3	f	f	f	f	0	0	$\cdots$	0	0	0	$\cdots$	0	0	0	$\cdots$	0	0	$\cdots$	0
$\vdots$				$\vdots$			$\ddots$				$\ddots$				$\ddots$			$\ddots$	
j_0	f	f	0	0	0	0	$\cdots$	f	0	0	$\cdots$	0	0	0	$\cdots$	0	0	$\cdots$	0
+1	b_{N_2}	0	0	c_{N_2}	b_{N_2}	a_{N_2}	$\cdots$	0	c_{N_2}	b_{N_2}	$\cdots$	c_{N_2}	0	0	$\cdots$	0	0	$\cdots$	0
+2	e	0	0	0	0	d	$\cdots$	0	0	0	$\cdots$	e	0	0	$\cdots$	0	0	$\cdots$	0
$\vdots$				$\vdots$			$\ddots$				$\ddots$				$\ddots$			$\ddots$	
j_1	0	0	0	0	0	f	$\cdots$	0	0	0	$\cdots$	f	0	0	$\cdots$	0	0	$\cdots$	0
+1	c_{N_3}	0	0	b_{N_3}	a_{N_3}	b_{N_3}	$\cdots$	0	b_{N_3}	c_{N_3}	$\cdots$	0	c_{N_3}	b_{N_3}	$\cdots$	c_{N_3}	0	$\cdots$	0
+2	0	0	0	0	d	e	$\cdots$	0	d	e	$\cdots$	0	0	0	$\cdots$	e	0	$\cdots$	0
$\vdots$				$\vdots$			$\ddots$				$\ddots$				$\ddots$			$\ddots$	
j_2	0	0	0	0	f	0	$\cdots$	0	f	0	$\cdots$	0	0	0	$\cdots$	f	0	$\cdots$	0
+1	b_{N_4}	c_{N_4}	b_{N_4}	a_{N_4}	b_{N_4}	c_{N_4}	$\cdots$	0	0	0	$\cdots$	0	b_{N_4}	c_{N_4}	$\cdots$	0	c_{N_4}	$\cdots$	c_{N_4}
+2	0	0	0	d	e	0	$\cdots$	0	0	0	$\cdots$	0	d	e	$\cdots$	0	0	$\cdots$	e
$\vdots$				$\vdots$			$\ddots$				$\ddots$				$\ddots$			$\ddots$	
j_3	0	0	0	f	0	0	$\cdots$	0	0	0	$\cdots$	0	f	0	$\cdots$	0	0	$\cdots$	f

(29)

where $a_{N_k} = 1 - \frac{7}{4N_k^2}$, $b_{N_k} = \frac{3}{2N_k^2}$, $c_{N_k} = \frac{1}{4N_k^2}$ $(k = 0, 1, 2, 3)$, $d = \frac{3}{8}$, $e = \frac{1}{16}$, and $f = \frac{1}{4}$. The generalized subdivision matrix depends on the valence number of four corners of the element.

Appendix B: Geometry Preservation and Nested Property

In Section 3, we claim that eTHCCS can preserve the geometry and possesses nested property. In the following, we mathematically prove these two properties.

Proposition 1. *The geometry is preserved during the construction of eTHCCS from Level ℓ ($\ell \geq 0$) to Level $\ell + 1$.*

Proof. We prove this proposition by constructing Level $\ell + 1$ from Level ℓ ($\ell \geq 0$), showing that the geometry is the same before and after refinement. After refinement, active Level-ℓ elements remain the same, leading to the same geometry as before. Therefore, we only need to focus on the Level-$(\ell + 1)$ elements. Let $\Omega_k^{\ell+1}$ be a Level-$(\ell + 1)$ element obtained by refining a Level-ℓ element $\Omega_{k'}^{\ell}$. Thus we have $\Omega_k^{\ell+1} \subseteq \Omega_{k'}^{\ell}$. After refinement, the portion of the geometry ($\mathcal{S}|_{\Omega_k^{\ell+1}}$) corresponding to $\Omega_k^{\ell+1}$ is calculated as

$$\mathcal{S}|_{\Omega_k^{\ell+1}} = \sum_{i \in I_a^{\ell+1}} P_i^{\ell+1} B_i^{\ell+1} + \sum_{j \in T^{\ell}} P_j^{\ell} \mathrm{trun} B_j^{\ell} + \sum_{j \in I_a^{\ell} \backslash T^{\ell}} P_j^{\ell} B_j^{\ell}, \tag{30}$$

where I_a^{ℓ}, $I_a^{\ell+1}$ denote the index set of active Level-ℓ and Level-$(\ell + 1)$ non-truncated basis functions, T^{ℓ} is the index set of Level-ℓ truncated basis functions, and P_i^{ℓ}, $P_i^{\ell+1}$ are Level-ℓ and Level-$(\ell + 1)$ control points, respectively. Eq. (30) consists of a summation of three terms because active Level-$(\ell+1)$ basis functions ($B_j^{\ell+1}$), Level-ℓ truncated basis functions ($\mathrm{trun}B_j^{\ell}$) and other active Level-ℓ basis functions (B_j^{ℓ}) may all have support on $\Omega_k^{\ell+1}$. According to Eq. (22), $\mathrm{trun}B_j^{\ell}$ can be expressed as

$$\mathrm{trun} B_j^{\ell} = \sum_{i \in I^{\ell+1} \backslash I_a^{\ell+1}} c_{ji} B_i^{\ell+1}, \tag{31}$$

where $I^{\ell+1}$ represents an index set of basis functions associated with the Catmull-Clark mesh obtained by $\ell + 1$ subdivisions. Note that an active non-truncated Level-ℓ basis function B_j^{ℓ} ($j \in I_a^{\ell} \backslash T^{\ell}$) does not have any active

children at Level $\ell+1$ (otherwise it contradicts the definition of truncation). According to Eq. (21), we have

$$B_j^\ell = \sum_{i \in I^{\ell+1} \backslash I_a^{\ell+1}} c_{ji} B_i^{\ell+1}. \tag{32}$$

Substituting Eqs. (31) and (32) into Eq. (30), we have

$$\begin{aligned}
\mathcal{S}|_{\Omega_n^{\ell+1}} &= \sum_{i \in I_a^{\ell+1}} P_i^{\ell+1} B_i^{\ell+1} + \sum_{j \in T^\ell} P_j^\ell \left(\sum_{i \in I^{\ell+1} \backslash I_a^{\ell+1}} c_{ji} B_i^{\ell+1} \right) + \sum_{j \in I_a^\ell \backslash T^\ell} P_j^\ell \left(\sum_{i \in I^{\ell+1} \backslash I_a^{\ell+1}} c_{ji} B_i^{\ell+1} \right) \\
&= \sum_{i \in I_a^{\ell+1}} P_i^{\ell+1} B_i^{\ell+1} + \sum_{i \in I^{\ell+1} \backslash I_a^{\ell+1}} B_i^{\ell+1} \left(\sum_{j \in I_a^\ell} c_{ji} P_j^\ell \right) \\
&= \sum_{i \in I_a^{\ell+1}} P_i^{\ell+1} B_i^{\ell+1} + \sum_{i \in I^{\ell+1} \backslash I_a^{\ell+1}} B_i^{\ell+1} P_i^{\ell+1} \\
&= \sum_{i \in I^{\ell+1}} P_i^{\ell+1} B_i^{\ell+1}.
\end{aligned} \tag{33}$$

Note that $\sum_{j \in I_a^\ell} c_{ji} P_j^\ell = P_i^{\ell+1}$ ($i \in I^{\ell+1} \backslash I_a^{\ell+1}$) directly comes from the Catmull-Clark subdivision rule. Recall that the limit Catmull-Clark subdivision surface can be equivalently calculated from any control mesh in the global refinement sequence. Thus, Eq. (33) means that the limit surface is calculated by a Level-$(\ell+1)$ Catmull-Clark control mesh. Any other portions of the geometry can be handled in the same manner. Therefore, the limit surface does not change during the construction of eTHCCS.

□

The proof of Proposition 1 is similar to the proof of geometry preservation in [28]. In Eq. (31), c_{ji} can all be zero and B_j^ℓ does not contribute to the calculation of $\mathcal{S}|_{\Omega_n^{\ell+1}}$. Actually, this case implies that the children of B_j^ℓ are all active basis functions at Level $\ell+1$ and B_j^ℓ becomes passive at Level ℓ, which is exactly the same case as for THCCS construction [28]. The eTHCCS space also has nested property as described in Eq. (26) and Eq. (27). As the hierarchical level increases, the sub-domain decreases and the eTHCCS is enlarged. Since eTHCCS is recursively constructed, without loss of generality, we only need to prove that the nested property holds for two consecutive levels, Level ℓ ($\ell \geq 0$) and Level $\ell+1$.

Proposition 2. *Given an eTHCCS with levels up to ℓ ($\ell \geq 0$), Level $\ell+1$ is constructed using the basis-function-insertion scheme. The eTHCCS space is enlarged, that is,*

$$span\mathcal{B}_{eTHCCS}^\ell \subseteq span\mathcal{B}_{eTHCCS}^{\ell+1}, \tag{34}$$

where $\mathcal{B}_{eTHCCS}^\ell$ and $\mathcal{B}_{eTHCCS}^\ell$ contain the eTHCCS basis functions of ℓ levels and $\ell+1$ levels, respectively.

Proof. To prove Eq. (34), we only need to prove each basis function in $\mathcal{B}_{\text{eTHCCS}}^\ell$ can be represented by a linear combination of basis functions in $\mathcal{B}_{\text{eTHCCS}}^{\ell+1}$. During eTHCCS construction, the to-be-truncated Level-ℓ basis functions in $\mathcal{B}_{\text{eTHCCS}}^\ell$ are used to construct the Level-ℓ truncated basis functions in $\mathcal{B}_{\text{eTHCCS}}^{\ell+1}$. The other basis functions in $\mathcal{B}_{\text{eTHCCS}}^\ell$ remain the same in $\mathcal{B}_{\text{eTHCCS}}^{\ell+1}$. Therefore, we only need to check those to-be-truncated basis functions. Before truncation, a to-be-truncated basis function $B_i^\ell \in \mathcal{B}_{\text{THCCS}}^\ell$ can be expressed by a linear combination of its children,

$$B_i^\ell = \sum_{j \in C_i^\ell} c_{ij} B_j^{\ell+1} = \sum_{j \in I^{\ell+1}} c_{ij} B_j^{\ell+1} + \sum_{j \in C_i^\ell \backslash I^{\ell+1}} c_{ij} B_j^{\ell+1}, \tag{35}$$

where C_i^ℓ is the index set of the children of B_i^ℓ, and $I^{\ell+1}$ is the index set of newly inserted Level-$(\ell+1)$ basis functions. Note that $\sum_{i \in C_i^\ell \backslash I^{\ell+1}} c_{ij} B_j^{\ell+1}$ is actually the truncated basis function with respect to B_i^ℓ, and we have

$$B_i^\ell = \sum_{j \in I_a^{\ell+1}} c_{ij} B_j^{\ell+1} + \text{trun} B_i^\ell, \tag{36}$$

where $B_j^{\ell+1} \in \mathcal{B}_{\text{eTHCCS}}^{\ell+1}$ for $j \in I^{\ell+1}$, and $\text{trun}B_i^{\ell} \in \mathcal{B}_{\text{eTHCCS}}^{\ell+1}$. Therefore, any to-be-truncated basis function can be expressed by a linear combination of basis functions in $\mathcal{B}_{\text{eTHCCS}}^{\ell+1}$. To this end, we prove that any basis function in $\mathcal{B}_{\text{eTHCCS}}^{\ell}$ can be represented by a linear combination of those in $\mathcal{B}_{\text{eTHCCS}}^{\ell+1}$. Therefore Eq. (34) holds and Proposition 2 holds.

□

References

[1] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:229–263, 2010.

[2] 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.

[3] 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.

[4] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10:350–355, 1978.

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

[6] 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.

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

[8] D. R. Forsey and R. H. Bartels. Hierarchical B-spline refinement. *Computer Graphics*, 22:205–212, 1988.

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

[10] C. Giannelli, B. Jüttler, and H. Speleers. Strongly stable bases for adaptively refined multilevel spline spaces. *Advances in Computational Mathematics*, pages 1–32, 2013.

[11] E. Grinspun, P. Krysl, and P. Schröder. CHARMS: A simple framework for adaptive simulation. *ACM Transactions on Graphics*, 21:281–290, 2002.

[12] 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.

[13] 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.

[14] 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.

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

[16] L. Liu, Y. Zhang, T. J. R. Hughes, M. A. Scott, and T. W. Sederberg. Volumetric T-spline construction using Boolean operations. *Engineering with Computers*, 2014. DOI = 10.1007/s00366-013-0346-6.

[17] T. Nguyen, K. Karčiauskas, and J. Peters. A comparative study of several classical, discrete differential and isogeometric methods for solving Poisson's equation on the disk. *Axioms*, 3:280–300, 2014.

[18] J. Peters and X. Wu. On the local linear independence of generalized subdivision functions. *SIAM Journal on Numerical Analysis*, 44(6):2389 – 2407, 2006.

[19] M. Sabin. Recent progress in subdivision: a survey. In *Advances in Multiresolution for Geometric Modelling*, pages 203–230. 2005.

[20] M. A. Scott. *T-splines as a Design-Through-Analysis technology*. PhD thesis, The University of Texas at Austin, 2011.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] W. Wang, Y. Zhang, L. Liu, and T. J. R. Hughes. Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. *A Special Issue of Solid and Physical Modeling 2012 in Computer Aided Design*, 45:351–360, 2013.

[27] 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.

[28] 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.

[29] U. Zore, B. Jüttler, and J. Kosinka. On the linear independence of (truncated) hierarchical subdivision splines. *Geometry + Simulation Report*, (17), 2014.

[30] D. Zorin and P. Schröder. Subdivision for modeling and animation. *ACM SIGGRAGH Course Notes*, 2000.